Gene Golub SIAM Summer School 2013
Matrix Functions and Matrix Equations
July 22 – August 2, 2013
Fudan University, Shanghai, China

Matrix Equations and Model Reduction

Peter Benner

Max Planck Institute for Dynamics of Complex Technical Systems
Computational Methods in Systems and Control Theory
Magdeburg, Germany

benner@mpi-magdeburg.mpg.de

Outline

Outline

1. Introduction
   - Model Reduction for Dynamical Systems
   - Application Areas
   - Motivating Examples
2. Mathematical Basics
3. Model Reduction by Projection
4. Interpolatory Model Reduction
5. Balanced Truncation
6. Solving Large-Scale Matrix Equations
7. Final Remarks

Introduction

Model Reduction — Abstract Definition

Problem

Given a physical problem with dynamics described by the states $x \in \mathbb{R}^n$, where n is the dimension of the state space.

Because of redundancies, complexity, etc., we want to describe the dynamics of the system using a reduced number of states.

This is the task of model reduction (also: dimension reduction, order reduction).

Introduction

Model Reduction — Abstract Definition

Problem

Given a physical problem with dynamics described by the states $x \in \mathbb{R}^n$, where n is the dimension of the state space.

Because of redundancies, complexity, etc., we want to describe the dynamics of the system using a reduced number of states.

This is the task of model reduction (also: dimension reduction, order reduction).

Introduction

Model Reduction — Abstract Definition

Problem

Given a physical problem with dynamics described by the states $x \in \mathbb{R}^n$, where n is the dimension of the state space.

Because of redundancies, complexity, etc., we want to describe the dynamics of the system using a reduced number of states.

This is the task of model reduction (also: dimension reduction, order reduction).

Introduction

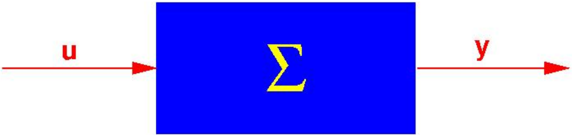
Model Reduction for Dynamical Systems

Dynamical Systems

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

with

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

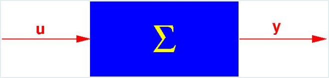


Model Reduction for Dynamical Systems

Original System

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

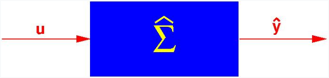
- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^q$.



Reduced-Order Model (ROM)

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

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



Goal:

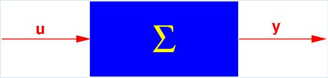
$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\|$ for all admissible input signals.

Model Reduction for Dynamical Systems

Original System

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

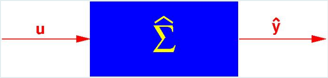
- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^q$.



Reduced-Order Model (ROM)

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

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



Goal:

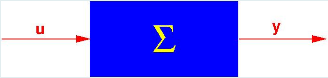
$\|y - \hat{y}\| <$ tolerance $\cdot \|u\|$ for all admissible input signals.

Model Reduction for Dynamical Systems

Original System

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

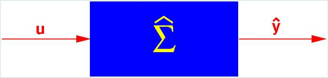
- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^q$.



Reduced-Order Model (ROM)

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

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



Goal:

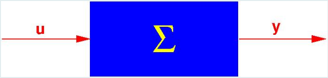
$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\|$ for all admissible input signals.

Model Reduction for Dynamical Systems

Original System

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

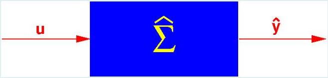
- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^q$.



Reduced-Order Model (ROM)

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

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



Goal:

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

Secondary goal: reconstruct approximation of x from $\hat{x}$.

Model Reduction for Dynamical Systems

Parameter-Dependent Dynamical Systems

Dynamical Systems

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

with

- (generalized) states $x(t;p) \in \mathbb{R}^n$ ($E \in \mathbb{R}^{n \times n}$),
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t;p) \in \mathbb{R}^q$, (b) is called output equation,
- $p \in \Omega \subset \mathbb{R}^d$ is a parameter vector, Ω is bounded.

Applications:

- Repeated simulation for varying material or geometry parameters, boundary conditions,
- Control, optimization and design.

Requirement: keep parameters as symbolic quantities in ROM.

Model Reduction for Dynamical Systems

Parameter-Dependent Dynamical Systems

Dynamical Systems

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

with

- (generalized) states $x(t;p) \in \mathbb{R}^n$ ($E \in \mathbb{R}^{n\times n}$),
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t;p) \in \mathbb{R}^q$, (b) is called output equation,
- $p \in \Omega \subset \mathbb{R}^d$ is a parameter vector, Ω is bounded.

Applications:

- Repeated simulation for varying material or geometry parameters, boundary conditions,
- Control, optimization and design.

Requirement: keep parameters as symbolic quantities in ROM.

Model Reduction for Dynamical Systems

Linear Systems

Linear, Time-Invariant (LTI) Systems

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

Model Reduction for Dynamical Systems

Linear Systems

Linear, Time-Invariant (LTI) Systems

$$
\begin{array}{rclclll}
E\dot{x} & = & f(t,x,u) & = & Ax + Bu, & E, A \in \mathbb{R}^{n\times n}, & B \in \mathbb{R}^{n\times m}, \\
y & = & g(t,x,u) & = & Cx + Du, & C \in \mathbb{R}^{q\times n}, & D \in \mathbb{R}^{q\times m}.
\end{array}
$$

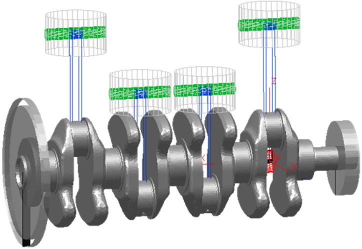
Linear, Time-Invariant Parametric Systems

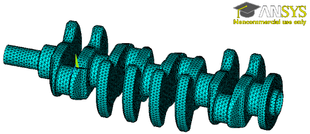
$$
\begin{array}{rcl}
E(p)\dot{x}(t;p) & = & A(p)x(t;p) + B(p)u(t), \\
y(t;p) & = & C(p)x(t;p) + D(p)u(t),
\end{array}
$$

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

Application Areas

Structural Mechanics / Finite Element Modeling **since ∼1960ies**



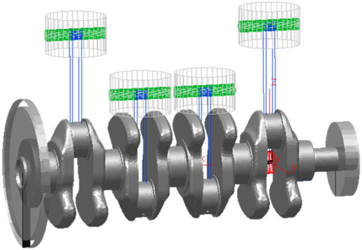


- Resolving complex 3D geometries ⇒ millions of degrees of freedom.
- Analysis of elastic deformations requires many simulation runs for varying external forces.

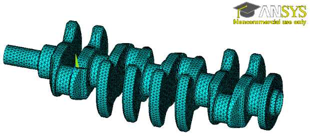
Standard MOR techniques in structural mechanics: modal truncation, combined with Guyan reduction (static condensation) ⇝ Craig-Bampton method.

Application Areas

Structural Mechanics / Finite Element Modeling since ∼1960ies



$\rightsquigarrow$



- Resolving complex 3D geometries $\Rightarrow$ millions of degrees of freedom.
- Analysis of elastic deformations requires many simulation runs for varying external forces.

Standard MOR techniques in structural mechanics: modal truncation, combined with Guyan reduction (static condensation) $\rightsquigarrow$ Craig-Bampton method.

Application Areas

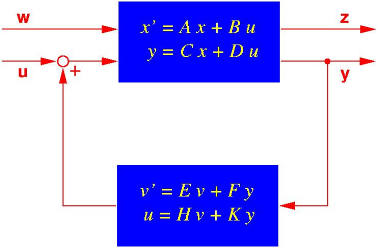
(Optimal) Control — **since ∼1980ies**

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.

$\Longrightarrow$ reduce order of plant (n) and/or controller (N).

Standard MOR techniques in systems and control: balanced truncation and related methods.

Application Areas

(Optimal) Control

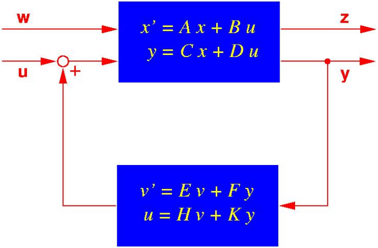
since ∼1980ies

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.

$\Longrightarrow$ reduce order of plant (n) and/or controller (N).

Standard MOR techniques in systems and control: balanced truncation and related methods.

Application Areas

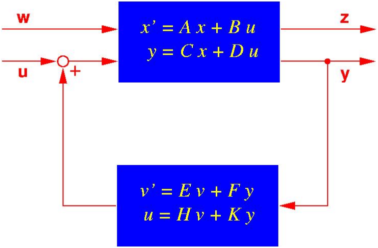
(Optimal) Control **since ∼1980ies**

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.

$\Longrightarrow$ reduce order of plant (n) and/or controller (N).

Standard MOR techniques in systems and control: balanced truncation and related methods.

Application Areas

(Optimal) Control

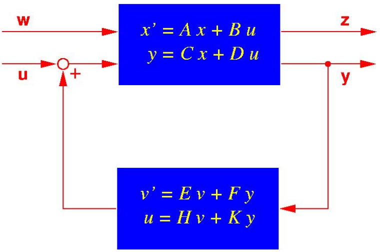
since ∼1980ies

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.

$\Longrightarrow$ reduce order of plant (n) and/or controller (N).

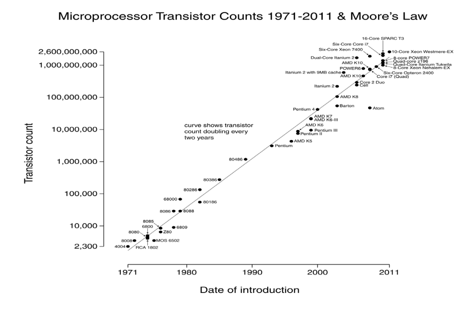
Standard MOR techniques in systems and control: balanced truncation and related methods.

Application Areas

Micro Electronics/Circuit Simulation since ∼1990ies

Progressive miniaturization

- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.
- **Moore's Law (1965/75)** states that the number of on-chip transistors doubles each 24 months.



Source: http://en.wikipedia.org/wiki/File:Transistor_Count_and_Moore'sLaw_-_2011.svg

Application Areas

Micro Electronics/Circuit Simulation **since ∼1990ies**

Progressive miniaturization

- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.
- **Moore's Law (1965/75)** ⇝ steady increase of describing equations, i.e., network topology (Kirchhoff's laws) and characteristic element/semi-conductor equations.

Application Areas

Micro Electronics/Circuit Simulation

since ∼1990ies

Progressive miniaturization

- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.
- **Moore's Law (1965/75)** $\rightsquigarrow$ steady increase of describing equations, i.e., network topology (Kirchhoff's laws) and characteristic element/semi-conductor equations.
- Increase in packing density and multilayer technology requires modeling of interconncet to ensure that thermic/electro-magnetic effects do not disturb signal transmission.

Intel 4004 (1971)	Intel Core 2 Extreme (quad-core) (2007)
1 layer, 10μ technology	9 layers, $45nm$ technology
2,300 transistors	$> 8,200,000$ transistors
64 kHz clock speed	> 3 GHz clock speed.

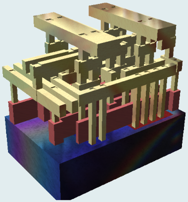
Application Areas

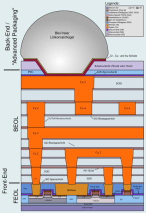
Micro Electronics/Circuit Simulation

since ∼1990ies

Progressive miniaturization

- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.
- **Moore's Law (1965/75)** ⇝ steady increase of describing equations, i.e., network topology (Kirchhoff's laws) and characteristic element/semi-conductor equations.
- Increase in packing density and multilayer technology requires modeling of interconncet to ensure that thermic/electro-magnetic effects do not disturb signal transmission.





Source: http://en.wikipedia.org/wiki/Image:Silicon_chip_3d.png.

Application Areas

Micro Electronics/Circuit Simulation **since $\sim$1990ies**

Progressive miniaturization

- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.
- **Moore's Law (1965/75)** $\rightsquigarrow$ steady increase of describing equations, i.e., network topology (Kirchhoff's laws) and characteristic element/semi-conductor equations.
- Here: mostly MOR for linear systems, they occur in micro electronics through modified nodal analysis (MNA) for RLC networks. e.g., when
  - decoupling large linear subcircuits,
  - modeling transmission lines,
  - modeling pin packages in VLSI chips,
  - modeling circuit elements described by Maxwell's equation using partial element equivalent circuits (PEEC).

Application Areas

Micro Electronics/Circuit Simulation **since ∼1990ies**

Progressive miniaturization

- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.
- **Moore's Law (1965/75)** $\rightsquigarrow$ steady increase of describing equations, i.e., network topology (Kirchhoff's laws) and characteristic element/semi-conductor equations.

$\rightsquigarrow$ Clear need for model reduction techniques in order to facilitate or even enable circuit simulation for current and future VLSI design.

Application Areas

Micro Electronics/Circuit Simulation **since ∼1990ies**

Progressive miniaturization

- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.
- **Moore's Law (1965/75)** ⇝ steady increase of describing equations, i.e., network topology (Kirchhoff's laws) and characteristic element/semi-conductor equations.

⇝ Clear need for model reduction techniques in order to facilitate or even enable circuit simulation for current and future VLSI design.

Standard MOR techniques in circuit simulation:
Krylov subspace / Padé approximation / rational interpolation methods.

Application Areas

Many other disciplines in computational sciences and engineering like

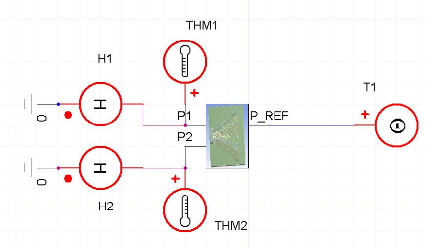
- computational fluid dynamics (CFD),
- computational electromagnetics,
- chemical process engineering,
- design of MEMS/NEMS (micro/nano-electrical-mechanical systems),
- computational acoustics,
- ...

Motivating Examples

Electro-Thermic Simulation of Integrated Circuit (IC)

[Source: Evgenii Rudnyi, CADFEM GmbH]

- SIMPLORER® test circuit with 2 transistors.



- Conservative thermic sub-system in SIMPLORER:
  voltage ⇝ temperature, current ⇝ heat flow.
- Original model: $n = 270.593$, $m = q = 2$ $\Rightarrow$
  Computing time (on Intel Xeon dualcore 3GHz, 1 Thread):
  - Main computational cost for set-up data $\approx 22min$.
  - Computation of reduced models from set-up data: 44–49sec. ($r = 20$–70).
  - Bode plot (MATLAB on Intel Core i7, 2,67GHz, 12GB):
    7.5h for original system, $<$ 1min for reduced system.
  - Speed-up factor: 18 including / $\geq$ 450 excluding reduced model generation!

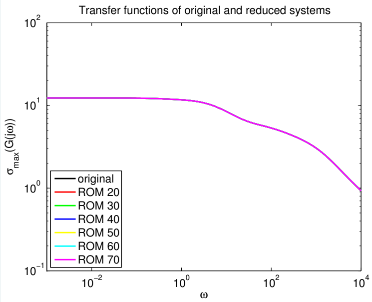
Motivating Examples

Electro-Thermic Simulation of Integrated Circuit (IC)

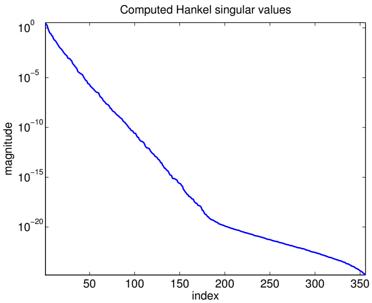
[Source: Evgenii Rudnyi, CADFEM GmbH]

- Original model: $n = 270.593$, $m = q = 2 \Rightarrow$
  Computing time (on Intel Xeon dualcore 3GHz, 1 Thread):
  - Main computational cost for set-up data $\approx 22min$.
  - Computation of reduced models from set-up data: 44–49sec. ($r = 20$–70).
  - Bode plot (MATLAB on Intel Core i7, 2,67GHz, 12GB):
    7.5h for original system, $<$ 1min for reduced system.
  - Speed-up factor: 18 including / $\geq$ 450 excluding reduced model generation!

Bode Plot (Amplitude)



Hankel Singular Values



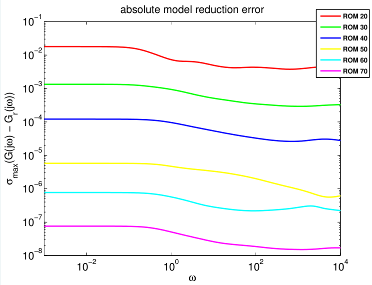
Motivating Examples

Electro-Thermic Simulation of Integrated Circuit (IC)

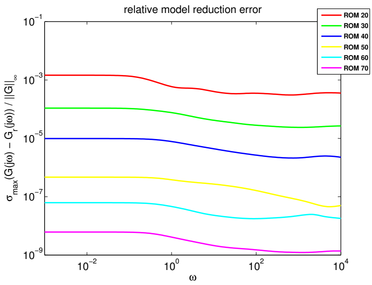
[Source: Evgenii Rudnyi, CADFEM GmbH]

- Original model: $n = 270.593$, $m = q = 2 \Rightarrow$
  Computing time (on Intel Xeon dualcore 3GHz, 1 Thread):
  - Main computational cost for set-up data $\approx 22min$.
  - Computation of reduced models from set-up data: 44–49sec. ($r = 20$–70).
  - Bode plot (MATLAB on Intel Core i7, 2,67GHz, 12GB):
    7.5h for original system, $<$ 1min for reduced system.
  - Speed-up factor: 18 including / $\geq$ 450 excluding reduced model generation!

Absolute Error



Relative Error



Motivating Examples

A Nonlinear Model from Computational Neurosciences: the FitzHugh-Nagumo System

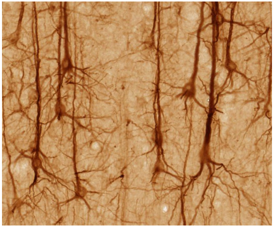
- Simple model for neuron (de-)activation [CHATURANTABUT/SORENSEN 2009]

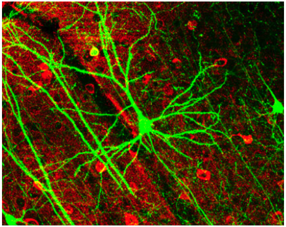
$$\epsilon v_t(x,t) = \epsilon^2 v_{xx}(x,t) + f(v(x,t)) - w(x,t) + g,$$
$$w_t(x,t) = hv(x,t) - \gamma w(x,t) + g,$$

with $f(v) = v(v-0.1)(1-v)$ and initial and boundary conditions

$$v(x,0) = 0, \qquad w(x,0) = 0, \qquad x \in [0,1]$$
$$v_x(0,t) = -i_0(t), \qquad v_x(1,t) = 0, \qquad t \geq 0,$$

where $\epsilon = 0.015, h = 0.5, \gamma = 2, g = 0.05, i_0(t) = 50000t^3 \exp(-15t)$.





Source: http://en.wikipedia.org/wiki/Neuron

Motivating Examples

A Nonlinear Model from Computational Neurosciences: the FitzHugh-Nagumo System

- Simple model for neuron (de-)activation [Chaturantabut/Sorensen 2009]

$$\epsilon v_t(x,t) = \epsilon^2 v_{xx}(x,t) + f(v(x,t)) - w(x,t) + g,$$
$$w_t(x,t) = hv(x,t) - \gamma w(x,t) + g,$$

with $f(v) = v(v-0.1)(1-v)$ and initial and boundary conditions

$$v(x,0) = 0, \qquad w(x,0) = 0, \qquad x \in [0,1]$$
$$v_x(0,t) = -i_0(t), \qquad v_x(1,t) = 0, \qquad t \geq 0,$$

where $\epsilon = 0.015, h = 0.5, \gamma = 2, g = 0.05, i_0(t) = 50000t^3 \exp(-15t)$.

- Parameter g handled as an additional input.
- Original state dimension $n = 2 \cdot 400$, QBDAE dimension $N = 3 \cdot 400$, reduced QBDAE dimension $r = 26$, chosen expansion point $\sigma = 1$.

Motivating Examples

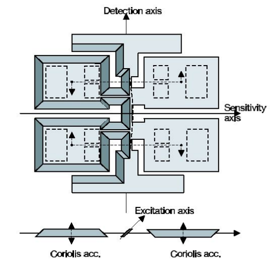
A Nonlinear Model from Computational Neurosciences: the FitzHugh-Nagumo System

Motivating Examples

Parametric MOR: Applications in Microsystems/MEMS Design

Microgyroscope (butterfly gyro)

- Application: inertial navigation.



- Voltage applied to electrodes induces vibration of wings, resulting rotation due to Coriolis force yields sensor data.
- FE model of second order: $N = 17.361 \rightsquigarrow n = 34.722,\ m = 1,\ q = 12.$
- Sensor for position control based on acceleration and rotation.

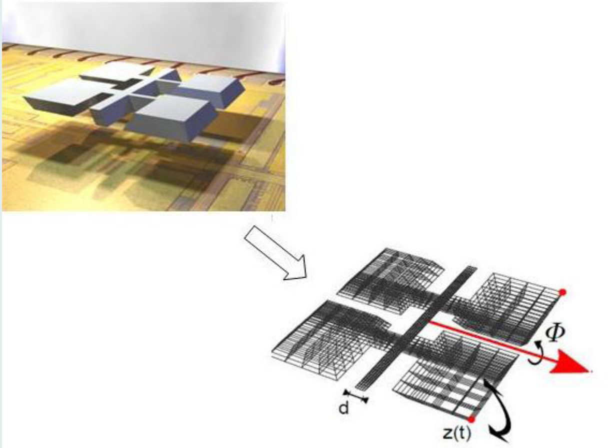
Source: The Oberwolfach Benchmark Collection http://www.imtek.de/simulation/benchmark

Motivating Examples

Parametric MOR: Applications in Microsystems/MEMS Design

Microgyroscope (butterfly gyro)

Parametric FE model: $M(d)\ddot{x}(t) + D(\Phi, d, \alpha, \beta)\dot{x}(t) + T(d)x(t) = Bu(t).$



Motivating Examples

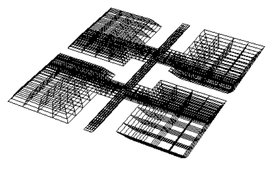
Parametric MOR: Applications in Microsystems/MEMS Design

Microgyroscope (butterfly gyro)

Parametric FE model:

$$M(d)\ddot{x}(t) + D(\Phi, d, \alpha, \beta)\dot{x}(t) + T(d)x(t) = Bu(t),$$

wobei



$$\begin{aligned} M(d) &= M_1 + dM_2, \\ D(\Phi, d, \alpha, \beta) &= \Phi(D_1 + dD_2) + \alpha M(d) + \beta T(d), \\ T(d) &= T_1 + \frac{1}{d}T_2 + dT_3, \end{aligned}$$

with

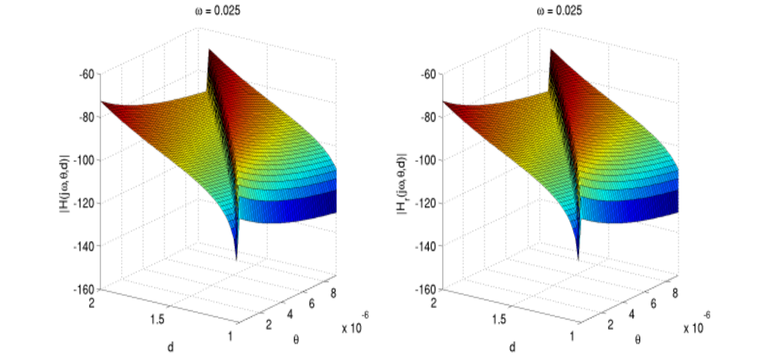
- width of bearing: d,
- angular velocity: Φ,
- Rayleigh damping parameters: α, β.

Motivating Examples

Parametric MOR: Applications in Microsystems/MEMS Design

Microgyroscope (butterfly gyro)

Original. . . and reduced-order model.



Outline

Numerical Linear Algebra

Image Compression by Truncated SVD

- A digital image with $n_x \times n_y$ pixels can be represented as matrix $X \in \mathbb{R}^{n_x \times n_y}$, where x_{ij} contains color information of pixel (i, j).
- Memory (in single precision): $4 \cdot n_x \cdot n_y$ bytes.

Theorem (Schmidt-Mirsky/Eckart-Young)

Best rank-r approximation to $X \in \mathbb{R}^{n_x \times n_y}$ w.r.t. spectral norm:

$$\widehat{X} = \sum_{j=1}^{r} \sigma_j u_j v_j^T,$$

where $X = U\Sigma V^T$ is the singular value decomposition (SVD) of X.
The approximation error is $\|X - \widehat{X}\|_2 = \sigma_{r+1}$.

Idea for dimension reduction

Instead of X save $u_1, \ldots, u_r$, $\sigma_1 v_1, \ldots, \sigma_r v_r$.
$\rightsquigarrow$ memory $= 4r \times (n_x + n_y)$ bytes.

Numerical Linear Algebra

Image Compression by Truncated SVD

- A digital image with $n_x \times n_y$ pixels can be represented as matrix $X \in \mathbb{R}^{n_x \times n_y}$, where x_{ij} contains color information of pixel (i, j).
- Memory (in single precision): $4 \cdot n_x \cdot n_y$ bytes.

Theorem (Schmidt-Mirsky/Eckart-Young)

Best rank-r approximation to $X \in \mathbb{R}^{n_x \times n_y}$ w.r.t. spectral norm:

$$\widehat{X} = \sum_{j=1}^{r} \sigma_j u_j v_j^T,$$

where $X = U \Sigma V^T$ is the singular value decomposition (SVD) of X.
The approximation error is $\|X - \widehat{X}\|_2 = \sigma_{r+1}$.

Idea for dimension reduction

Instead of X save $u_1, \ldots, u_r$, $\sigma_1 v_1, \ldots, \sigma_r v_r$.
$\rightsquigarrow$ memory $= 4r \times (n_x + n_y)$ bytes.

Numerical Linear Algebra

Image Compression by Truncated SVD

- A digital image with $n_x \times n_y$ pixels can be represented as matrix $X \in \mathbb{R}^{n_x \times n_y}$, where x_{ij} contains color information of pixel (i, j).
- Memory (in single precision): $4 \cdot n_x \cdot n_y$ bytes.

Theorem (Schmidt-Mirsky/Eckart-Young)

Best rank-r approximation to $X \in \mathbb{R}^{n_x \times n_y}$ w.r.t. spectral norm:

$$\widehat{X} = \sum_{j=1}^{r} \sigma_j u_j v_j^T,$$

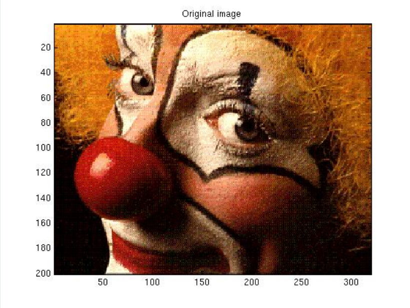
where $X = U \Sigma V^T$ is the singular value decomposition (SVD) of X.
The approximation error is $\|X - \widehat{X}\|_2 = \sigma_{r+1}$.

Idea for dimension reduction

Instead of X save $u_1, \ldots, u_r, \sigma_1 v_1, \ldots, \sigma_r v_r$.
$\rightsquigarrow$ memory $= 4r \times (n_x + n_y)$ bytes.

Example: Image Compression by Truncated SVD

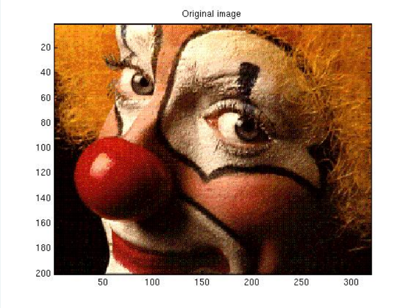
Example: Clown



320×200 pixel
$\rightsquigarrow \quad \approx 256$ kB

Example: Image Compression by Truncated SVD

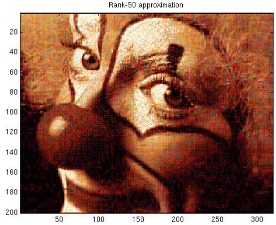
Example: Clown



320×200 pixel

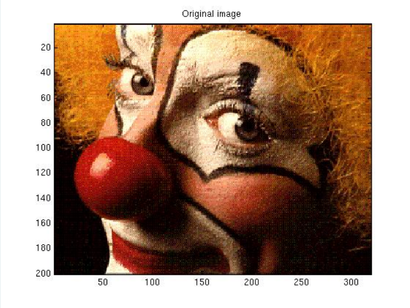
$\rightsquigarrow \quad \approx 256$ kB

- rank $r = 50$, ≈ 104 kB



Example: Image Compression by Truncated SVD

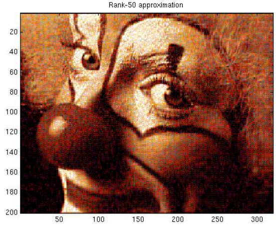
Example: Clown



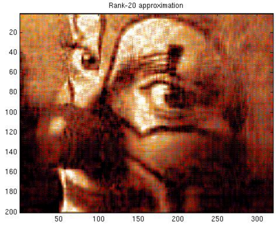
320×200 pixel

$\rightsquigarrow \quad \approx 256$ kB

- rank $r = 50$, ≈ 104 kB



- rank $r = 20$, ≈ 42 kB



Dimension Reduction via SVD

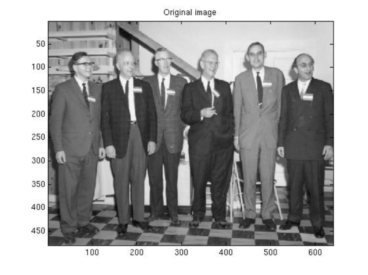
Example: Gatlinburg

Organizing committee
Gatlinburg/Householder Meeting 1964:

James H. Wilkinson, Wallace Givens,

George Forsythe, Alston Householder,

Peter Henrici, Fritz L. Bauer.



640×480 pixel, ≈ 1229 kB

Dimension Reduction via SVD

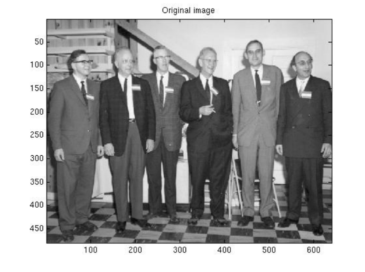
Example: Gatlinburg

Organizing committee
Gatlinburg/Householder Meeting 1964:

James H. Wilkinson, Wallace Givens,

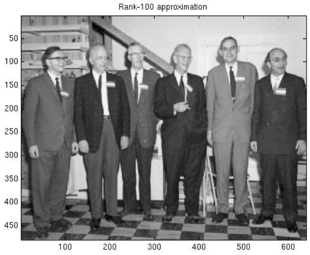
George Forsythe, Alston Householder,

Peter Henrici, Fritz L. Bauer.

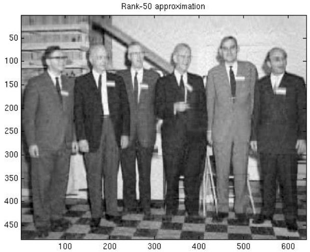


640×480 pixel, ≈ 1229 kB

- rank $r = 100$, ≈ 448 kB



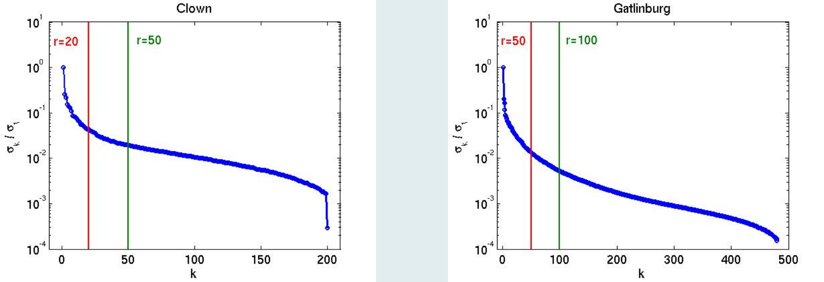
- rank $r = 50$, ≈ 224 kB



Background: Singular Value Decay

Image data compression via SVD works, if the singular values decay (exponentially).

Singular Values of the Image Data Matrices



Systems and Control Theory
The Laplace transform

Definition

The Laplace transform of a time domain function $f \in L_{1,\mathrm{loc}}$ with $\mathrm{dom}\,(f) = \mathbb{R}_0^+$ is

$$\mathcal{L} : f(t) \mapsto f(s) := \mathcal{L}\{f(t)\}(s) := \int_0^\infty e^{-st} f(t)\,dt, \quad s \in \mathbb{C}.$$

F is a function in the (Laplace or) frequency domain.

Note: for frequency domain evaluations ("frequency response analysis"), one takes $\mathrm{re}\, s = 0$ and $\mathrm{im}\, s \geq 0$. Then $\omega := \mathrm{im}\, s$ takes the role of a frequency (in [rad/s], i.e., $\omega = 2\pi v$ with v measured in [Hz]).

Systems and Control Theory

The Laplace transform

Definition

The Laplace transform of a time domain function $f \in L_{1,\mathrm{loc}}$ with $\mathrm{dom}\,(f) = \mathbb{R}_0^+$ is

$$\mathcal{L} : f(t) \mapsto f(s) := \mathcal{L}\{f(t)\}(s) := \int_0^\infty e^{-st} f(t)\, dt, \quad s \in \mathbb{C}.$$

F is a function in the (Laplace or) frequency domain.

Note: for frequency domain evaluations ("frequency response analysis"), one takes $\mathrm{re}\, s = 0$ and $\mathrm{im}\, s \geq 0$. Then $\omega := \mathrm{im}\, s$ takes the role of a frequency (in [rad/s], i.e., $\omega = 2\pi v$ with v measured in [Hz]).

Lemma

$$\mathcal{L}\{\dot{f}(t)\}(s) = sF(s).$$

Systems and Control Theory
The Laplace transform

Definition

The Laplace transform of a time domain function $f \in L_{1,\mathrm{loc}}$ with $\mathrm{dom}\,(f) = \mathbb{R}_0^+$ is

$$\mathcal{L} : f(t) \mapsto f(s) := \mathcal{L}\{f(t)\}(s) := \int_0^\infty e^{-st} f(t)\, dt, \quad s \in \mathbb{C}.$$

F is a function in the (Laplace or) frequency domain.

Lemma

$$\mathcal{L}\{\dot{f}(t)\}(s) = sF(s).$$

Note: for ease of notation, in the following we will use lower-case letters for both, a function and its Laplace transform!

Systems and Control Theory

The Model Reduction Problem as Approximation Problem in Frequency Domain

Linear Systems in Frequency Domain

Application of Laplace transform $(x(t) \mapsto x(s), \dot{x}(t) \mapsto sx(s))$ to linear system

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

with $x(0) = 0$ yields:

$$sEx(s) = Ax(s) + Bu(s), \quad y(s) = Cx(s) + Du(s),$$

Systems and Control Theory

The Model Reduction Problem as Approximation Problem in Frequency Domain

Linear Systems in Frequency Domain

Application of Laplace transform $(x(t) \mapsto x(s),\ \dot{x}(t) \mapsto sx(s))$ to linear system

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

with $x(0) = 0$ yields:

$$sEx(s) = Ax(s) + Bu(s), \quad y(s) = Cx(s) + Du(s),$$

$\Longrightarrow$ I/O-relation in frequency domain:

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

$G(s)$ is the transfer function of Σ.

Systems and Control Theory

The Model Reduction Problem as Approximation Problem in Frequency Domain

Linear Systems in Frequency Domain

Application of Laplace transform $(x(t) \mapsto x(s),\ \dot{x}(t) \mapsto sx(s))$ to linear system

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

with $x(0) = 0$ yields:

$$sEx(s) = Ax(s) + Bu(s), \quad y(s) = Cx(s) + Du(s),$$

$\Longrightarrow$ I/O-relation in frequency domain:

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

$G(s)$ is the transfer function of Σ.

Goal: Fast evaluation of mapping $u \to y$.

Systems and Control Theory

The Model Reduction Problem as Approximation Problem in Frequency Domain

Linear Systems in Frequency Domain

Application of Laplace transform $(x(t) \mapsto x(s), \dot{x}(t) \mapsto sx(s))$ to linear system

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

with $x(0) = 0$ yields:

$$sEx(s) = Ax(s) + Bu(s), \quad y(s) = Cx(s) + Du(s),$$

$\Longrightarrow$ I/O-relation in frequency domain:

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

$G(s)$ is the transfer function of Σ.

Goal: Fast evaluation of mapping $u \rightarrow y$.

Example.

Systems and Control Theory

The Model Reduction Problem as Approximation Problem in Frequency Domain

Formulating model reduction in frequency domain

Approximate the dynamical system

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

by reduced-order system

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

of order $r \ll n$, such that

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

Systems and Control Theory

The Model Reduction Problem as Approximation Problem in Frequency Domain

Formulating model reduction in frequency domain

Approximate the dynamical system

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

by reduced-order system

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

of order $r \ll n$, such that

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

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

Systems and Control Theory

Properties of linear systems

Definition

A linear system

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

is stable if its transfer function $G(s)$ has all its poles in the left half plane and it is asymptotically (or Lyapunov or exponentially) stable if all poles are in the open left half plane $\mathbb{C}^- := \{z \in \mathbb{C} \,|\, \Re(z) < 0\}$.

Lemma

Sufficient for asymptotic stability is that A is asymptotically stable (or Hurwitz), i.e., the spectrum of $A - \lambda E$, denoted by $\Lambda(A, E)$, satisfies $\Lambda(A, E) \subset \mathbb{C}^-$.

Note that by abuse of notation, often *stable system* is used for asymptotically stable systems.

Systems and Control Theory
Properties of linear systems

Definition

A linear system

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

is stable if its transfer function $G(s)$ has all its poles in the left half plane and it is asymptotically (or Lyapunov or exponentially) stable if all poles are in the open left half plane $\mathbb{C}^- := \{z \in \mathbb{C} \,|\, \Re(z) < 0\}$.

Lemma

Sufficient for asymptotic stability is that A is asymptotically stable (or Hurwitz), i.e., the spectrum of $A - \lambda E$, denoted by $\Lambda(A, E)$, satisfies $\Lambda(A, E) \subset \mathbb{C}^-$.

Note that by abuse of notation, often *stable system* is used for asymptotically stable systems.

Systems and Control Theory

Properties of linear systems

Further properties to be discussed:

- Controllability/reachability
- Observability
- Stabilizability
- Detectability

See handout "Mathematical Basics".

Systems and Control Theory

Realizations of Linear Systems (with $E = I_n$ for simplicity)

Definition

For a linear (time-invariant) system

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t) + Du(t), \end{cases} \qquad \begin{array}{l} \text{with transfer function} \\ G(s) = C(sI - A)^{-1}B + D, \end{array}$$

the quadruple $(A, B, C, D) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{q\times n} \times \mathbb{R}^{q\times m}$ is called a realization of Σ.

Systems and Control Theory

Realizations of Linear Systems (with $E = I_n$ for simplicity)

Definition

For a linear (time-invariant) system

$$\Sigma : \begin{cases} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{cases} \qquad \begin{array}{l} \text{with transfer function} \\ G(s) = C(sI - A)^{-1}B + D, \end{array}$$

the quadruple $(A, B, C, D) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{q\times n} \times \mathbb{R}^{q\times m}$ is called a realization of Σ.

Realizations are not unique!

Transfer function is invariant under state-space transformations,

$$\mathcal{T} : \begin{cases} x & \to & Tx, \\ (A, B, C, D) & \to & (TAT^{-1}, TB, CT^{-1}, D), \end{cases}$$

Systems and Control Theory

Realizations of Linear Systems (with $E = I_n$ for simplicity)

Definition

For a linear (time-invariant) system

$$\Sigma : \begin{cases} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{cases} \qquad \begin{array}{l}\text{with transfer function} \\ G(s) = C(sI - A)^{-1}B + D,\end{array}$$

the quadruple $(A, B, C, D) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{q\times n} \times \mathbb{R}^{q\times m}$ is called a realization of Σ.

Realizations are not unique!

Transfer function is invariant under addition of uncontrollable/unobservable states:

$$\frac{d}{dt}\begin{bmatrix} x \\ x_1 \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A_1 \end{bmatrix}\begin{bmatrix} x \\ x_1 \end{bmatrix} + \begin{bmatrix} B \\ B_1 \end{bmatrix} u(t), \quad y(t) = \begin{bmatrix} C & 0 \end{bmatrix}\begin{bmatrix} x \\ x_1 \end{bmatrix} + Du(t),$$

$$\frac{d}{dt}\begin{bmatrix} x \\ x_2 \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A_2 \end{bmatrix}\begin{bmatrix} x \\ x_2 \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t), \quad y(t) = \begin{bmatrix} C & C_2 \end{bmatrix}\begin{bmatrix} x \\ x_2 \end{bmatrix} + Du(t),$$

for arbitrary $A_j \in \mathbb{R}^{n_j\times n_j}$, $j = 1, 2$, $B_1 \in \mathbb{R}^{n_1\times m}$, $C_2 \in \mathbb{R}^{q\times n_2}$ and any $n_1, n_2 \in \mathbb{N}$.

Systems and Control Theory

Realizations of Linear Systems (with $E = I_n$ for simplicity)

Definition

For a linear (time-invariant) system

$$\Sigma : \begin{cases} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{cases} \quad \text{with transfer function} \quad G(s) = C(sI - A)^{-1}B + D,$$

the quadruple $(A, B, C, D) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{q\times n} \times \mathbb{R}^{q\times m}$ is called a realization of Σ.

Realizations are not unique!

Hence,

$$(A, B, C, D), \qquad \left(\begin{bmatrix} A & 0 \\ 0 & A_1 \end{bmatrix}, \begin{bmatrix} B \\ B_1 \end{bmatrix}, \begin{bmatrix} C & 0 \end{bmatrix}, D \right),$$

$$(TAT^{-1}, TB, CT^{-1}, D), \qquad \left(\begin{bmatrix} A & 0 \\ 0 & A_2 \end{bmatrix}, \begin{bmatrix} B \\ 0 \end{bmatrix}, \begin{bmatrix} C & C_2 \end{bmatrix}, D \right),$$

are all realizations of Σ!

Systems and Control Theory

Realizations of Linear Systems (with $E = I_n$ for simplicity)

Definition

For a linear (time-invariant) system

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t) + Du(t), \end{cases} \quad \begin{array}{l} \text{with transfer function} \\ G(s) = C(sI - A)^{-1}B + D, \end{array}$$

the quadruple $(A, B, C, D) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{q\times n} \times \mathbb{R}^{q\times m}$ is called a realization of Σ.

Definition

The McMillan degree of Σ is the unique minimal number $\hat{n} \geq 0$ of states necessary to describe the input-output behavior completely.
A minimal realization is a realization $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ of Σ with order $\hat{n}$.

Systems and Control Theory

Realizations of Linear Systems (with $E = I_n$ for simplicity)

Definition

For a linear (time-invariant) system

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t) + Du(t), \end{cases} \qquad \text{with transfer function} \quad G(s) = C(sI - A)^{-1}B + D,$$

the quadruple $(A, B, C, D) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{q\times n} \times \mathbb{R}^{q\times m}$ is called a realization of Σ.

Definition

The McMillan degree of Σ is the unique minimal number $\hat{n} \geq 0$ of states necessary to describe the input-output behavior completely.
A minimal realization is a realization $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ of Σ with order $\hat{n}$.

Theorem

A realization (A, B, C, D) of a linear system is minimal $\iff$
(A, B) is controllable and (A, C) is observable.

Systems and Control Theory
Balanced Realizations

Definition

A realization (A, B, C, D) of a linear system Σ is balanced if its infinite controllability/observability Gramians P/Q satisfy

$$P = Q = \operatorname{diag}\{\sigma_1, \ldots, \sigma_n\} \quad (\text{w.l.o.g. } \sigma_j \geq \sigma_{j+1},\ j = 1, \ldots, n-1).$$

Systems and Control Theory

Balanced Realizations

Definition

A realization (A, B, C, D) of a linear system Σ is balanced if its infinite controllability/observability Gramians P/Q satisfy

$$P = Q = \operatorname{diag}\{\sigma_1, \ldots, \sigma_n\} \quad (\text{w.l.o.g. } \sigma_j \geq \sigma_{j+1},\ j = 1, \ldots, n-1).$$

When does a balanced realization exist?

Systems and Control Theory

Balanced Realizations

Definition

A realization (A, B, C, D) of a linear system Σ is balanced if its infinite controllability/observability Gramians P/Q satisfy

$$P = Q = \operatorname{diag}\{\sigma_1, \ldots, \sigma_n\} \quad (\text{w.l.o.g. } \sigma_j \geq \sigma_{j+1},\ j = 1, \ldots, n-1).$$

When does a balanced realization exist?
Assume A to be Hurwitz, i.e. $\Lambda(A) \subset \mathbb{C}^-$. Then:

Theorem

Given a stable minimal linear system $\Sigma : (A, B, C, D)$, a balanced realization is obtained by the state-space transformation with

$$T_b := \Sigma^{-\frac{1}{2}} V^T R,$$

where $P = S^T S$, $Q = R^T R$ (e.g., Cholesky decompositions) and $SR^T = U \Sigma V^T$ is the SVD of SR^T.

Proof. Exercise!

Systems and Control Theory
Balanced Realizations

Definition

A realization (A, B, C, D) of a stable linear system Σ is balanced if its infinite controllability/observability Gramians P/Q satisfy

$$P = Q = \operatorname{diag}\{\sigma_1, \ldots, \sigma_n\} \quad (\text{w.l.o.g. } \sigma_j \geq \sigma_{j+1},\ j = 1, \ldots, n-1).$$

$\sigma_1, \ldots, \sigma_n$ are the Hankel singular values of Σ.

Note: $\sigma_1, \ldots, \sigma_n \geq 0$ as $P, Q \geq 0$ by definition, and $\sigma_1, \ldots, \sigma_n > 0$ in case of minimality!

Systems and Control Theory

Balanced Realizations

Definition

A realization (A, B, C, D) of a stable linear system Σ is balanced if its infinite controllability/observability Gramians P/Q satisfy

$$P = Q = \operatorname{diag}\{\sigma_1, \ldots, \sigma_n\} \quad (\text{w.l.o.g. } \sigma_j \geq \sigma_{j+1},\ j = 1, \ldots, n-1).$$

$\sigma_1, \ldots, \sigma_n$ are the Hankel singular values of Σ.

Note: $\sigma_1, \ldots, \sigma_n \geq 0$ as $P, Q \geq 0$ by definition, and $\sigma_1, \ldots, \sigma_n > 0$ in case of minimality!

Theorem

The infinite controllability/observability Gramians P/Q satisfy the Lyapunov equations

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

Systems and Control Theory
Balanced Realizations

Theorem

The infinite controllability/observability Gramians P/Q satisfy the Lyapunov equations

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

Proof. (For controllability Gramian only, observability case is analogous!)

$$\begin{aligned} AP + PA^T + BB^T &= A\int_0^\infty e^{At}BB^Te^{A^Tt}dt + \int_0^\infty e^{At}BB^Te^{A^Tt}dt\,A^T + BB^T \\ &= \int_0^\infty \underbrace{Ae^{At}BB^Te^{A^Tt} + e^{At}BB^Te^{A^Tt}A^T}_{=\frac{d}{dt}e^{At}BB^Te^{A^Tt}}dt + BB^T \\ &= \underbrace{\lim_{t\to\infty} e^{At}BB^Te^{A^Tt}}_{=0} - \underbrace{e^{A\cdot 0}}_{=I_n}BB^T\underbrace{e^{A^T\cdot 0}}_{=I_n} + BB^T \\ &= 0. \end{aligned}$$

Systems and Control Theory

Balanced Realizations

Definition

A realization (A, B, C, D) of a stable linear system Σ is balanced if its infinite controllability/observability Gramians P/Q satisfy

$$P = Q = \operatorname{diag}\{\sigma_1, \ldots, \sigma_n\} \quad (\text{w.l.o.g. } \sigma_j \geq \sigma_{j+1},\ j = 1, \ldots, n-1).$$

$\sigma_1, \ldots, \sigma_n$ are the Hankel singular values of Σ.

Note: $\sigma_1, \ldots, \sigma_n \geq 0$ as $P, Q \geq 0$ by definition, and $\sigma_1, \ldots, \sigma_n > 0$ in case of minimality!

Theorem

The Hankel singular values (HSVs) of a stable minimal linear system are system invariants, i.e. they are unaltered by state-space transformations!

Systems and Control Theory
Balanced Realizations

Theorem

The Hankel singular values (HSVs) of a stable minimal linear system are system invariants, i.e. they are unaltered by state-space transformations!

Proof. In balanced coordinates, the HSVs are $\Lambda(PQ)^{\frac{1}{2}}$. Now let

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

be any transformed realization with associated controllability Lyapunov equation

$$0 = \hat{A}\hat{P} + \hat{P}\hat{A}^T + \hat{B}\hat{B}^T = TAT^{-1}\hat{P} + \hat{P}T^{-T}A^TT^T + TBB^TT^T.$$

This is equivalent to

$$0 = A(T^{-1}\hat{P}T^{-T}) + (T^{-1}\hat{P}T^{-T})A^T + BB^T.$$

The uniqueness of the solution of the Lyapunov equation implies that $\hat{P} = TPT^T$ and, analogously, $\hat{Q} = T^{-T}QT^{-1}$. Therefore,

$$\hat{P}\hat{Q} = TPQT^{-1},$$

showing that $\Lambda(\hat{P}\hat{Q}) = \Lambda(PQ) = \{\sigma_1^2, \ldots, \sigma_n^2\}$.

Systems and Control Theory
Balanced Realizations

Definition

A realization (A, B, C, D) of a stable linear system Σ is balanced if its infinite controllability/observability Gramians P/Q satisfy

$$P = Q = \operatorname{diag}\{\sigma_1, \ldots, \sigma_n\} \quad \text{(w.l.o.g. } \sigma_j \geq \sigma_{j+1},\ j = 1, \ldots, n-1).$$

$\sigma_1, \ldots, \sigma_n$ are the Hankel singular values of Σ.

Note: $\sigma_1, \ldots, \sigma_n \geq 0$ as $P, Q \geq 0$ by definition, and $\sigma_1, \ldots, \sigma_n > 0$ in case of minimality!

Remark

For non-minimal systems, the Gramians can also be transformed into diagonal matrices with the leading $\hat{n} \times \hat{n}$ submatrices equal to $\operatorname{diag}(\sigma_1, \ldots, \sigma_{\hat{n}})$, and

$$\hat{P}\hat{Q} = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_{\hat{n}}^2, 0, \ldots, 0).$$

see [Laub/Heath/Paige/Ward 1987, Tombs/Postlethwaite 1987].

Qualitative and Quantitative Study of the Approximation Error
System Norms

Consider transfer function

$$G(s) = C\,(sI - A)^{-1}\,B + D$$

and input functions $u \in \mathcal{L}_2^m \cong L_2^m(-\infty, \infty)$, with the L_2-norm

$$\|u\|_2^2 := \frac{1}{2\pi}\int_{-\infty}^{\infty} u(\jmath\omega)^H u(\jmath\omega)\, d\omega.$$

Assume A (asympotically) stable: $\Lambda\,(A) \subset \mathbb{C}^- := \{z \in \mathbb{C} \,:\, \operatorname{re} z < 0\}$.

Qualitative and Quantitative Study of the Approximation Error
System Norms

Consider transfer function

$$G(s) = C\,(sI - A)^{-1}\,B + D$$

and input functions $u \in \mathcal{L}_2^m \cong L_2^m(-\infty, \infty)$, with the L_2-norm

$$\|u\|_2^2 := \frac{1}{2\pi}\int_{-\infty}^{\infty} u(\jmath\omega)^H u(\jmath\omega)\,d\omega.$$

Assume A (asympotically) stable: $\Lambda\,(A) \subset \mathbb{C}^- := \{z \in \mathbb{C} \,:\, \operatorname{re} z < 0\}$.
Then for all $s \in \mathbb{C}^+ \cup \jmath\mathbb{R}$, $\|G(s)\| \leq M < \infty \Rightarrow$

$$\int_{-\infty}^{\infty} y(\jmath\omega)^H y(\jmath\omega)\,d\omega \quad = \quad \int_{-\infty}^{\infty} u(\jmath\omega)^H G(\jmath\omega)^H G(\jmath\omega) u(\jmath\omega)\,d\omega$$

(Here, $\|\,.\,\|$ denotes the Euclidian vector or spectral matrix norm.)

Qualitative and Quantitative Study of the Approximation Error
System Norms

Consider transfer function

$$G(s) = C\,(sI - A)^{-1}\,B + D$$

and input functions $u \in \mathcal{L}_2^m \cong L_2^m(-\infty,\infty)$, with the L_2-norm

$$\|u\|_2^2 := \frac{1}{2\pi}\int_{-\infty}^{\infty} u(\jmath\omega)^H u(\jmath\omega)\,d\omega.$$

Assume A (asympotically) stable: $\Lambda\,(A) \subset \mathbb{C}^- := \{z \in \mathbb{C} \,:\, \operatorname{re} z < 0\}$.
Then for all $s \in \mathbb{C}^+ \cup \jmath\mathbb{R}$, $\|G(s)\| \leq M < \infty \Rightarrow$

$$\begin{aligned}\int_{-\infty}^{\infty} y(\jmath\omega)^H y(\jmath\omega)\,d\omega &= \int_{-\infty}^{\infty} u(\jmath\omega)^H G(\jmath\omega)^H G(\jmath\omega) u(\jmath\omega)\,d\omega \\ &= \int_{-\infty}^{\infty} \|G(\jmath\omega)u(\jmath\omega)\|^2\,d\omega \leq \int_{-\infty}^{\infty} M^2 \|u(\jmath\omega)\|^2\,d\omega\end{aligned}$$

(Here, $\|\,.\,\|$ denotes the Euclidian vector or spectral matrix norm.)

Qualitative and Quantitative Study of the Approximation Error

System Norms

Consider transfer function

$$G(s) = C\,(sI - A)^{-1}\,B + D$$

and input functions $u \in \mathcal{L}_2^m \cong L_2^m(-\infty, \infty)$, with the L_2-norm

$$\|u\|_2^2 := \frac{1}{2\pi} \int_{-\infty}^{\infty} u(\jmath\omega)^H u(\jmath\omega)\, d\omega.$$

Assume A (asympotically) stable: $\Lambda\,(A) \subset \mathbb{C}^- := \{z \in \mathbb{C} \,:\, \operatorname{re} z < 0\}$.
Then for all $s \in \mathbb{C}^+ \cup \jmath\mathbb{R}$, $\|G(s)\| \leq M < \infty \Rightarrow$

$$\begin{aligned}
\int_{-\infty}^{\infty} y(\jmath\omega)^H y(\jmath\omega)\, d\omega &= \int_{-\infty}^{\infty} u(\jmath\omega)^H G(\jmath\omega)^H G(\jmath\omega) u(\jmath\omega)\, d\omega \\
&= \int_{-\infty}^{\infty} \|G(\jmath\omega) u(\jmath\omega)\|^2\, d\omega \leq \int_{-\infty}^{\infty} M^2 \|u(\jmath\omega)\|^2\, d\omega \\
&= M^2 \int_{-\infty}^{\infty} u(\jmath\omega)^H u(\jmath\omega)\, d\omega \;<\; \infty.
\end{aligned}$$

(Here, $\|\,.\,\|$ denotes the Euclidian vector or spectral matrix norm.)

Qualitative and Quantitative Study of the Approximation Error
System Norms

Consider transfer function

$$G(s) = C\,(sI - A)^{-1}\,B + D$$

and input functions $u \in \mathcal{L}_2^m \cong L_2^m(-\infty,\infty)$, with the L_2-norm

$$\|u\|_2^2 := \frac{1}{2\pi}\int_{-\infty}^{\infty} u(\jmath\omega)^H u(\jmath\omega)\,d\omega.$$

Assume A (asympotically) stable: $\Lambda\,(A) \subset \mathbb{C}^- := \{z \in \mathbb{C} \,:\, \operatorname{re} z < 0\}$.
Then for all $s \in \mathbb{C}^+ \cup \jmath\mathbb{R}$, $\|G(s)\| \le M < \infty \Rightarrow$

$$\begin{aligned}
\int_{-\infty}^{\infty} y(\jmath\omega)^H y(\jmath\omega)\,d\omega &= \int_{-\infty}^{\infty} u(\jmath\omega)^H G(\jmath\omega)^H G(\jmath\omega) u(\jmath\omega)\,d\omega \\
&= \int_{-\infty}^{\infty} \|G(\jmath\omega)u(\jmath\omega)\|^2\,d\omega \le \int_{-\infty}^{\infty} M^2 \|u(\jmath\omega)\|^2\,d\omega \\
&= M^2 \int_{-\infty}^{\infty} u(\jmath\omega)^H u(\jmath\omega)\,d\omega \;<\; \infty.
\end{aligned}$$

$$\Longrightarrow y \in \mathcal{L}_2^q \cong L_2^q(-\infty,\infty).$$

Qualitative and Quantitative Study of the Approximation Error
System Norms

Consider transfer function

$$G(s) = C\,(sI - A)^{-1}\,B + D$$

and input functions $u \in \mathcal{L}_2^m \cong L_2^m(-\infty,\infty)$, with the L_2-norm

$$\|u\|_2^2 := \frac{1}{2\pi}\int_{-\infty}^{\infty} u(\jmath\omega)^H u(\jmath\omega)\,d\omega.$$

Assume A (asympotically) stable: $\Lambda\,(A) \subset \mathbb{C}^- := \{z \in \mathbb{C} \,:\, \operatorname{re} z < 0\}$.
Consequently, the 2-induced operator norm

$$\|G\|_\infty := \sup_{\|u\|_2 \neq 0} \frac{\|Gu\|_2}{\|u\|_2}$$

is well defined. It can be shown that

$$\|G\|_\infty = \sup_{\omega\in\mathbb{R}} \|G(\jmath\omega)\| = \sup_{\omega\in\mathbb{R}} \sigma_{max}\,(G(\jmath\omega))\,.$$

Qualitative and Quantitative Study of the Approximation Error
System Norms

Consider transfer function

$$G(s) = C\,(sI - A)^{-1}\,B + D$$

and input functions $u \in \mathcal{L}_2^m \cong L_2^m(-\infty, \infty)$, with the L_2-norm

$$\|u\|_2^2 := \frac{1}{2\pi} \int_{-\infty}^{\infty} u(\jmath\omega)^H u(\jmath\omega)\, d\omega.$$

Assume A (asympotically) stable: $\Lambda\,(A) \subset \mathbb{C}^- := \{z \in \mathbb{C} \,:\, \operatorname{re} z < 0\}$.
Consequently, the 2-induced operator norm

$$\|G\|_\infty := \sup_{\|u\|_2 \neq 0} \frac{\|Gu\|_2}{\|u\|_2}$$

is well defined. It can be shown that

$$\|G\|_\infty = \sup_{\omega\in\mathbb{R}} \|G(\jmath\omega)\| = \sup_{\omega\in\mathbb{R}} \sigma_{max}\,(G(\jmath\omega))\,.$$

Sketch of proof:
$\|G(\jmath\omega)u(\jmath\omega)\| \leq \|G(\jmath\omega)\|\|u(\jmath\omega)\| \Rightarrow$ "$\leq$".
Construct u with $\|Gu\|_2 = \sup_{\omega\in\mathbb{R}} \|G(\jmath\omega)\|\|u\|_2$.

Qualitative and Quantitative Study of the Approximation Error

System Norms

Consider transfer function

$$G(s) = C\,(sI - A)^{-1}\,B + D.$$

Hardy space $\mathcal{H}_\infty$

Function space of matrix-/scalar-valued functions that are analytic and bounded in $\mathbb{C}^+$.

The $\mathcal{H}_\infty$-norm is

$$\|F\|_\infty := \sup_{\mathrm{re}\, s>0} \sigma_{\max}\left(F(s)\right) = \sup_{\omega\in\mathbb{R}} \sigma_{max}\left(F(\jmath\omega)\right).$$

Stable transfer functions are in the Hardy spaces

- $\mathcal{H}_\infty$ in the SISO case (single-input, single-output, $m = q = 1$);
- $\mathcal{H}_\infty^{q\times m}$ in the MIMO case (multi-input, multi-output, $m > 1, q > 1$).

Qualitative and Quantitative Study of the Approximation Error

System Norms

Consider transfer function

$$G(s) = C\,(sI - A)^{-1}\,B + D.$$

Paley-Wiener Theorem (Parseval's equation/Plancherel Theorem)

$$L_2(-\infty,\infty) \cong \mathcal{L}_2, \quad L_2(0,\infty) \cong \mathcal{H}_2$$

Consequently, 2-norms in time and frequency domains coincide!

Qualitative and Quantitative Study of the Approximation Error
System Norms

Consider transfer function

$$G(s) = C\,(sI - A)^{-1}\,B + D.$$

Paley-Wiener Theorem (Parseval's equation/Plancherel Theorem)

$$L_2(-\infty,\infty) \cong \mathcal{L}_2, \quad L_2(0,\infty) \cong \mathcal{H}_2$$

Consequently, 2-norms in time and frequency domains coincide!

$\mathcal{H}_\infty$ approximation error

Reduced-order model $\Rightarrow$ transfer function $\hat{G}(s) = \hat{C}(sI_r - \hat{A})^{-1}\hat{B} + \hat{D}$.

$$\|y - \hat{y}\|_2 = \|Gu - \hat{G}u\|_2 \leq \|G - \hat{G}\|_\infty \|u\|_2.$$

$\Longrightarrow$ compute reduced-order model such that $\|G - \hat{G}\|_\infty < tol$!

Note: error bound holds in time- and frequency domain due to Paley-Wiener!

Qualitative and Quantitative Study of the Approximation Error
System Norms

Consider stable transfer function

$$G(s) = C\,(sI - A)^{-1}\,B, \quad \text{i.e. } D = 0.$$

Hardy space $\mathcal{H}_2$

Function space of matrix-/scalar-valued functions that are analytic $\mathbb{C}^+$ and bounded w.r.t. the $\mathcal{H}_2$-norm

$$\begin{aligned}\|F\|_2 &:= \frac{1}{2\pi}\left(\sup_{\operatorname{re}\sigma>0}\int_{-\infty}^{\infty}\|F(\sigma+\jmath\omega)\|_F^2\,d\omega\right)^{\frac{1}{2}}\\ &= \frac{1}{2\pi}\left(\int_{-\infty}^{\infty}\|F(\jmath\omega)\|_F^2\,d\omega\right)^{\frac{1}{2}}.\end{aligned}$$

Stable transfer functions are in the Hardy spaces

- $\mathcal{H}_2$ in the SISO case (single-input, single-output, $m = q = 1$);
- $\mathcal{H}_2^{q\times m}$ in the MIMO case (multi-input, multi-output, $m > 1, q > 1$).

Qualitative and Quantitative Study of the Approximation Error
System Norms

Consider stable transfer function

$$G(s) = C\,(sI - A)^{-1}\,B, \quad \text{i.e. } D = 0.$$

Hardy space $\mathcal{H}_2$

Function space of matrix-/scalar-valued functions that are analytic $\mathbb{C}^+$ and bounded w.r.t. the $\mathcal{H}_2$-norm

$$\|F\|_2 \;=\; \frac{1}{2\pi}\left(\int_{-\infty}^{\infty} \|F(\jmath\omega)\|_F^2\, d\omega\right)^{\frac{1}{2}}.$$

$\mathcal{H}_2$ approximation error for impulse response $(u(t) = u_0\delta(t))$

Reduced-order model $\Rightarrow$ transfer function $\hat{G}(s) = \hat{C}(sI_r - \hat{A})^{-1}\hat{B}$.

$$\|y - \hat{y}\|_2 = \|Gu_0\delta - \hat{G}u_0\delta\|_2 \le \|G - \hat{G}\|_2\|u_0\|.$$

$\Longrightarrow$ compute reduced-order model such that $\|G - \hat{G}\|_2 < tol$!

Qualitative and Quantitative Study of the Approximation Error

System Norms

Consider stable transfer function

$$G(s) = C\,(sI - A)^{-1}\,B, \quad \text{i.e. } D = 0.$$

Hardy space $\mathcal{H}_2$

Function space of matrix-/scalar-valued functions that are analytic $\mathbb{C}^+$ and bounded w.r.t. the $\mathcal{H}_2$-norm

$$\|F\|_2 \quad = \quad \frac{1}{2\pi}\left(\int_{-\infty}^{\infty} \|F(\jmath\omega)\|_F^2 \, d\omega\right)^{\frac{1}{2}}.$$

Theorem (Practical Computation of the $\mathcal{H}_2$-norm)

$$\|F\|_2^2 = \operatorname{tr}\left(B^T Q B\right) = \operatorname{tr}\left(C P C^T\right),$$

where P, Q are the controllability and observability Gramians of the corresponding LTI system.

Qualitative and Quantitative Study of the Approximation Error

Approximation Problems

Output errors in time-domain

$$\begin{aligned} \|y-\hat{y}\|_2 &\leq \|G-\hat{G}\|_\infty \|u\|_2 \qquad \Longrightarrow \|G-\hat{G}\|_\infty < \mathrm{tol} \\ \|y-\hat{y}\|_\infty &\leq \|G-\hat{G}\|_2 \|u\|_2 \qquad \Longrightarrow \|G-\hat{G}\|_2 < \mathrm{tol} \end{aligned}$$

Qualitative and Quantitative Study of the Approximation Error

Approximation Problems

Output errors in time-domain

$$\begin{aligned} \|y-\hat{y}\|_2 &\leq \|G-\hat{G}\|_\infty \|u\|_2 & \Longrightarrow \|G-\hat{G}\|_\infty < \mathrm{tol} \\ \|y-\hat{y}\|_\infty &\leq \|G-\hat{G}\|_2 \|u\|_2 & \Longrightarrow \|G-\hat{G}\|_2 < \mathrm{tol} \end{aligned}$$

$\mathcal{H}_\infty$-norm	best approximation problem for given reduced order r in general open; balanced truncation yields suboptimal solution with computable $\mathcal{H}_\infty$-norm bound.
$\mathcal{H}_2$-norm	necessary conditions for best approximation known; (local) optimizer computable with iterative rational Krylov algorithm (IRKA)
Hankel-norm $\|G\|_H := \sigma_{\max}$	optimal Hankel norm approximation (AAK theory).

Qualitative and Quantitative Study of the Approximation Error

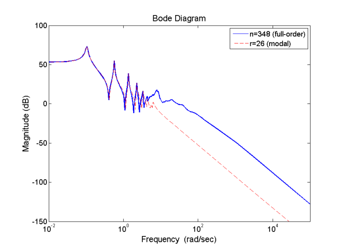
Computable error measures

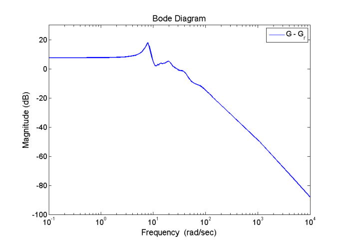
Evaluating system norms is computationally very (sometimes too) expensive.

Other measures

- absolute errors $\|G(\jmath\omega_j) - \hat{G}(\jmath\omega_j)\|_2$, $\|G(\jmath\omega_j) - \hat{G}(\jmath\omega_j)\|_\infty$ $(j = 1, \ldots, N_\omega)$;
- relative errors $\frac{\|G(\jmath\omega_j) - \hat{G}(\jmath\omega_j)\|_2}{\|G(\jmath\omega_j)\|_2}$, $\frac{\|G(\jmath\omega_j) - \hat{G}(\jmath\omega_j)\|_\infty}{\|G(\jmath\omega_j)\|_\infty}$;
- "eyeball norm", i.e. look at frequency response/Bode (magnitude) plot: for SISO system, log-log plot frequency vs. $|G(\jmath\omega)|$ (or $|G(\jmath\omega) - \hat{G}(\jmath\omega)|$) in decibels, 1 dB $\simeq 20\log_{10}(\text{value})$.

  For MIMO systems, $q \times m$ array of plots G_{ij}.





Outline

Model Reduction by Projection

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 $\mathbb{C}^-$),
  - 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").

Model Reduction by Projection

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 $\mathbb{C}^-$),
  - 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"*).

Model Reduction by Projection

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 $\mathbb{C}^-$),
  - 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").

Model Reduction by Projection

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 $\mathbb{C}^-$),
  - 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").

Model Reduction by Projection

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 $\mathbb{C}^-$),
  - 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").

Model Reduction by Projection

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 $\mathbb{C}^-$),
  - 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").

Model Reduction by Projection

Projection Basics

Definition 3.1 (Projector)

A projector is a matrix $P \in \mathbb{R}^{n\times n}$ with $P^2 = P$. Let $\mathcal{V} = \operatorname{range}(P)$, then P is projector onto $\mathcal{V}$. On the other hand, if $\{v_1, \ldots, v_r\}$ is a basis of $\mathcal{V}$ and $V = [\, v_1, \ldots, v_r\,]$, then $P = V(V^TV)^{-1}V^T$ is a projector onto $\mathcal{V}$.

Model Reduction by Projection

Projection Basics

Definition 3.1 (Projector)

A projector is a matrix $P \in \mathbb{R}^{n \times n}$ with $P^2 = P$. Let $\mathcal{V} = \mathrm{range}\,(P)$, then P is projector onto $\mathcal{V}$. On the other hand, if $\{v_1, \ldots, v_r\}$ is a basis of $\mathcal{V}$ and $V = [\, v_1, \ldots, v_r \,]$, then $P = V(V^T V)^{-1} V^T$ is a projector onto $\mathcal{V}$.

Lemma 3.2 (Projector Properties)

- If $P = P^T$, then P is an orthogonal projector (aka: Galerkin projection), otherwise an oblique projector (aka: Petrov-Galerkin projection).
- P is the identity operator on $\mathcal{V}$, i.e., $Pv = v \;\; \forall v \in \mathcal{V}$.
- $I - P$ is the complementary projector onto $\ker P$.
- If $\mathcal{V}$ is an A-invariant subspace corresponding to a subset of A's spectrum, then we call P a spectral projector.
- Let $\mathcal{W} \subset \mathbb{R}^n$ be another r-dimensional subspace and $W = [\, w_1, \ldots, w_r \,]$ be a basis matrix for $\mathcal{W}$, then $P = V(W^T V)^{-1} W^T$ is an oblique projector onto $\mathcal{V}$ along $\mathcal{W}$.

Model Reduction by Projection

Projection Basics

Definition 3.1 (Projector)

A projector is a matrix $P \in \mathbb{R}^{n \times n}$ with $P^2 = P$. Let $\mathcal{V} = \operatorname{range}(P)$, then P is projector onto $\mathcal{V}$. On the other hand, if $\{v_1, \ldots, v_r\}$ is a basis of $\mathcal{V}$ and $V = [\, v_1, \ldots, v_r \,]$, then $P = V(V^T V)^{-1} V^T$ is a projector onto $\mathcal{V}$.

Lemma 3.2 (Projector Properties)

- If $P = P^T$, then P is an orthogonal projector (aka: Galerkin projection), otherwise an oblique projector (aka: Petrov-Galerkin projection).
- P is the identity operator on $\mathcal{V}$, i.e., $Pv = v \;\; \forall v \in \mathcal{V}$.
- $I - P$ is the complementary projector onto $\ker P$.
- If $\mathcal{V}$ is an A-invariant subspace corresponding to a subset of A's spectrum, then we call P a spectral projector.
- Let $\mathcal{W} \subset \mathbb{R}^n$ be another r-dimensional subspace and $W = [\, w_1, \ldots, w_r \,]$ be a basis matrix for $\mathcal{W}$, then $P = V(W^T V)^{-1} W^T$ is an oblique projector onto $\mathcal{V}$ along $\mathcal{W}$.

Model Reduction by Projection

Projection Basics

Definition 3.1 (Projector)

A projector is a matrix $P \in \mathbb{R}^{n \times n}$ with $P^2 = P$. Let $\mathcal{V} = \operatorname{range}(P)$, then P is projector onto $\mathcal{V}$. On the other hand, if $\{v_1, \ldots, v_r\}$ is a basis of $\mathcal{V}$ and $V = [\, v_1, \ldots, v_r \,]$, then $P = V(V^T V)^{-1} V^T$ is a projector onto $\mathcal{V}$.

Lemma 3.2 (Projector Properties)

- If $P = P^T$, then P is an orthogonal projector (aka: Galerkin projection), otherwise an oblique projector (aka: Petrov-Galerkin projection).
- P is the identity operator on $\mathcal{V}$, i.e., $Pv = v \;\; \forall v \in \mathcal{V}$.
- $I - P$ is the complementary projector onto $\ker P$.
- If $\mathcal{V}$ is an A-invariant subspace corresponding to a subset of A's spectrum, then we call P a spectral projector.
- Let $\mathcal{W} \subset \mathbb{R}^n$ be another r-dimensional subspace and $W = [\, w_1, \ldots, w_r \,]$ be a basis matrix for $\mathcal{W}$, then $P = V(W^T V)^{-1} W^T$ is an oblique projector onto $\mathcal{V}$ along $\mathcal{W}$.

Model Reduction by Projection

Projection Basics

Definition 3.1 (Projector)

A projector is a matrix $P \in \mathbb{R}^{n \times n}$ with $P^2 = P$. Let $\mathcal{V} = \mathrm{range}\,(P)$, then P is projector onto $\mathcal{V}$. On the other hand, if $\{v_1, \ldots, v_r\}$ is a basis of $\mathcal{V}$ and $V = [\, v_1, \ldots, v_r \,]$, then $P = V(V^T V)^{-1} V^T$ is a projector onto $\mathcal{V}$.

Lemma 3.2 (Projector Properties)

- If $P = P^T$, then P is an orthogonal projector (aka: Galerkin projection), otherwise an oblique projector (aka: Petrov-Galerkin projection).
- P is the identity operator on $\mathcal{V}$, i.e., $Pv = v \;\; \forall v \in \mathcal{V}$.
- $I - P$ is the complementary projector onto $\ker P$.
- If $\mathcal{V}$ is an A-invariant subspace corresponding to a subset of A's spectrum, then we call P a spectral projector.
- Let $\mathcal{W} \subset \mathbb{R}^n$ be another r-dimensional subspace and $W = [\, w_1, \ldots, w_r \,]$ be a basis matrix for $\mathcal{W}$, then $P = V(W^T V)^{-1} W^T$ is an oblique projector onto $\mathcal{V}$ along $\mathcal{W}$.

Model Reduction by Projection

Projection Basics

Definition 3.1 (Projector)

A projector is a matrix $P \in \mathbb{R}^{n\times n}$ with $P^2 = P$. Let $\mathcal{V} = \mathrm{range}\,(P)$, then P is projector onto $\mathcal{V}$. On the other hand, if $\{v_1, \ldots, v_r\}$ is a basis of $\mathcal{V}$ and $V = [\,v_1, \ldots, v_r\,]$, then $P = V(V^T V)^{-1} V^T$ is a projector onto $\mathcal{V}$.

Lemma 3.2 (Projector Properties)

- If $P = P^T$, then P is an orthogonal projector (aka: Galerkin projection), otherwise an oblique projector (aka: Petrov-Galerkin projection).
- P is the identity operator on $\mathcal{V}$, i.e., $Pv = v \;\; \forall v \in \mathcal{V}$.
- $I - P$ is the complementary projector onto $\ker P$.
- If $\mathcal{V}$ is an A-invariant subspace corresponding to a subset of A's spectrum, then we call P a spectral projector.
- Let $\mathcal{W} \subset \mathbb{R}^n$ be another r-dimensional subspace and $W = [\,w_1, \ldots, w_r\,]$ be a basis matrix for $\mathcal{W}$, then $P = V(W^T V)^{-1} W^T$ is an oblique projector onto $\mathcal{V}$ along $\mathcal{W}$.

Model Reduction by Projection

Projection and Interpolation

Methods:

1. Modal Truncation
2. Rational Interpolation (Padé-Approximation and (rational) Krylov Subspace Methods)
3. Balanced Truncation
4. many more...

Joint feature of these methods:
computation of reduced-order model (ROM) by projection!

Model Reduction by Projection

Projection and Interpolation

Joint feature of these methods:
computation of reduced-order model (ROM) by projection!

Assume trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V}$. Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ along complementary subspace $\mathcal{W}$: $x \approx VW^T x =: \tilde{x}$, where

$$\mathrm{range}\,(V) = \mathcal{V}, \quad \mathrm{range}\,(W) = \mathcal{W}, \quad W^T V = I_r.$$

Then, with $\hat{x} = W^T x$, we obtain $x \approx V\hat{x}$ so that

$$\|x - \tilde{x}\| = \|x - V\hat{x}\|,$$

and the reduced-order model is

$$\hat{A} := W^T AV, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

Model Reduction by Projection

Projection and Interpolation

Joint feature of these methods:
computation of reduced-order model (ROM) by projection!

Assume trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V}$. Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ along complementary subspace $\mathcal{W}$: $x \approx VW^T x =: \tilde{x}$, and the reduced-order model is $\hat{x} = W^T x$

$$\hat{A} := W^T AV, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

Important observation:

- The state equation residual satisfies $\dot{\tilde{x}} - A\tilde{x} - Bu \perp \mathcal{W}$, since

$$W^T\left(\dot{\tilde{x}} - A\tilde{x} - Bu\right) = W^T\left(VW^T\dot{x} - AVW^T x - Bu\right)$$

Model Reduction by Projection

Projection and Interpolation

Joint feature of these methods:
computation of reduced-order model (ROM) by projection!

Assume trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V}$. Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ along complementary subspace $\mathcal{W}$: $x \approx VW^T x =: \tilde{x}$, and the reduced-order model is $\hat{x} = W^T x$

$$\hat{A} := W^T AV, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

Important observation:

- The state equation residual satisfies $\dot{\tilde{x}} - A\tilde{x} - Bu \perp \mathcal{W}$, since

$$\begin{aligned} W^T\left(\dot{\tilde{x}} - A\tilde{x} - Bu\right) &= W^T\left(VW^T\dot{x} - AVW^Tx - Bu\right) \\ &= \underbrace{W^T\dot{x}}_{\dot{\hat{x}}} - \underbrace{W^TAV}_{=\hat{A}}\underbrace{W^Tx}_{=\hat{x}} - \underbrace{W^TB}_{=\hat{B}}u \end{aligned}$$

Model Reduction by Projection

Projection and Interpolation

Joint feature of these methods:
computation of reduced-order model (ROM) by projection!

Assume trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V}$. Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ along complementary subspace $\mathcal{W}$: $x \approx VW^T x =: \tilde{x}$, and the reduced-order model is $\hat{x} = W^T x$

$$\hat{A} := W^T AV, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

Important observation:

- The state equation residual satisfies $\dot{\tilde{x}} - A\tilde{x} - Bu \perp \mathcal{W}$, since

$$\begin{aligned} W^T\left(\dot{\tilde{x}} - A\tilde{x} - Bu\right) &= W^T\left(VW^T\dot{x} - AVW^T x - Bu\right) \\ &= \underbrace{W^T\dot{x}}_{\dot{\hat{x}}} - \underbrace{W^T AV}_{=\hat{A}}\underbrace{W^T x}_{=\hat{x}} - \underbrace{W^T B}_{=\hat{B}} u \\ &= \dot{\hat{x}} - \hat{A}\hat{x} - \hat{B}u = 0. \end{aligned}$$

Model Reduction by Projection

Projection and Interpolation

Projection ⇝ Rational Interpolation

Given the ROM

$$\hat{A} = W^T AV, \quad \hat{B} = W^T B, \quad \hat{C} = CV, \quad (\hat{D} = D),$$

the error transfer function can be written as

$$G(s) - \hat{G}(s) \quad = \quad \left(C(sI_n - A)^{-1}B + D \right) - \left(\hat{C}(sI_r - \hat{A})^{-1}\hat{B} + \hat{D} \right)$$

Model Reduction by Projection

Projection and Interpolation

Projection $\rightsquigarrow$ Rational Interpolation

Given the ROM

$$\hat{A} = W^T AV, \quad \hat{B} = W^T B, \quad \hat{C} = CV, \quad (\hat{D} = D),$$

the error transfer function can be written as

$$\begin{aligned} G(s) - \hat{G}(s) &= \left(C(sI_n - A)^{-1}B + D\right) - \left(\hat{C}(sI_r - \hat{A})^{-1}\hat{B} + \hat{D}\right) \\ &= C\left((sI_n - A)^{-1} - V(sI_r - \hat{A})^{-1}W^T\right)B \end{aligned}$$

Model Reduction by Projection

Projection and Interpolation

Projection $\rightsquigarrow$ Rational Interpolation

Given the ROM

$$\hat{A} = W^T AV, \quad \hat{B} = W^T B, \quad \hat{C} = CV, \quad (\hat{D} = D),$$

the error transfer function can be written as

$$\begin{aligned} G(s) - \hat{G}(s) &= \left(C(sI_n - A)^{-1}B + D\right) - \left(\hat{C}(sI_r - \hat{A})^{-1}\hat{B} + \hat{D}\right) \\ &= C\left((sI_n - A)^{-1} - V(sI_r - \hat{A})^{-1}W^T\right)B \\ &= C\big(I_n - \underbrace{V(sI_r - \hat{A})^{-1}W^T(sI_n - A)}_{=:P(s)}\big)(sI_n - A)^{-1}B. \end{aligned}$$

Model Reduction by Projection

Projection and Interpolation

Projection ⇝ Rational Interpolation

Given the ROM

$$\hat{A} = W^T AV, \quad \hat{B} = W^T B, \quad \hat{C} = CV, \quad (\hat{D} = D),$$

the error transfer function can be written as

$$\begin{aligned} G(s) - \hat{G}(s) &= \left(C(sI_n - A)^{-1}B + D\right) - \left(\hat{C}(sI_r - \hat{A})^{-1}\hat{B} + \hat{D}\right) \\ &= C\Big(I_n - \underbrace{V(sI_r - \hat{A})^{-1}W^T(sI_n - A)}_{=:P(s)}\Big)(sI_n - A)^{-1}B. \end{aligned}$$

If $s_* \in \mathbb{C} \setminus (\Lambda(A) \cup \Lambda(\hat{A}))$, then $P(s_*)$ is a projector onto $\mathcal{V}$:

$\mathrm{range}\,(P(s_*)) \subset \mathrm{range}\,(V)$, all matrices have full rank $\Rightarrow$ "=",

$$P(s_*)^2 = V(s_* I_r - \hat{A})^{-1}W^T(s_* I_n - A)V(s_* I_r - \hat{A})^{-1}W^T(s_* I_n - A)$$

Model Reduction by Projection

Projection and Interpolation

Projection ⇝ Rational Interpolation

Given the ROM

$$\hat{A} = W^T AV, \quad \hat{B} = W^T B, \quad \hat{C} = CV, \quad (\hat{D} = D),$$

the error transfer function can be written as

$$\begin{aligned} G(s) - \hat{G}(s) &= \Big(C(sI_n - A)^{-1}B + D\Big) - \Big(\hat{C}(sI_r - \hat{A})^{-1}\hat{B} + \hat{D}\Big) \\ &= C\Big(I_n - \underbrace{V(sI_r - \hat{A})^{-1}W^T(sI_n - A)}_{=:P(s)}\Big)(sI_n - A)^{-1}B. \end{aligned}$$

If $s_* \in \mathbb{C} \setminus (\Lambda(A) \cup \Lambda(\hat{A}))$, then $P(s_*)$ is a projector onto $\mathcal{V}$:

$\mathrm{range}\,(P(s_*)) \subset \mathrm{range}\,(V)$, all matrices have full rank $\Rightarrow$ "$=$",

$$\begin{aligned} P(s_*)^2 &= V(s_*I_r - \hat{A})^{-1}W^T(s_*I_n - A)V(s_*I_r - \hat{A})^{-1}W^T(s_*I_n - A) \\ &= V(s_*I_r - \hat{A})^{-1}\underbrace{(s_*I_r - \hat{A})(s_*I_r - \hat{A})^{-1}}_{=I_r}W^T(s_*I_n - A) = P(s_*). \end{aligned}$$

Model Reduction by Projection

Projection and Interpolation

Projection ⇝ Rational Interpolation

Given the ROM

$$\hat{A} = W^T AV, \quad \hat{B} = W^T B, \quad \hat{C} = CV, \quad (\hat{D} = D),$$

the error transfer function can be written as

$$\begin{aligned} G(s) - \hat{G}(s) &= \Big(C(sI_n - A)^{-1}B + D\Big) - \Big(\hat{C}(sI_r - \hat{A})^{-1}\hat{B} + \hat{D}\Big) \\ &= C\big(I_n - \underbrace{V(sI_r - \hat{A})^{-1}W^T(sI_n - A)}_{=:P(s)}\big)(sI_n - A)^{-1}B. \end{aligned}$$

If $s_* \in \mathbb{C} \setminus (\Lambda(A) \cup \Lambda(\hat{A}))$, then $P(s_*)$ is a projector onto $\mathcal{V}$ $\Longrightarrow$

if $(s_* I_n - A)^{-1}B \in \mathcal{V}$, *then* $(I_n - P(s_*))(s_* I_n - A)^{-1}B = 0$,

hence

$$G(s_*) - \hat{G}(s_*) = 0 \;\Rightarrow\; G(s_*) = \hat{G}(s_*), \text{ i.e., } \hat{G} \text{ interpolates } G \text{ in } s_*!$$

Model Reduction by Projection

Projection and Interpolation

Projection ⇝ Rational Interpolation

Given the ROM

$$\hat{A} = W^T A V, \quad \hat{B} = W^T B, \quad \hat{C} = CV, \quad (\hat{D} = D),$$

the error transfer function can be written as

$$\begin{aligned} G(s) - \hat{G}(s) &= \Big(C(sI_n - A)^{-1}B + D\Big) - \Big(\hat{C}(sI_r - \hat{A})^{-1}\hat{B} + \hat{D}\Big) \\ &= C\Big(I_n - \underbrace{V(sI_r - \hat{A})^{-1}W^T(sI_n - A)}_{=:P(s)}\Big)(sI_n - A)^{-1}B. \end{aligned}$$

Analogously, $= C(sI_n - A)^{-1}\Big(I_n - \underbrace{(sI_n - A)V(sI_r - \hat{A})^{-1}W^T}_{=:Q(s)}\Big)B.$

If $s_* \in \mathbb{C} \setminus (\Lambda(A) \cup \Lambda(\hat{A}))$, then $Q(s)^H$ is a projector onto $\mathcal{W}$ $\Longrightarrow$

if $(s_* I_n - A)^{-*} C^T \in \mathcal{W}$, *then* $C(s_* I_n - A)^{-1}(I_n - Q(s_*)) = 0$,

hence

$$G(s_*) - \hat{G}(s_*) = 0 \;\Rightarrow\; G(s_*) = \hat{G}(s_*), \text{ i.e., } \hat{G} \text{ interpolates } G \text{ in } s_*!$$

Model Reduction by Projection

Projection and Interpolation

Theorem 3.3 [Grimme '97, Villemagne/Skelton '87]

Given the ROM

$$\hat{A} = W^T AV, \quad \hat{B} = W^T B, \quad \hat{C} = CV, \quad (\hat{D} = D),$$

and $s_* \in \mathbb{C} \setminus (\Lambda(A) \cup \Lambda(\hat{A}))$, if either

- $(s_* I_n - A)^{-1} B \in \mathrm{range}(V)$, or
- $(s_* I_n - A)^{-*} C^T \in \mathrm{range}(W)$,

then the interpolation condition

$$G(s_*) = \hat{G}(s_*).$$

in s_* holds.

Note: extension to Hermite interpolation conditions later!

Modal Truncation

Basic method:

Assume A is diagonalizable, $T^{-1}AT = D_A$, project state-space onto A-invariant subspace $\mathcal{V} = \mathrm{span}(t_1, \ldots, t_r)$, t_k = eigenvectors corresp. to "dominant" modes / eigenvalues of A. Then with

$$V = T(:, 1:r) = [\, t_1, \ldots, t_r \,], \quad \tilde{W}^H = T^{-1}(1:r, :), \quad W = \tilde{W}(V^H \tilde{W})^{-1},$$

reduced-order model is

$$\hat{A} := W^H A V = \mathrm{diag}\,\{\lambda_1, \ldots, \lambda_r\}, \quad \hat{B} := W^H B, \quad \hat{C} = CV$$

Also computable by truncation:

$$T^{-1}AT = \begin{bmatrix} \hat{A} & \\ & A_2 \end{bmatrix}, \quad T^{-1}B = \begin{bmatrix} \hat{B} \\ B_2 \end{bmatrix}, \quad CT = [\, \hat{C}, \; C_2 \,], \quad \hat{D} = D.$$

Modal Truncation

Basic method:

Assume A is diagonalizable, $T^{-1}AT = D_A$, project state-space onto A-invariant subspace $\mathcal{V} = \mathrm{span}(t_1, \ldots, t_r)$, $t_k =$ eigenvectors corresp. to "dominant" modes / eigenvalues of A. Then with

$$V = T(:, 1:r) = [\, t_1, \ldots, t_r \,], \quad \tilde{W}^H = T^{-1}(1:r, :), \quad W = \tilde{W}(V^H \tilde{W})^{-1},$$

reduced-order model is

$$\hat{A} := W^H A V = \mathrm{diag}\,\{\lambda_1, \ldots, \lambda_r\}, \quad \hat{B} := W^H B, \quad \hat{C} = CV$$

Also computable by truncation:

$$T^{-1}AT = \begin{bmatrix} \hat{A} & \\ & A_2 \end{bmatrix}, \quad T^{-1}B = \begin{bmatrix} \hat{B} \\ B_2 \end{bmatrix}, \quad CT = [\, \hat{C}, \; C_2 \,], \quad \hat{D} = D.$$

Properties:

Simple computation for large-scale systems, using, e.g., Krylov subspace methods (Lanczos, Arnoldi), Jacobi-Davidson method.

Modal Truncation

Basic method:

$$T^{-1}AT = \begin{bmatrix} \hat{A} & \\ & A_2 \end{bmatrix}, \quad T^{-1}B = \begin{bmatrix} \hat{B} \\ B_2 \end{bmatrix}, \quad CT = [\,\hat{C},\ C_2\,], \quad \hat{D} = D.$$

Properties:

Error bound:

$$\|G - \hat{G}\|_\infty \leq \|C_2\|\|B_2\| \frac{1}{\min_{\lambda \in \Lambda(A_2)} |\mathrm{Re}(\lambda)|}.$$

Proof:

$$\begin{aligned} G(s) &= C(sI - A)^{-1}B + D = CTT^{-1}(sI - A)^{-1}TT^{-1}B + D \\ &= CT(sI - T^{-1}AT)^{-1}T^{-1}B + D \\ &= [\,\hat{C},\ C_2\,] \begin{bmatrix} (sI_r - \hat{A})^{-1} & \\ & (sI_{n-r} - A_2)^{-1} \end{bmatrix} \begin{bmatrix} \hat{B} \\ B_2 \end{bmatrix} + D \\ &= \hat{G}(s) + C_2(sI_{n-r} - A_2)^{-1}B_2, \end{aligned}$$

Modal Truncation

Basic method:

$$T^{-1}AT = \begin{bmatrix} \hat{A} & \\ & A_2 \end{bmatrix}, \quad T^{-1}B = \begin{bmatrix} \hat{B} \\ B_2 \end{bmatrix}, \quad CT = [\,\hat{C},\ C_2\,], \quad \hat{D} = D.$$

Properties:

Error bound:

$$\|G - \hat{G}\|_\infty \le \|C_2\|\|B_2\| \frac{1}{\min_{\lambda\in\Lambda(A_2)} |\mathrm{Re}(\lambda)|}.$$

Proof:

$$G(s) \quad = \quad \hat{G}(s) + C_2(sI_{n-r} - A_2)^{-1}B_2,$$

observing that $\|G - \hat{G}\|_\infty = \sup_{\omega\in\mathbb{R}} \sigma_{\max}(C_2(\jmath\omega I_{n-r} - A_2)^{-1}B_2)$, and

$$C_2(\jmath\omega I_{n-r} - A_2)^{-1}B_2 = C_2\mathrm{diag}\left(\frac{1}{\jmath\omega - \lambda_{r+1}}, \ldots, \frac{1}{\jmath\omega - \lambda_n}\right) B_2.$$

Modal Truncation

Basic method:

Assume A is diagonalizable, $T^{-1}AT = D_A$, project state-space onto A-invariant subspace $\mathcal{V} = \mathrm{span}(t_1, \ldots, t_r)$, t_k = eigenvectors corresp. to "dominant" modes / eigenvalues of A. Then reduced-order model is

$$\hat{A} := W^H AV = \mathrm{diag}\{\lambda_1, \ldots, \lambda_r\}, \quad \hat{B} := W^H B, \quad \hat{C} = CV$$

Also computable by truncation:

$$T^{-1}AT = \begin{bmatrix} \hat{A} & \\ & A_2 \end{bmatrix}, \quad T^{-1}B = \begin{bmatrix} \hat{B} \\ B_2 \end{bmatrix}, \quad CT = [\,\hat{C},\ C_2\,], \quad \hat{D} = D.$$

Difficulties:

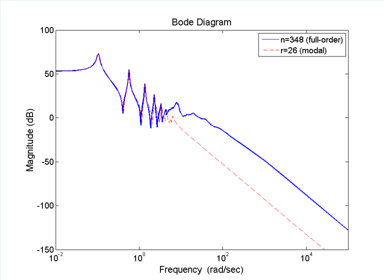
- Eigenvalues contain only limited system information.
- Dominance measures are difficult to compute.
  ([Litz '79] use Jordan canoncial form; otherwise merely heuristic criteria, e.g., [Varga '95]. Recent improvement: dominant pole algorithm.)
- Error bound not computable for really large-scale problems.

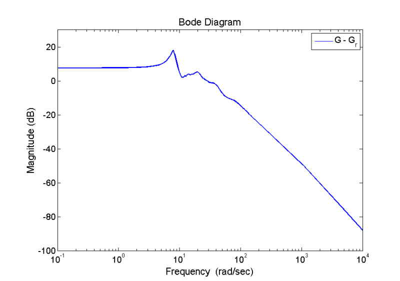
Modal Truncation

Example

BEAM, SISO system from SLICOT Benchmark Collection for Model Reduction, $n = 348$, $m = q = 1$, reduced using 13 dominant complex conjugate eigenpairs, error bound yields $\|G - \hat{G}\|_\infty \leq 1.21 \cdot 10^3$

Bode plots of transfer functions and error function





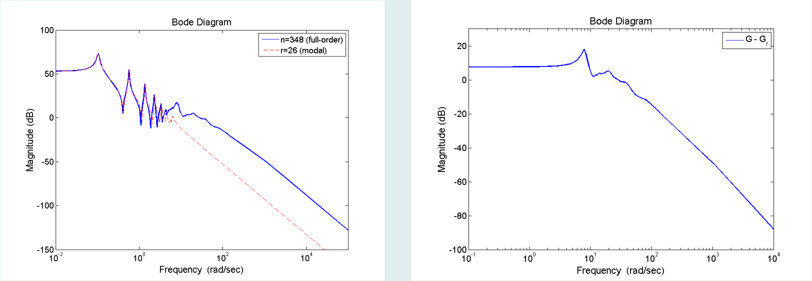
MATLAB® demo.

Modal Truncation

Example

BEAM, SISO system from SLICOT Benchmark Collection for Model Reduction, $n = 348$, $m = q = 1$, reduced using 13 dominant complex conjugate eigenpairs, error bound yields $\|G - \hat{G}\|_\infty \leq 1.21 \cdot 10^3$

Bode plots of transfer functions and error function



MATLAB® demo.

Modal Truncation

Extensions

Base enrichment

Static modes are defined by setting $\dot{x} = 0$ and assuming unit loads, i.e., $u(t) \equiv e_j$, $j = 1, \ldots, m$:

$$0 = Ax(t) + Be_j \quad \Longrightarrow \quad x(t) \equiv -A^{-1}b_j.$$

Projection subspace $\mathcal{V}$ is then augmented by $A^{-1}[\, b_1, \ldots, b_m\,] = A^{-1}B$.

Interpolation-projection framework $\Longrightarrow$ $G(0) = \hat{G}(0)$!

If two sided projection is used, complimentary subspace can be augmented by $A^{-T}C^T \Longrightarrow G'(0) = \hat{G}'(0)$! (If $m \neq q$, add random vectors or delete some of the columns in $A^{-T}C^T$).

Modal Truncation

Extensions

Guyan reduction (static condensation)

Partition states in masters $x_1 \in \mathbb{R}^r$ and slaves $x_2 \in \mathbb{R}^{n-r}$ (FEM terminology)
Assume stationarity, i.e., $\dot{x} = 0$ and solve for x_2 in

$$\begin{aligned} 0 &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ \Rightarrow \quad x_2 &= -A_{22}^{-1} A_{21} x_1 - A_{22}^{-1} B_2 u. \end{aligned}$$

Inserting this into the first part of the dynamic system

$$\dot{x}_1 = A_{11} x_1 + A_{12} x_2 + B_1 u, \quad y = C_1 x_1 + C_2 x_2$$

then yields the reduced-order model

$$\begin{aligned} \dot{x}_1 &= (A_{11} - A_{12} A_{22}^{-1} A_{21}) x_1 + (B_1 - A_{12} A_{22}^{-1} B_2) u \\ y &= (C_1 - C_2 A_{22}^{-1} A_{21}) x_1 - C_2 A_{22}^{-1} B_2 u. \end{aligned}$$

Modal Truncation

Dominant Poles

Pole-Residue Form of Transfer Function

Consider partial fraction expansion of transfer function with $D = 0$:

$$G(s) = \sum_{k=1}^{n} \frac{R_k}{s - \lambda_k}$$

with the residues $R_k := (Cx_k)(y_k^H B) \in \mathbb{C}^{q \times m}$.

Modal Truncation

Dominant Poles

Pole-Residue Form of Transfer Function

Consider partial fraction expansion of transfer function with $D = 0$:

$$G(s) = \sum_{k=1}^{n} \frac{R_k}{s - \lambda_k}$$

with the residues $R_k := (Cx_k)(y_k^H B) \in \mathbb{C}^{q \times m}$.

Note: this follows using the spectral decomposition $A = XDX^{-1}$, with $X = [\, x_1, \ldots, x_n]$ the right and $X^{-1} =: Y = [y_1, \ldots, y_n]^H$ the left eigenvector matrices:

$$\begin{aligned} G(s) &= C(sI - XDX^{-1})^{-1}B = CX(sI - \operatorname{diag}\{\lambda_1, \ldots, \lambda_n\})^{-1}YB \\ &= [\, Cx_1, \ldots, Cx_n\,] \begin{bmatrix} \frac{1}{s-\lambda_1} & & \\ & \ddots & \\ & & \frac{1}{s-\lambda_n} \end{bmatrix} \begin{bmatrix} y_1^H B \\ \vdots \\ y_n^H B \end{bmatrix} \\ &= \sum_{k=1}^{n} \frac{(Cx_k)(y_k^H B)}{s - \lambda_k}. \end{aligned}$$

Modal Truncation

Dominant Poles

Pole-Residue Form of Transfer Function

Consider partial fraction expansion of transfer function with $D = 0$:

$$G(s) = \sum_{k=1}^{n} \frac{R_k}{s - \lambda_k}$$

with the residues $R_k := (Cx_k)(y_k^H B) \in \mathbb{C}^{q \times m}$.

Note: $R_k = (Cx_k)(y_k^H B)$ are the residues of G in the sense of the residue theorem of complex analysis:

$$\mathrm{res}\,(G, \lambda_\ell) = \lim_{s \to \lambda_\ell} (s - \lambda_\ell) G(s) = \sum_{k=1}^{n} \underbrace{\lim_{s \to \lambda_\ell} \frac{s - \lambda_\ell}{s - \lambda_k}}_{= \begin{cases} 0 \text{ for } k \neq \ell \\ 1 \text{ for } k = \ell \end{cases}} R_k = R_\ell.$$

Modal Truncation

Dominant Poles

Pole-Residue Form of Transfer Function

Consider partial fraction expansion of transfer function with $D = 0$:

$$G(s) = \sum_{k=1}^{n} \frac{R_k}{s - \lambda_k}$$

with the residues $R_k := (Cx_k)(y_k^H B) \in \mathbb{C}^{q \times m}$.

As projection basis use spaces spanned by right/left eigenvectors corresponding to dominant poles, i.e.. $(\lambda_j,\ x_j,\ y_j)$ with largest

$$\|R_k\| / |\operatorname{re}(\lambda_k)|.$$

Modal Truncation

Dominant Poles

Pole-Residue Form of Transfer Function

Consider partial fraction expansion of transfer function with $D = 0$:

$$G(s) = \sum_{k=1}^{n} \frac{R_k}{s - \lambda_k}$$

with the residues $R_k := (Cx_k)(y_k^H B) \in \mathbb{C}^{q \times m}$.

As projection basis use spaces spanned by right/left eigenvectors corresponding to dominant poles, i.e.. $(\lambda_j,\ x_j,\ y_j)$ with largest

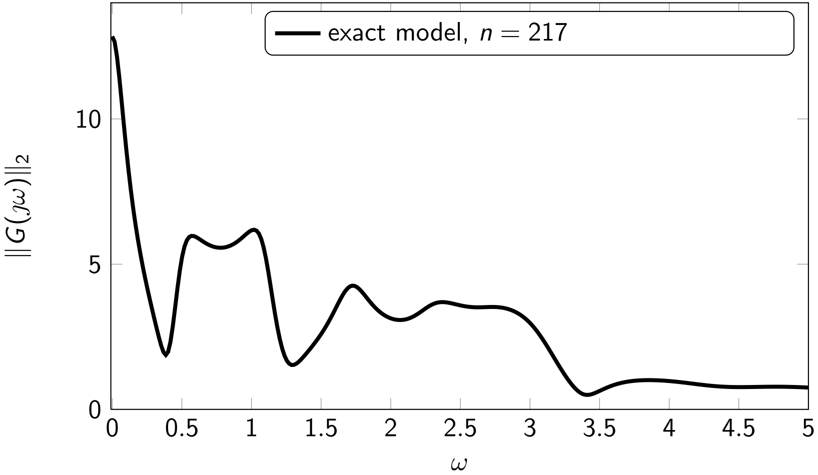
$$\|R_k\| / |\operatorname{re}(\lambda_k)|.$$

Remark

The dominant modes have most important influence on the input-output behavior of the system and are responsible for the "peaks"' in the frequency response.

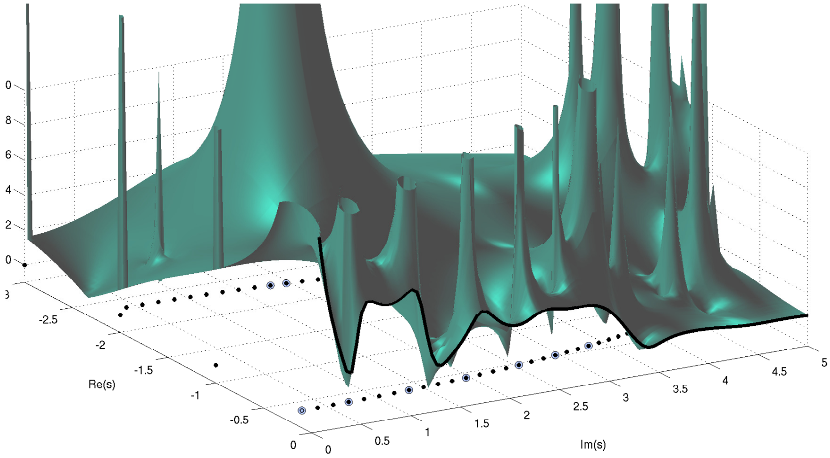
Dominant Poles

Random SISO Example ($B,\ C^T \in \mathbb{R}^n$)



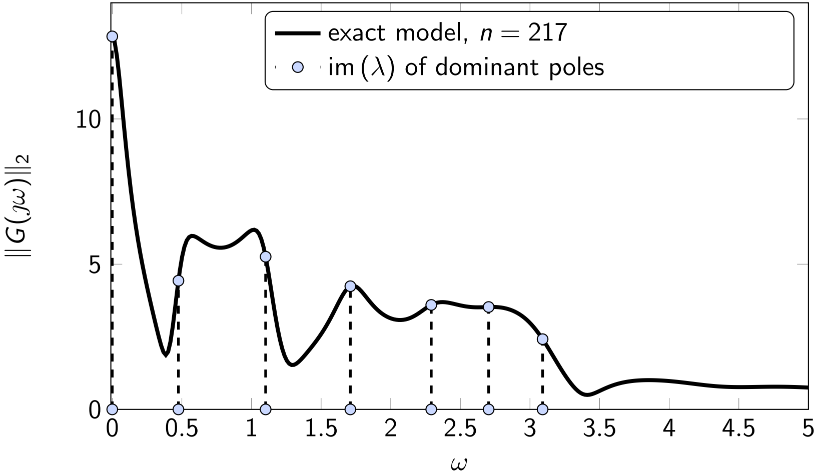
Dominant Poles

Random SISO Example ($B,\ C^T \in \mathbb{R}^n$)



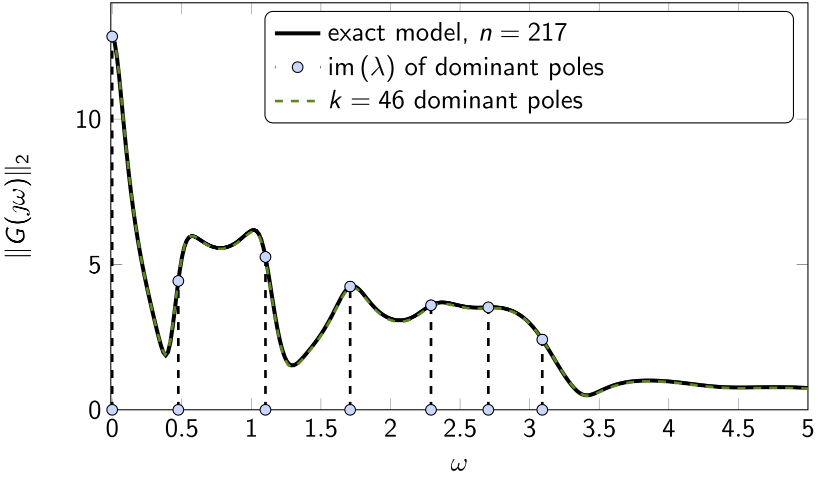
Dominant Poles

Random SISO Example ($B,\ C^T \in \mathbb{R}^n$)



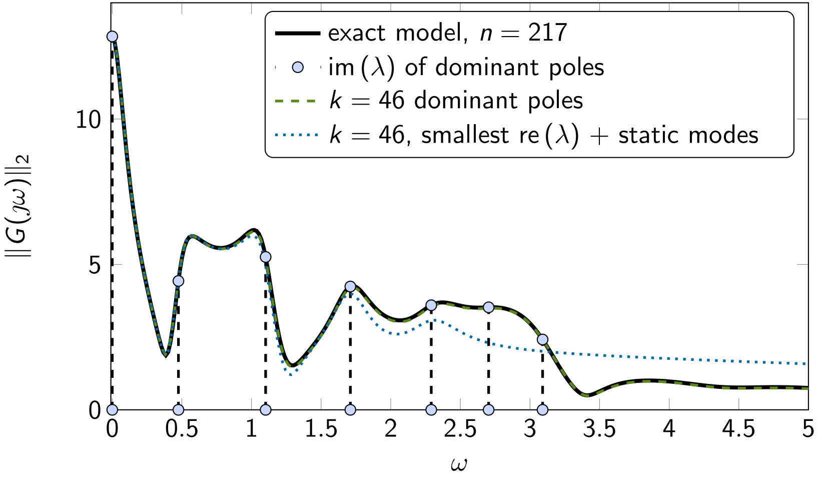
Dominant Poles

Random SISO Example ($B,\ C^T \in \mathbb{R}^n$)



Dominant Poles

Random SISO Example ($B,\ C^T \in \mathbb{R}^n$)

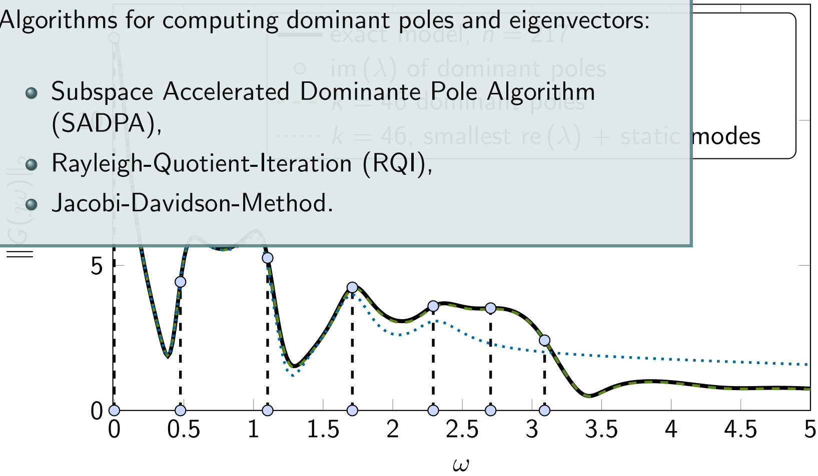


Dominant Poles

Random SISO Example (B, $C^T \in \mathbb{R}^n$)

Algorithms for computing dominant poles and eigenvectors:

- Subspace Accelerated Dominante Pole Algorithm (SADPA),
- Rayleigh-Quotient-Iteration (RQI),
- Jacobi-Davidson-Method.



Outline

1. Introduction
2. Mathematical Basics
3. Model Reduction by Projection
4. Interpolatory Model Reduction
   - Padé Approximation
   - A Change of Perspective: Rational Interpolation
   - $\mathcal{H}_2$-Optimal Model Reduction
5. Balanced Truncation
6. Solving Large-Scale Matrix Equations
7. Final Remarks

Padé Approximation

Idea:

- Consider (even for possibly singular E if $\lambda E - A$ regular):

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

with transfer function $G(s) = C(sE - A)^{-1}B$.

Padé Approximation

Idea:

- Consider (even for possibly singular E if $\lambda E - A$ regular):

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

with transfer function $G(s) = C(sE - A)^{-1}B$.

- For $s_0 \notin \Lambda(A, E)$:

$$G(s) \quad = \quad C\big((s_0E - A) + (s - s_0)E\big)^{-1}B$$

Padé Approximation

Idea:

- Consider (even for possibly singular E if $\lambda E - A$ regular):

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

with transfer function $G(s) = C(sE - A)^{-1}B$.

- For $s_0 \notin \Lambda(A, E)$:

$$\begin{aligned} G(s) &= C\left((s_0E - A) + (s - s_0)E\right)^{-1}B \\ &= C\Big(I + (s - s_0)\underbrace{(s_0E - A)^{-1}E}_{:=\tilde{A}}\Big)^{-1}\underbrace{(s_0E - A)^{-1}B}_{:=\tilde{B}} \end{aligned}$$

Padé Approximation

Idea:

- Consider (even for possibly singular E if $\lambda E - A$ regular):

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

with transfer function $G(s) = C(sE - A)^{-1}B$.

- For $s_0 \notin \Lambda(A, E)$:

$$\begin{aligned} G(s) &= C\left((s_0E - A) + (s - s_0)E\right)^{-1}B \\ &= C\Big(I + (s - s_0)\underbrace{(s_0E - A)^{-1}E}_{:=\tilde{A}}\Big)^{-1}\underbrace{(s_0E - A)^{-1}B}_{:=\tilde{B}} \\ &= C\left(I + (s - s_0)\tilde{A}\right)^{-1}\tilde{B} \end{aligned}$$

Padé Approximation

Idea:

- Consider (even for possibly singular E if $\lambda E - A$ regular):

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

with transfer function $G(s) = C(sE - A)^{-1}B$.

- For $s_0 \notin \Lambda(A, E)$:

$$\begin{aligned} G(s) &= C\left((s_0E - A) + (s - s_0)E\right)^{-1}B \\ &= C\Big(I + (s - s_0)\underbrace{(s_0E - A)^{-1}E}_{:=\tilde{A}}\Big)^{-1}\underbrace{(s_0E - A)^{-1}B}_{:=\tilde{B}} \\ &= C\left(I + (s - s_0)\tilde{A}\right)^{-1}\tilde{B} \end{aligned}$$

Neumann Lemma. $\|F\| < 1 \Rightarrow I - F$ invertible, $(I - F)^{-1} = \sum_{k=0}^{\infty} F^k$.

Padé Approximation

Idea:

- Consider (even for possibly singular E if $\lambda E - A$ regular):

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

with transfer function $G(s) = C(sE - A)^{-1}B$.

- For $s_0 \notin \Lambda(A, E)$:

$$\begin{aligned} G(s) &= C\big((s_0E - A) + (s - s_0)E\big)^{-1}B \\ &= C\Big(I + (s - s_0)\underbrace{(s_0E - A)^{-1}E}_{:=\tilde{A}}\Big)^{-1}\underbrace{(s_0E - A)^{-1}B}_{:=\tilde{B}} \\ &= C\left(I + (s - s_0)\tilde{A}\right)^{-1}\tilde{B} = C\Big(I - \underbrace{\big(-(s - s_0)\tilde{A}\big)}_{=F}\Big)^{-1}\tilde{B} \end{aligned}$$

Neumann Lemma. $\|F\| < 1 \;\Rightarrow\; I - F$ invertible, $(I - F)^{-1} = \sum_{k=0}^{\infty} F^k$.

Padé Approximation

Idea:

- Consider (even for possibly singular E if $\lambda E - A$ regular):

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

with transfer function $G(s) = C(sE - A)^{-1}B$.

- For $s_0 \notin \Lambda(A, E)$, and $\tilde{A} = (s_0 E - A)^{-1}E$, $\tilde{B} = (s_0 E - A)^{-1}B$:

$$\begin{aligned} G(s) &= C\left(I + (s - s_0)\tilde{A}\right)^{-1}\tilde{B} = C\Big(I - \underbrace{\big(-(s - s_0)\tilde{A}\big)}_{=F}\Big)^{-1}\tilde{B} \\ &= C\left(\sum_{k=0}^{\infty}(-1)^k(s - s_0)^k\tilde{A}^k\right)\tilde{B} \end{aligned}$$

Neumann Lemma. $\|F\| < 1 \;\Rightarrow\; I - F$ invertible, $(I - F)^{-1} = \sum_{k=0}^{\infty} F^k$.

Padé Approximation

Idea:

- Consider (even for possibly singular E if $\lambda E - A$ regular):

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

  with transfer function $G(s) = C(sE - A)^{-1}B$.

- For $s_0 \notin \Lambda(A, E)$, and $\tilde{A} = (s_0 E - A)^{-1}E$, $\tilde{B} = (s_0 E - A)^{-1}B$:

$$\begin{aligned} G(s) &= C\left(I + (s - s_0)\tilde{A}\right)^{-1}\tilde{B} = C\Big(I - \underbrace{\big(-(s - s_0)\tilde{A}\big)}_{=F}\Big)^{-1}\tilde{B} \\ &= C\left(\sum_{k=0}^{\infty}(-1)^k(s - s_0)^k\tilde{A}^k\right)\tilde{B} \\ &= \sum_{k=0}^{\infty}\underbrace{(-1)^k C\tilde{A}^k\tilde{B}}_{=: m_k}(s - s_0)^k \end{aligned}$$

Padé Approximation

Idea:

- Consider (even for possibly singular E if $\lambda E - A$ regular):

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

with transfer function $G(s) = C(sE - A)^{-1}B$.

- For $s_0 \notin \Lambda(A, E)$, and $\tilde{A} = (s_0 E - A)^{-1}E$, $\tilde{B} = (s_0 E - A)^{-1}B$:

$$\begin{aligned}
G(s) &= C\left(I + (s - s_0)\tilde{A}\right)^{-1}\tilde{B} = C\Big(I - \underbrace{\big(-(s - s_0)\tilde{A}\big)}_{=F}\Big)^{-1}\tilde{B} \\
&= C\left(\sum_{k=0}^{\infty}(-1)^k(s - s_0)^k\tilde{A}^k\right)\tilde{B} \\
&= \sum_{k=0}^{\infty}\underbrace{(-1)^k C\tilde{A}^k\tilde{B}}_{=:\, m_k}(s - s_0)^k \\
&= m_0 + m_1(s - s_0) + m_2(s - s_0)^2 + \ldots
\end{aligned}$$

Padé Approximation

Idea:

- Consider (even for possibly singular E if $\lambda E - A$ regular):

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

with transfer function $G(s) = C(sE - A)^{-1}B$.

- For $s_0 \notin \Lambda(A, E)$, and $\tilde{A} = (s_0 E - A)^{-1}E$, $\tilde{B} = (s_0 E - A)^{-1}B$:

$$G(s) = m_0 + m_1(s - s_0) + m_2(s - s_0)^2 + \dots$$

with $m_k = (-1)^k C\tilde{A}^k\tilde{B}$.

Padé Approximation

Idea:

- Consider (even for possibly singular E if $\lambda E - A$ regular):

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

with transfer function $G(s) = C(sE - A)^{-1}B$.

- For $s_0 \notin \Lambda(A, E)$, and $\tilde{A} = (s_0 E - A)^{-1}E$, $\tilde{B} = (s_0 E - A)^{-1}B$:

$$G(s) = m_0 + m_1(s - s_0) + m_2(s - s_0)^2 + \dots$$

with $m_k = (-1)^k C\tilde{A}^k\tilde{B}$.

  - For $s_0 = 0$: $m_k := -C(A^{-1}E)^k A^{-1}B \rightsquigarrow$ moments.
    $(m_k = -CA^{-(k+1)}B$ for $E = I_n)$
  - For $s_0 = \infty$ and $E = I_n$: $m_0 = 0$, $m_k := CA^{k-1}B$ for $k \geq 1 \rightsquigarrow$ Markov parameters.

Padé Approximation

Idea:

- Consider (even for possibly singular E if $\lambda E - A$ regular):

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

with transfer function $G(s) = C(sE - A)^{-1}B$.

- For $s_0 \notin \Lambda(A, E)$, and $\tilde{A} = (s_0E - A)^{-1}E$, $\tilde{B} = (s_0E - A)^{-1}B$:

$$G(s) = m_0 + m_1(s - s_0) + m_2(s - s_0)^2 + \ldots$$

with $m_k = (-1)^k C\tilde{A}^k\tilde{B}$.

- As reduced-order model use rth Padé approximant $\hat{G}$ to G:

$$G(s) = \hat{G}(s) + \mathcal{O}((s - s_0)^{2r}),$$

i.e., $m_k = \widehat{m}_k$ for $k = 0, \ldots, 2r - 1$

$\rightsquigarrow$ moment matching if $s_0 < \infty$,

$\rightsquigarrow$ partial realization if $s_0 = \infty$.

Padé Approximation

Asymptotic Waveform Evaluation (AWE) [Pillage/Rohrer 1990]

Consider SISO case ($m = q = 1$) and $s_0 = 0$ for simplicity. With

$$\hat{G}(s) = \frac{\alpha_{r-1}s^{r-1} + \alpha_{r-2}s^{r-2} + \ldots + \alpha_1 s + \alpha_0}{\beta_r s^r + \beta_{r-1}s^{r-1} + \ldots + \beta_1 s + 1},$$

the solution of the Padé approximation problem is obtained via solving

$$M \begin{bmatrix} \beta_r \\ \vdots \\ \beta_1 \end{bmatrix} = \begin{bmatrix} -m_r \\ \vdots \\ -m_{2r-1} \end{bmatrix},$$

with the Hankel matrix $M = \begin{bmatrix} m_0 & m_1 & m_2 & \ldots & m_{r-1} \\ m_1 & m_2 & & \iddots & m_r \\ m_2 & & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & \iddots & \\ m_{r-1} & m_r & \ldots & & m_{2r-2} \end{bmatrix}$.

Then, with $\beta_0 := 1$: $\alpha_j = \sum_{k=0}^{j} m_k \beta_{j-k}$.

Padé Approximation

The Padé-Lanczos Connection [Gallivan/Grimme/Van Dooren 1994, Freund/Feldmann 1994]

Theorem [Grimme '97, Villemagne/Skelton '87]

Let $s_* \notin \Lambda(A, E)$ and

$$\begin{aligned} \tilde{A} &:= (s_* E - A)^{-1} E, \qquad \tilde{B} := (s_* E - A)^{-1} B, \\ \tilde{A}^* &:= (s_* E - A)^{-T} E^T, \quad \tilde{C} := (s_* E - A)^{-T} C^T. \end{aligned}$$

If the reduced-order model is obtained by oblique projection onto $\mathcal{V} \subset \mathbb{R}^n$ along $\mathcal{W} \subset \mathbb{R}^n$, and

$$\begin{aligned} \operatorname{span}\left\{\tilde{B}, \tilde{A}\tilde{B}, \ldots, \tilde{A}^{K-1}\tilde{B}\right\} &\subset \mathcal{V}, \\ \operatorname{span}\left\{\tilde{C}, \tilde{A}^*\tilde{C}, \ldots, (\tilde{A}^*)^{K-1}\tilde{C}\right\} &\subset \mathcal{W}, \end{aligned}$$

then $G(s_*) = \hat{G}(s_*)$, $\frac{d^k}{ds^k} G(s_*) = \frac{d^k}{ds^k} \hat{G}(s_*)$ for $k = 1, \ldots, \ell - 1$, where

$$\ell \begin{cases} = 2K & \text{if } m = q = 1; \\ \geq \lfloor \frac{K}{m} \rfloor + \lfloor \frac{K}{q} \rfloor & \text{if } m \neq 1 \text{ or } q \neq 1. \end{cases}$$

Padé Approximation

The Padé-Lanczos Connection [Gallivan/Grimme/Van Dooren 1994, Freund/Feldmann 1994]

Padé-via-Lanczos Method (PVL)

- Padé approximation/moment matching yield:

$$m_k = \frac{1}{k!} G^{(k)}(s_0) = \frac{1}{k!} \hat{G}^{(k)}(s_0) = \hat{m}_k, \quad k = 0, \ldots, 2K-1,$$

  i.e., Hermite interpolation in s_0.

- Recall interpolation via projection result $\Rightarrow$ moments need not be computed explicitly; moment matching is equivalent to projecting state-space onto

$$\mathcal{V} = \text{span}(\tilde{B}, \tilde{A}\tilde{B}, \ldots, \tilde{A}^{K-1}\tilde{B}) =: \mathcal{K}_K(\tilde{A}, \tilde{B})$$

  (where $\tilde{A} = (s_0 E - A)^{-1} E$, $\tilde{B} = (s_0 E - A)^{-1} B$) along

$$\mathcal{W} = \text{span}(\tilde{C}, \tilde{A}^* \tilde{C}^T, \ldots, (\tilde{A}^*)^{K-1} \tilde{C}) =: \mathcal{K}_K(\tilde{A}^*, \tilde{C}).$$

  (where $\tilde{A}^* = (s_* E - A)^{-T} E^T$, $\tilde{C} = (s_* E - A)^{-T} C^T$).

- Computation via unsymmetric Lanczos method.

Padé Approximation

The Padé-Lanczos Connection [Gallivan/Grimme/Van Dooren 1994, Freund/Feldmann 1994]

Padé-via-Lanczos Method (PVL)

- Padé approximation/moment matching yield:

$$m_k = \frac{1}{k!} G^{(k)}(s_0) = \frac{1}{k!} \hat{G}^{(k)}(s_0) = \hat{m}_k, \quad k = 0, \ldots, 2K-1,$$

  i.e., Hermite interpolation in s_0.

- Recall interpolation via projection result $\Rightarrow$ moments need not be computed explicitly; moment matching is equivalent to projecting state-space onto

$$\mathcal{V} = \mathrm{span}(\tilde{B}, \tilde{A}\tilde{B}, \ldots, \tilde{A}^{K-1}\tilde{B}) =: \mathcal{K}_K(\tilde{A}, \tilde{B})$$

  (where $\tilde{A} = (s_0 E - A)^{-1} E$, $\tilde{B} = (s_0 E - A)^{-1} B$) along

$$\mathcal{W} = \mathrm{span}(\tilde{C}, \tilde{A}^* \tilde{C}^T, \ldots, (\tilde{A}^*)^{K-1} \tilde{C}) =: \mathcal{K}_K(\tilde{A}^*, \tilde{C}).$$

  (where $\tilde{A}^* = (s_* E - A)^{-T} E^T$, $\tilde{C} = (s_* E - A)^{-T} C^T$).

- Computation via unsymmetric Lanczos method.

Padé Approximation

The Padé-Lanczos Connection [Gallivan/Grimme/Van Dooren 1994, Freund/Feldmann 1994]

Padé-via-Lanczos Method (PVL)

- Padé approximation/moment matching yield:

$$m_k = \frac{1}{k!} G^{(k)}(s_0) = \frac{1}{k!} \hat{G}^{(k)}(s_0) = \hat{m}_k, \quad k = 0, \ldots, 2K-1,$$

  i.e., Hermite interpolation in s_0.

- Recall interpolation via projection result $\Rightarrow$ moments need not be computed explicitly; moment matching is equivalent to projecting state-space onto

$$\mathcal{V} = \operatorname{span}(\tilde{B}, \tilde{A}\tilde{B}, \ldots, \tilde{A}^{K-1}\tilde{B}) =: \mathcal{K}_K(\tilde{A}, \tilde{B})$$

  (where $\tilde{A} = (s_0 E - A)^{-1} E$, $\tilde{B} = (s_0 E - A)^{-1} B$) along

$$\mathcal{W} = \operatorname{span}(\tilde{C}, \tilde{A}^* \tilde{C}^T, \ldots, (\tilde{A}^*)^{K-1}\tilde{C}) =: \mathcal{K}_K(\tilde{A}^*, \tilde{C}).$$

  (where $\tilde{A}^* = (s_* E - A)^{-T} E^T$, $\tilde{C} = (s_* E - A)^{-T} C^T$).

- Computation via unsymmetric Lanczos method.

Padé Approximation

The Padé-Lanczos Connection [Gallivan/Grimme/Van Dooren 1994, Freund/Feldmann 1994]

Padé-via-Lanczos Method (PVL)

- Padé approximation/moment matching yield:

$$m_k = \frac{1}{k!} G^{(k)}(s_0) = \frac{1}{k!} \hat{G}^{(k)}(s_0) = \hat{m}_k, \quad k = 0, \ldots, 2K-1,$$

  i.e., Hermite interpolation in s_0.

- Recall interpolation via projection result $\Rightarrow$ moments need not be computed explicitly; moment matching is equivalent to projecting state-space onto

$$\mathcal{V} = \mathrm{span}(\tilde{B}, \tilde{A}\tilde{B}, \ldots, \tilde{A}^{K-1}\tilde{B}) =: \mathcal{K}_K(\tilde{A}, \tilde{B})$$

  (where $\tilde{A} = (s_0 E - A)^{-1} E$, $\tilde{B} = (s_0 E - A)^{-1} B$) along

$$\mathcal{W} = \mathrm{span}(\tilde{C}, \tilde{A}^* \tilde{C}^T, \ldots, (\tilde{A}^*)^{K-1} \tilde{C}) =: \mathcal{K}_K(\tilde{A}^*, \tilde{C}).$$

  (where $\tilde{A}^* = (s_* E - A)^{-T} E^T$, $\tilde{C} = (s_* E - A)^{-T} C^T$).

- Computation via unsymmetric Lanczos method.

Remark: Arnoldi (PRIMA) yields only $G(s) = \hat{G}(s) + \mathcal{O}((s - s_0)^r)$.

Padé Approximation

The Padé-Lanczos Connection [Gallivan/Grimme/Van Dooren 1994, Freund/Feldmann 1994]

Padé-via-Lanczos Method (PVL)

Difficulties:

- Computable error estimates/bounds for $\|y - \hat{y}\|_2$ often very pessimistic or expensive to evaluate.
- Mostly heuristic criteria for choice of expansion points.
  Optimal choice for second-order systems with proportional/Rayleigh damping (Beattie/Gugercin '05).
- Good approximation quality only locally.
- Preservation of physical properties only in special cases (e.g. PRIMA/Arnoldi: $V^T AV$ is stable if A is negative definite or dissipative ⇝ exercises); usually requires post processing which (partially) destroys moment matching properties.

Padé Approximation

The Padé-Lanczos Connection [Gallivan/Grimme/Van Dooren 1994, Freund/Feldmann 1994]

Padé-via-Lanczos Method (PVL)

Difficulties:

- Computable error estimates/bounds for $\|y - \hat{y}\|_2$ often very pessimistic or expensive to evaluate.
- Mostly heuristic criteria for choice of expansion points.
  Optimal choice for second-order systems with proportional/Rayleigh damping (Beattie/Gugercin '05).
- Good approximation quality only locally.
- Preservation of physical properties only in special cases (e.g. PRIMA/Arnoldi: $V^T AV$ is stable if A is negative definite or dissipative ⇝ exercises); usually requires post processing which (partially) destroys moment matching properties.

Padé Approximation

The Padé-Lanczos Connection [Gallivan/Grimme/Van Dooren 1994, Freund/Feldmann 1994]

Padé-via-Lanczos Method (PVL)

Difficulties:

- Computable error estimates/bounds for $\|y - \hat{y}\|_2$ often very pessimistic or expensive to evaluate.
- Mostly heuristic criteria for choice of expansion points.
  Optimal choice for second-order systems with proportional/Rayleigh damping (Beattie/Gugercin '05).
- Good approximation quality only locally.
- Preservation of physical properties only in special cases (e.g. PRIMA/Arnoldi: $V^T AV$ is stable if A is negative definite or dissipative ⇝ exercises); usually requires post processing which (partially) destroys moment matching properties.

Padé Approximation

The Padé-Lanczos Connection [Gallivan/Grimme/Van Dooren 1994, Freund/Feldmann 1994]

Padé-via-Lanczos Method (PVL)

Difficulties:

- Computable error estimates/bounds for $\|y - \hat{y}\|_2$ often very pessimistic or expensive to evaluate.
- Mostly heuristic criteria for choice of expansion points.
  Optimal choice for second-order systems with proportional/Rayleigh damping (Beattie/Gugercin '05).
- Good approximation quality only locally.
- Preservation of physical properties only in special cases (e.g. PRIMA/Arnoldi: $V^T AV$ is stable if A is negative definite or dissipative ⇝ exercises); usually requires post processing which (partially) destroys moment matching properties.

Interpolatory Model Reduction

A Change of Perspective: Rational Interpolation

Computation of reduced-order model by projection

Given an LTI system $\dot{x} = Ax + Bu,\ y = Cx$ with transfer function $G(s) = C(sI_n - A)^{-1}B,$ a reduced-order model is obtained using projection approach with $V, W \in \mathbb{R}^{n\times r}$ and $W^T V = I_r$ by computing

$$\hat{A} = W^T AV, \quad \hat{B} = W^T B, \quad \hat{C} = CV.$$

Petrov-Galerkin-type (two-sided) projection: $W \neq V$,

Galerkin-type (one-sided) projection: $W = V$.

Interpolatory Model Reduction

A Change of Perspective: Rational Interpolation

Computation of reduced-order model by projection

Given an LTI system $\dot{x} = Ax + Bu,\ y = Cx$ with transfer function $G(s) = C(sI_n - A)^{-1}B,$ a reduced-order model is obtained using projection approach with $V, W \in \mathbb{R}^{n\times r}$ and $W^T V = I_r$ by computing

$$\hat{A} = W^T AV, \quad \hat{B} = W^T B, \quad \hat{C} = CV.$$

Petrov-Galerkin-type (two-sided) projection: $W \neq V$,

Galerkin-type (one-sided) projection: $W = V$.

Rational Interpolation/Moment-Matching

Choose V, W such that

$$G(s_j) = \hat{G}(s_j), \quad j = 1, \dots, k,$$

and

$$\frac{d^i}{ds^i} G(s_j) = \frac{d^i}{ds^i} \hat{G}(s_j), \quad i = 1, \dots, K_j, \quad j = 1, \dots, k.$$

Interpolatory Model Reduction

A Change of Perspective: Rational Interpolation

Theorem (simplified) [Grimme '97, Villemagne/Skelton '87]

If

$$\begin{aligned} \operatorname{span}\left\{(s_1 I_n - A)^{-1}B, \ldots, (s_k I_n - A)^{-1}B\right\} &\subset \operatorname{Ran}(V), \\ \operatorname{span}\left\{(s_1 I_n - A)^{-T}C^T, \ldots, (s_k I_n - A)^{-T}C^T\right\} &\subset \operatorname{Ran}(W), \end{aligned}$$

then

$$G(s_j) = \hat{G}(s_j), \quad \frac{d}{ds}G(s_j) = \frac{d}{ds}\hat{G}(s_j), \quad \text{for } j = 1, \ldots, k.$$

Interpolatory Model Reduction

A Change of Perspective: Rational Interpolation

Theorem (simplified) [Grimme '97, Villemagne/Skelton '87]

If

$$\operatorname{span}\left\{(s_1 I_n - A)^{-1}B, \dots, (s_k I_n - A)^{-1}B\right\} \subset \operatorname{Ran}(V),$$
$$\operatorname{span}\left\{(s_1 I_n - A)^{-T}C^T, \dots, (s_k I_n - A)^{-T}C^T\right\} \subset \operatorname{Ran}(W),$$

then

$$G(s_j) = \hat{G}(s_j), \quad \frac{d}{ds}G(s_j) = \frac{d}{ds}\hat{G}(s_j), \quad \text{for } j = 1, \dots, k.$$

Remarks:

using Galerkin/one-sided projection yields $G(s_j) = \hat{G}(s_j)$, but in general

$$\frac{d}{ds}G(s_j) \neq \frac{d}{ds}\hat{G}(s_j).$$

Interpolatory Model Reduction

A Change of Perspective: Rational Interpolation

Theorem (simplified) [Grimme '97, Villemagne/Skelton '87]

If

$$\begin{aligned} \operatorname{span}\left\{(s_1 I_n - A)^{-1}B, \ldots, (s_k I_n - A)^{-1}B\right\} &\subset \operatorname{Ran}(V), \\ \operatorname{span}\left\{(s_1 I_n - A)^{-T}C^T, \ldots, (s_k I_n - A)^{-T}C^T\right\} &\subset \operatorname{Ran}(W), \end{aligned}$$

then

$$G(s_j) = \hat{G}(s_j), \quad \frac{d}{ds}G(s_j) = \frac{d}{ds}\hat{G}(s_j), \quad \text{for } j = 1, \ldots, k.$$

Remarks:

$k = 1$, standard Krylov subspace(s) of dimension $K \rightsquigarrow$ moment-matching methods/Padé approximation,

$$\frac{d^i}{ds^i}G(s_1) = \frac{d^i}{ds^i}\hat{G}(s_1), \quad i = 0, \ldots, K-1 (+K).$$

Interpolatory Model Reduction

A Change of Perspective: Rational Interpolation

Theorem (simplified) [Grimme '97, Villemagne/Skelton '87]

If

$$\begin{aligned} \operatorname{span}\left\{(s_1 I_n - A)^{-1}B, \ldots, (s_k I_n - A)^{-1}B\right\} &\subset \operatorname{Ran}(V), \\ \operatorname{span}\left\{(s_1 I_n - A)^{-T}C^T, \ldots, (s_k I_n - A)^{-T}C^T\right\} &\subset \operatorname{Ran}(W), \end{aligned}$$

then

$$G(s_j) = \hat{G}(s_j), \quad \frac{d}{ds}G(s_j) = \frac{d}{ds}\hat{G}(s_j), \quad \text{for } j = 1, \ldots, k.$$

Remarks:

computation of V, W from rational Krylov subspaces, e.g.,

- dual rational Arnoldi/Lanczos [Grimme '97],
- Iterative Rational Krylov-Algo. [Antoulas/Beattie/Gugercin '07].

$\mathcal{H}_2$-Optimal Model Reduction

Best $\mathcal{H}_2$-norm approximation problem

$$\text{Find} \quad \arg\min_{\hat{G}\in\mathcal{H}_2 \text{ of order } \leq r} \|G - \hat{G}\|_2.$$

$\mathcal{H}_2$-Optimal Model Reduction

Best $\mathcal{H}_2$-norm approximation problem

$$\text{Find} \quad \arg\min_{\hat{G} \in \mathcal{H}_2 \text{ of order } \leq r} \|G - \hat{G}\|_2.$$

$\rightsquigarrow$ First-order necessary $\mathcal{H}_2$-optimality conditions:

For SISO systems

$$G(-\mu_i) = \hat{G}(-\mu_i),$$
$$G'(-\mu_i) = \hat{G}'(-\mu_i),$$

where μ_i are the poles of the reduced transfer function $\hat{G}$.

$\mathcal{H}_2$-Optimal Model Reduction

Best $\mathcal{H}_2$-norm approximation problem

Find $\arg\min_{\hat{G}\in\mathcal{H}_2 \text{ of order } \leq r} \|G - \hat{G}\|_2.$

$\rightsquigarrow$ First-order necessary $\mathcal{H}_2$-optimality conditions:

For MIMO systems

$$\begin{aligned} G(-\mu_i)\tilde{B}_i &= \hat{G}(-\mu_i)\tilde{B}_i, && \text{for } i = 1, \ldots, r, \\ \tilde{C}_i^T G(-\mu_i) &= \tilde{C}_i^T \hat{G}(-\mu_i), && \text{for } i = 1, \ldots, r, \\ \tilde{C}_i^T G'(-\mu_i)\tilde{B}_i &= \tilde{C}_i^T \hat{G}'(-\mu_i)\tilde{B}_i, && \text{for } i = 1, \ldots, r, \end{aligned}$$

where $T^{-1}\hat{A}T = \operatorname{diag}\{\mu_1, \ldots, \mu_r\}$ = spectral decomposition and

$$\tilde{B} = \hat{B}^T T^{-T}, \quad \tilde{C} = \hat{C}T.$$

$\rightsquigarrow$ tangential interpolation conditions.

Interpolatory Model Reduction

Interpolation of the Transfer Function by Projection

Construct reduced transfer function by Petrov-Galerkin projection $\mathcal{P} = VW^T$, i.e.

$$\hat{G}(s) = CV\left(sI - W^TAV\right)^{-1}W^TB,$$

where V and W are given as the rational Krylov subspaces

$$V = \left[(-\mu_1 I - A)^{-1}B, \ldots, (-\mu_r I - A)^{-1}B\right],$$
$$W = \left[(-\mu_1 I - A^T)^{-1}C^T, \ldots, (-\mu_r I - A^T)^{-1}C^T\right].$$

Then

$$G(-\mu_i) = \hat{G}(-\mu_i) \quad \text{and} \quad G'(-\mu_i) = \hat{G}'(-\mu_i),$$

for $i = 1, \ldots, r$ as desired.
$\rightsquigarrow$ iterative algorithms (IRKA/MIRIAm) that yield $\mathcal{H}_2$-optimal models.

[Gugercin et al. '06], [Bunse-Gerstner et al. '07],
[Van Dooren et al. '08]

Interpolatory Model Reduction

Interpolation of the Transfer Function by Projection

Construct reduced transfer function by Petrov-Galerkin projection $\mathcal{P} = VW^T$, i.e.

$$\hat{G}(s) = CV\left(sI - W^TAV\right)^{-1}W^TB,$$

where V and W are given as the rational Krylov subspaces

$$V = \left[(-\mu_1 I - A)^{-1}B, \ldots, (-\mu_r I - A)^{-1}B\right],$$
$$W = \left[(-\mu_1 I - A^T)^{-1}C^T, \ldots, (-\mu_r I - A^T)^{-1}C^T\right].$$

Then

$$G(-\mu_i) = \hat{G}(-\mu_i) \quad \text{and} \quad G'(-\mu_i) = \hat{G}'(-\mu_i),$$

for $i = 1, \ldots, r$ as desired.

$\rightsquigarrow$ iterative algorithms (IRKA/MIRIAm) that yield $\mathcal{H}_2$-optimal models.

[Gugercin et al. '06], [Bunse-Gerstner et al. '07], [Van Dooren et al. '08]

Interpolatory Model Reduction

Interpolation of the Transfer Function by Projection

Construct reduced transfer function by Petrov-Galerkin projection $\mathcal{P} = VW^T$, i.e.

$$\hat{G}(s) = CV\left(sI - W^TAV\right)^{-1} W^TB,$$

where V and W are given as the rational Krylov subspaces

$$\begin{aligned} V &= \left[(-\mu_1 I - A)^{-1}B, \ldots, (-\mu_r I - A)^{-1}B\right], \\ W &= \left[(-\mu_1 I - A^T)^{-1}C^T, \ldots, (-\mu_r I - A^T)^{-1}C^T\right]. \end{aligned}$$

Then

$$G(-\mu_i) = \hat{G}(-\mu_i) \quad \text{and} \quad G'(-\mu_i) = \hat{G}'(-\mu_i),$$

for $i = 1, \ldots, r$ as desired.

$\rightsquigarrow$ iterative algorithms (IRKA/MIRIAm) that yield $\mathcal{H}_2$-optimal models.

[Gugercin et al. '06], [Bunse-Gerstner et al. '07], [Van Dooren et al. '08]

Interpolatory Model Reduction

Interpolation of the Transfer Function by Projection

Construct reduced transfer function by Petrov-Galerkin projection $\mathcal{P} = VW^T$, i.e.

$$\hat{G}(s) = CV\left(sI - W^TAV\right)^{-1} W^TB,$$

where V and W are given as the rational Krylov subspaces

$$V = \left[(-\mu_1 I - A)^{-1}B, \ldots, (-\mu_r I - A)^{-1}B\right],$$
$$W = \left[(-\mu_1 I - A^T)^{-1}C^T, \ldots, (-\mu_r I - A^T)^{-1}C^T\right].$$

Then

$$G(-\mu_i) = \hat{G}(-\mu_i) \quad \text{and} \quad G'(-\mu_i) = \hat{G}'(-\mu_i),$$

for $i = 1, \ldots, r$ as desired.
$\rightsquigarrow$ iterative algorithms (IRKA/MIRIAm) that yield $\mathcal{H}_2$-optimal models.

[Gugercin et al. '06], [Bunse-Gerstner et al. '07],
[Van Dooren et al. '08]

$\mathcal{H}_2$-Optimal Model Reduction

The Basic IRKA Algorithm

Algorithm 1 IRKA (MIMO version/MIRIAm)

Input: A stable, B, C, $\hat{A}$ stable, $\hat{B}$, $\hat{C}$, $\delta > 0$.
Output: A^{opt}, B^{opt}, C^{opt}

while $(\max_{j=1,\ldots,r} \left\{ \frac{|\mu_j - \mu_j^{\text{old}}|}{|\mu_j|} \right\} > \delta)$ **do**

  $\operatorname{diag}\{\mu_1, \ldots, \mu_r\} := T^{-1}\hat{A}T =$ spectral decomposition,
  $\tilde{B} = \hat{B}^H T^{-T}$, $\tilde{C} = \hat{C}T$.

  $V = \left[(-\mu_1 I - A)^{-1} B\tilde{B}_1, \ldots, (-\mu_r I - A)^{-1} B\tilde{B}_r\right]$

  $W = \left[(-\mu_1 I - A^T)^{-1} C^T\tilde{C}_1, \ldots, (-\mu_r I - A^T)^{-1} C^T\tilde{C}_r\right]$

  $V = \operatorname{orth}(V)$, $W = \operatorname{orth}(W)$, $W = W(V^H W)^{-1}$

  $\hat{A} = W^H A V$, $\hat{B} = W^H B$, $\hat{C} = CV$

end while

$A^{opt} = \hat{A}$, $B^{opt} = \hat{B}$, $C^{opt} = \hat{C}$

Outline

Balanced Truncation

Basic principle:

- Recall: a stable system Σ, realized by (A, B, C, D), is called balanced, if the Gramians, i.e., 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 $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$.

- $\Lambda\,(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.

Balanced Truncation

Basic principle:

- Recall: a stable system Σ, realized by (A, B, C, D), is called balanced, if the Gramians, i.e., 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 $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$.

- $\Lambda\,(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.

Balanced Truncation

Basic principle:

- Recall: a stable system Σ, realized by (A, B, C, D), is called balanced, if the Gramians, i.e., 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 $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$.

- $\Lambda\,(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ 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 $\rightsquigarrow (\hat{A}, \hat{B}, \hat{C}, \hat{D}) := (A_{11}, B_1, C_1, D)$.

Balanced Truncation

Basic principle:

- Recall: a stable system Σ, realized by (A, B, C, D), is called balanced, if the Gramians, i.e., 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 $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$.

- $\Lambda\,(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.
- Compute balanced realization of the system via state-space transformation

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

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

Balanced Truncation

Motivation:

The HSVs $\Lambda\,(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are system invariants: they are preserved under

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

Balanced Truncation

Motivation:

The HSVs $\Lambda\,(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are system invariants: they are preserved under

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

in transformed coordinates, the Gramians satisfy

$$\begin{aligned} (TAT^{-1})(TPT^T) + (TPT^T)(TAT^{-1})^T + (TB)(TB)^T &= 0, \\ (TAT^{-1})^T(T^{-T}QT^{-1}) + (T^{-T}QT^{-1})(TAT^{-1}) + (CT^{-1})^T(CT^{-1}) &= 0 \end{aligned}$$

$$\Rightarrow \; (TPT^T)(T^{-T}QT^{-1}) = TPQT^{-1},$$

hence $\Lambda\,(PQ) = \Lambda\,((TPT^T)(T^{-T}QT^{-1}))$.

Balanced Truncation

Implementation: SR Method

1. Compute (Cholesky) factors of the Gramians, $P = S^T S$, $Q = R^T R$.
2. Compute SVD $SR^T = [\, U_1,\ U_2 \,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$.
3. ROM is $(W^T AV, W^T B, CV, D)$, where

$$W = R^T V_1 \Sigma_1^{-\frac{1}{2}}, \qquad V = S^T U_1 \Sigma_1^{-\frac{1}{2}}.$$

Balanced Truncation

Implementation: SR Method

1. Compute (Cholesky) factors of the Gramians, $P = S^T S, \; Q = R^T R$.
2. Compute SVD $SR^T = [\, U_1, \; U_2 \,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$.
3. ROM is $(W^T AV, W^T B, CV, D)$, where

$$W = R^T V_1 \Sigma_1^{-\frac{1}{2}}, \qquad V = S^T U_1 \Sigma_1^{-\frac{1}{2}}.$$

Balanced Truncation

Implementation: SR Method

1. Compute (Cholesky) factors of the Gramians, $P = S^T S, \; Q = R^T R.$
2. Compute SVD $SR^T = [\, U_1, \; U_2 \,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}.$
3. ROM is $(W^T AV, W^T B, CV, D)$, where

$$W = R^T V_1 \Sigma_1^{-\frac{1}{2}}, \qquad V = S^T U_1 \Sigma_1^{-\frac{1}{2}}.$$

Balanced Truncation

Implementation: SR Method

1. Compute (Cholesky) factors of the Gramians, $P = S^T S, \; Q = R^T R$.
2. Compute SVD $SR^T = [\,U_1, \; U_2\,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$.
3. ROM is $(W^T AV, W^T B, CV, D)$, where

$$W = R^T V_1 \Sigma_1^{-\frac{1}{2}}, \qquad V = S^T U_1 \Sigma_1^{-\frac{1}{2}}.$$

Note:

$$V^T W \;=\; (\Sigma_1^{-\frac{1}{2}} U_1^T S)(R^T V_1 \Sigma_1^{-\frac{1}{2}})$$

Balanced Truncation

Implementation: SR Method

1. Compute (Cholesky) factors of the Gramians, $P = S^T S$, $Q = R^T R$.
2. Compute SVD $SR^T = [\, U_1, \, U_2 \,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$.
3. ROM is $(W^T AV, W^T B, CV, D)$, where

$$W = R^T V_1 \Sigma_1^{-\frac{1}{2}}, \qquad V = S^T U_1 \Sigma_1^{-\frac{1}{2}}.$$

Note:

$$V^T W = (\Sigma_1^{-\frac{1}{2}} U_1^T S)(R^T V_1 \Sigma_1^{-\frac{1}{2}}) = \Sigma_1^{-\frac{1}{2}} U_1^T U \Sigma V^T V_1 \Sigma_1^{-\frac{1}{2}}$$

Balanced Truncation

Implementation: SR Method

1. Compute (Cholesky) factors of the Gramians, $P = S^T S, \; Q = R^T R$.
2. Compute SVD $SR^T = [\, U_1, \; U_2 \,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$.
3. ROM is $(W^T AV, W^T B, CV, D)$, where

$$W = R^T V_1 \Sigma_1^{-\frac{1}{2}}, \qquad V = S^T U_1 \Sigma_1^{-\frac{1}{2}}.$$

Note:

$$\begin{aligned} V^T W &= (\Sigma_1^{-\frac{1}{2}} U_1^T S)(R^T V_1 \Sigma_1^{-\frac{1}{2}}) = \Sigma_1^{-\frac{1}{2}} U_1^T U \Sigma V^T V_1 \Sigma_1^{-\frac{1}{2}} \\ &= \Sigma_1^{-\frac{1}{2}} [\, I_r, \, 0 \,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} I_r \\ 0 \end{bmatrix} \Sigma_1^{-\frac{1}{2}} \end{aligned}$$

Balanced Truncation

Implementation: SR Method

1. Compute (Cholesky) factors of the Gramians, $P = S^T S, \; Q = R^T R$.
2. Compute SVD $SR^T = [\, U_1, \; U_2 \,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}$.
3. ROM is $(W^T AV, W^T B, CV, D)$, where

$$W = R^T V_1 \Sigma_1^{-\frac{1}{2}}, \qquad V = S^T U_1 \Sigma_1^{-\frac{1}{2}}.$$

Note:

$$\begin{aligned} V^T W &= (\Sigma_1^{-\frac{1}{2}} U_1^T S)(R^T V_1 \Sigma_1^{-\frac{1}{2}}) = \Sigma_1^{-\frac{1}{2}} U_1^T U \Sigma V^T V_1 \Sigma_1^{-\frac{1}{2}} \\ &= \Sigma_1^{-\frac{1}{2}} [\, I_r, \, 0 \,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} I_r \\ 0 \end{bmatrix} \Sigma_1^{-\frac{1}{2}} = \Sigma_1^{-\frac{1}{2}} \Sigma_1 \Sigma_1^{-\frac{1}{2}} = I_r \end{aligned}$$

$\Longrightarrow VW^T$ is an oblique projector, hence balanced truncation is a Petrov-Galerkin projection method.

Balanced Truncation

Properties:

- Reduced-order model is stable with HSVs $\sigma_1, \ldots, \sigma_r$.
- Adaptive choice of r via computable error bound:

$$\|y - \hat{y}\|_2 \leq \left(2 \sum_{k=r+1}^{n} \sigma_k\right) \|u\|_2.$$

Balanced Truncation

Properties:

- Reduced-order model is stable with HSVs $\sigma_1, \ldots, \sigma_r$.
- Adaptive choice of r via computable error bound:

$$\|y - \hat{y}\|_2 \leq \left(2 \sum_{k=r+1}^{n} \sigma_k\right) \|u\|_2.$$

Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

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

Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

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

Assumptions (for now): $t_0 = 0$, $x_0 = x(0) = 0$, $D = 0$.

Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

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

State-Space Description for I/O-Relation

Variation-of-constants $\Longrightarrow$

$$\mathcal{S} : u \mapsto y, \quad y(t) = \int_{-\infty}^{t} Ce^{A(t-\tau)}Bu(\tau)\,d\tau \quad \text{for all } t \in \mathbb{R}.$$

Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

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

State-Space Description for I/O-Relation

Variation-of-constants $\Longrightarrow$

$$\mathcal{S} : u \mapsto y, \quad y(t) = \int_{-\infty}^{t} Ce^{A(t-\tau)}Bu(\tau)\, d\tau \quad \text{for all } t \in \mathbb{R}.$$

- $\mathcal{S} : \mathcal{U} \to \mathcal{Y}$ is a linear operator between (function) spaces.
- Recall: $A \in \mathbb{R}^{n\times m}$ is a linear operator, $A : \mathbb{R}^m \to \mathbb{R}^n$!
- Basic Idea: use SVD approximation as for matrix A!
- Problem: in general, $\mathcal{S}$ does not have a discrete SVD and can therefore not be approximated as in the matrix case!

Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

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

State-Space Description for I/O-Relation

Variation-of-constants $\Longrightarrow$

$$\mathcal{S} : u \mapsto y, \quad y(t) = \int_{-\infty}^{t} Ce^{A(t-\tau)}Bu(\tau)\, d\tau \quad \text{for all } t \in \mathbb{R}.$$

- $\mathcal{S} : \mathcal{U} \to \mathcal{Y}$ is a linear operator between (function) spaces.
- Recall: $A \in \mathbb{R}^{n\times m}$ is a linear operator, $A : \mathbb{R}^m \to \mathbb{R}^n$!
- Basic Idea: use SVD approximation as for matrix A!
- Problem: in general, $\mathcal{S}$ does not have a discrete SVD and can therefore not be approximated as in the matrix case!

Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

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

State-Space Description for I/O-Relation

Variation-of-constants $\Longrightarrow$

$$\mathcal{S} : u \mapsto y, \quad y(t) = \int_{-\infty}^{t} Ce^{A(t-\tau)}Bu(\tau)\, d\tau \quad \text{for all } t \in \mathbb{R}.$$

- $\mathcal{S} : \mathcal{U} \to \mathcal{Y}$ is a linear operator between (function) spaces.
- Recall: $A \in \mathbb{R}^{n\times m}$ is a linear operator, $A : \mathbb{R}^m \to \mathbb{R}^n$!
- Basic Idea: use SVD approximation as for matrix A!
- Problem: in general, $\mathcal{S}$ does not have a discrete SVD and can therefore not be approximated as in the matrix case!

Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

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

State-Space Description for I/O-Relation

Variation-of-constants $\Longrightarrow$

$$\mathcal{S} : u \mapsto y, \quad y(t) = \int_{-\infty}^{t} Ce^{A(t-\tau)}Bu(\tau)\,d\tau \quad \text{for all } t \in \mathbb{R}.$$

- $\mathcal{S} : \mathcal{U} \to \mathcal{Y}$ is a linear operator between (function) spaces.
- Recall: $A \in \mathbb{R}^{n\times m}$ is a linear operator, $A : \mathbb{R}^m \to \mathbb{R}^n$!
- Basic Idea: use SVD approximation as for matrix A!
- Problem: in general, $\mathcal{S}$ does not have a discrete SVD and can therefore not be approximated as in the matrix case!

Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

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

Alternative to State-Space Operator: Hankel Operator

Instead of

$$\mathcal{S} : u \mapsto y, \quad y(t) = \int_{-\infty}^{t} Ce^{A(t-\tau)}Bu(\tau)\,d\tau \quad \text{for all } t \in \mathbb{R}.$$

use Hankel operator

$$\mathcal{H} : u_- \mapsto y_+, \quad y_+(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu(\tau)\,d\tau \quad \text{for all } t > 0.$$

Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

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

Alternative to State-Space Operator: Hankel Operator

Instead of

$$\mathcal{S} : u \mapsto y, \quad y(t) = \int_{-\infty}^{t} Ce^{A(t-\tau)}Bu(\tau)\,d\tau \quad \text{for all } t \in \mathbb{R}.$$

use Hankel operator

$$\mathcal{H} : u_- \mapsto y_+, \quad y_+(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu(\tau)\,d\tau \quad \text{for all } t > 0.$$

$\mathcal{H}$ compact $\Rightarrow$ $\mathcal{H}$ has discrete SVD

$\rightsquigarrow$ *Hankel singular values* $\{\sigma_j\}_{j=1}^{\infty} : \ \sigma_1 \geq \sigma_2 \geq \ldots \geq 0.$

Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

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

Alternative to State-Space Operator: Hankel Operator

Instead of

$$\mathcal{S} : u \mapsto y, \quad y(t) = \int_{-\infty}^{t} Ce^{A(t-\tau)}Bu(\tau)\, d\tau \quad \text{for all } t \in \mathbb{R}.$$

use Hankel operator

$$\mathcal{H} : u_- \mapsto y_+, \quad y_+(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu(\tau)\, d\tau \quad \text{for all } t > 0.$$

$\mathcal{H}$ compact $\Rightarrow$ $\mathcal{H}$ has discrete SVD

$\rightsquigarrow$ *Hankel singular values* $\{\sigma_j\}_{j=1}^{\infty}: \ \sigma_1 \geq \sigma_2 \geq \ldots \geq 0$.

$\Longrightarrow$ SVD-type approximation of $\mathcal{H}$ possible!

Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

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

Alternative to State-Space Operator: Hankel Operator

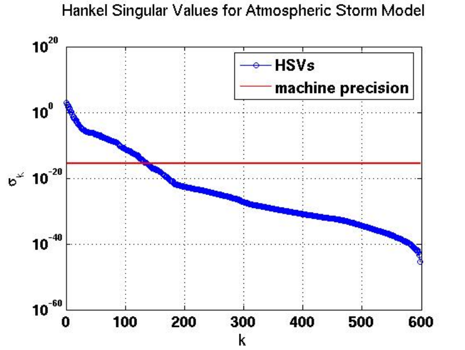
$\mathcal{H}$ compact

$\Downarrow$

$\mathcal{H}$ has discrete SVD

$\Downarrow$

Hankel singular values



Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

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

Alternative to State-Space Operator: Hankel Operator

$$\mathcal{H} : u_- \mapsto y_+, \quad y_+(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu(\tau)\, d\tau \quad \text{for all } t > 0.$$

$\mathcal{H}$ compact $\Rightarrow$ $\mathcal{H}$ has discrete SVD

$\Rightarrow$ Best approximation problem w.r.t. 2-induced operator norm well-posed

$\Rightarrow$ solution: Adamjan-Arov-Krein (AAK Theory, 1971/78).

But: computationally unfeasible for large-scale systems.

Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

$$\begin{array}{rclll} \dot{x} &=& Ax + Bu, & A \in \mathbb{R}^{n\times n}, & B \in \mathbb{R}^{n\times m}, \\ y &=& Cx, & C \in \mathbb{R}^{q\times n}. & \end{array}$$

Alternative to State-Space Operator: Hankel Operator

$$\mathcal{H} : u_- \mapsto y_+, \quad y_+(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu(\tau)\,d\tau \quad \text{for all } t > 0.$$

$\mathcal{H}$ compact $\Rightarrow$ $\mathcal{H}$ has discrete SVD

$\Rightarrow$ Best approximation problem w.r.t. 2-induced operator norm well-posed

$\Rightarrow$ solution: Adamjan-Arov-Krein (AAK Theory, 1971/78).

But: computationally unfeasible for large-scale systems.

Balanced Truncation

Theoretical Background

Linear, Time-Invariant (LTI) Systems

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

Alternative to State-Space Operator: Hankel Operator

$$\mathcal{H} : u_- \mapsto y_+, \quad y_+(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu(\tau)\, d\tau \quad \text{for all } t > 0.$$

$\mathcal{H}$ compact $\Rightarrow$ $\mathcal{H}$ has discrete SVD

$\Rightarrow$ Best approximation problem w.r.t. 2-induced operator norm well-posed

$\Rightarrow$ solution: Adamjan-Arov-Krein (AAK Theory, 1971/78).

But: computationally unfeasible for large-scale systems.

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Hankel operator

$$y_+(t) = \mathcal{H}u_-(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu_-(\tau)\, d\tau$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Hankel operator

$$y_+(t) = \mathcal{H}u_-(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu_-(\tau)\,d\tau =: Ce^{At} \underbrace{\int_{-\infty}^{0} e^{-A\tau}Bu_-(\tau)\,d\tau}_{=:z}$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Hankel operator

$$y_+(t) = \mathcal{H}u_-(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)} Bu_-(\tau)\, d\tau =: Ce^{At} \underbrace{\int_{-\infty}^{0} e^{-A\tau} Bu_-(\tau)\, d\tau}_{=:z} = Ce^{At}z.$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Hankel operator

$$y_+(t) = \mathcal{H}u_-(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu_-(\tau)\,d\tau = Ce^{At}z.$$

Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Hankel operator

$$y_+(t) = \mathcal{H}u_-(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu_-(\tau)\,d\tau = Ce^{At}z.$$

Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

$$\mathcal{H}^* y_+(t) = \int_0^{\infty} B^T e^{A^T(\tau-t)} C^T y_+(\tau)\,d\tau$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Hankel operator

$$y_+(t) = \mathcal{H}u_-(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu_-(\tau)\,d\tau = Ce^{At}z.$$

Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

$$\mathcal{H}^*y_+(t) = \int_0^\infty B^Te^{A^T(\tau-t)}C^Ty_+(\tau)\,d\tau = B^Te^{-A^Tt}\int_0^\infty e^{A^T\tau}C^Ty_+(\tau)\,d\tau.$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Hankel operator

$$y_+(t) = \mathcal{H}u_-(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu_-(\tau)\,d\tau = Ce^{At}z.$$

Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

$$\mathcal{H}^*y_+(t) = \; = B^T e^{-A^T t}\int_0^{\infty} e^{A^T\tau}C^T y_+(\tau)\,d\tau.$$

$$\mathcal{H}^*\mathcal{H}u_-(t) \quad = \quad B^T e^{-A^T t}\int_0^{\infty} e^{A^T\tau}C^T Ce^{A\tau}z\,d\tau$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Hankel operator

$$y_+(t) = \mathcal{H}u_-(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu_-(\tau)\,d\tau = Ce^{At}z.$$

Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

$$\mathcal{H}^* y_+(t) = \; = B^T e^{-A^T t} \int_0^\infty e^{A^T\tau} C^T y_+(\tau)\,d\tau.$$

Hence,

$$\begin{aligned} \mathcal{H}^*\mathcal{H}u_-(t) &= B^T e^{-A^T t} \int_0^\infty e^{A^T\tau} C^T Ce^{A\tau} z\,d\tau \\ &= B^T e^{-A^T t} \underbrace{\int_0^\infty e^{A^T\tau} C^T Ce^{A\tau}\,d\tau}_{\equiv Q}\; z \end{aligned}$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Hankel operator

$$y_+(t) = \mathcal{H}u_-(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu_-(\tau)\,d\tau = Ce^{At}z.$$

Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

$$\mathcal{H}^*y_+(t) = \; = B^T e^{-A^T t}\int_0^{\infty} e^{A^T\tau}C^T y_+(\tau)\,d\tau.$$

Hence,

$$\begin{aligned}\mathcal{H}^*\mathcal{H}u_-(t) &= B^T e^{-A^T t}\int_0^{\infty} e^{A^T\tau}C^T Ce^{A\tau}z\,d\tau \\ &= B^T e^{-A^T t}Qz\end{aligned}$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Hankel operator

$$y_+(t) = \mathcal{H}u_-(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu_-(\tau)\,d\tau = Ce^{At}z.$$

Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

$$\mathcal{H}^* y_+(t) = \; = B^T e^{-A^T t}\int_0^{\infty} e^{A^T\tau} C^T y_+(\tau)\,d\tau.$$

Hence,

$$\mathcal{H}^*\mathcal{H}u_-(t) \quad = \quad B^T e^{-A^T t} Qz$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Hankel operator

$$y_+(t) = \mathcal{H}u_-(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu_-(\tau)\,d\tau = Ce^{At}z.$$

Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

$$\mathcal{H}^* y_+(t) = \; = B^T e^{-A^T t} \int_0^{\infty} e^{A^T\tau} C^T y_+(\tau)\,d\tau.$$

Hence,

$$\mathcal{H}^*\mathcal{H}u_-(t) \quad = \quad B^T e^{-A^T t} Qz \;\doteq\; \sigma^2 u_-(t).$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$, Hence,

$$\mathcal{H}^*\mathcal{H}u_-(t) = B^T e^{-A^T t} Qz \doteq \sigma^2 u_-(t).$$

$\Longrightarrow u_-(t) = \frac{1}{\sigma^2} B^T e^{-A^T t} Qz$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

$$\mathcal{H}^*\mathcal{H}u_-(t) = B^T e^{-A^T t} Qz \doteq \sigma^2 u_-(t).$$

$\Longrightarrow u_-(t) = \frac{1}{\sigma^2} B^T e^{-A^T t} Qz \Longrightarrow$ (recalling $z = \int_{-\infty}^{0} e^{-A\tau} B u_-(\tau)\, d\tau$)

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

$$\mathcal{H}^*\mathcal{H}u_-(t) = B^T e^{-A^T t} Qz \doteq \sigma^2 u_-(t).$$

$$\Longrightarrow u_-(t) = \frac{1}{\sigma^2} B^T e^{-A^T t} Qz \Longrightarrow \text{(recalling } z = \int_{-\infty}^{0} e^{-A\tau} B u_-(\tau)\, d\tau\text{)}$$

$$z = \int_{-\infty}^{0} e^{-A\tau} B \frac{1}{\sigma^2} B^T e^{-A^T \tau} Qz \, d\tau$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

$$\mathcal{H}^*\mathcal{H}u_-(t) = B^T e^{-A^T t} Qz \doteq \sigma^2 u_-(t).$$

$\Longrightarrow u_-(t) = \frac{1}{\sigma^2} B^T e^{-A^T t} Qz \Longrightarrow$ (recalling $z = \int_{-\infty}^0 e^{-A\tau} B u_-(\tau)\, d\tau$)

$$\begin{aligned} z &= \int_{-\infty}^0 e^{-A\tau} B \frac{1}{\sigma^2} B^T e^{-A^T\tau} Qz\, d\tau \\ &= \frac{1}{\sigma^2} \int_{-\infty}^0 e^{-A\tau} BB^T e^{-A^T\tau}\, d\tau\, Qz \end{aligned}$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

$$\mathcal{H}^*\mathcal{H}u_-(t) \quad = \quad B^T e^{-A^T t} Qz \doteq \sigma^2 u_-(t).$$

$$\Longrightarrow u_-(t) = \frac{1}{\sigma^2} B^T e^{-A^T t} Qz \Longrightarrow (\text{recalling } z = \int_{-\infty}^0 e^{-A\tau} B u_-(\tau)\, d\tau)$$

$$\begin{aligned} z &= \int_{-\infty}^0 e^{-A\tau} B \frac{1}{\sigma^2} B^T e^{-A^T \tau} Qz \, d\tau \\ &= \frac{1}{\sigma^2} \int_{-\infty}^0 e^{-A\tau} BB^T e^{-A^T \tau} \, d\tau \, Qz \\ &= \frac{1}{\sigma^2} \underbrace{\int_0^\infty e^{At} BB^T e^{A^T t} \, dt}_{\equiv P} \, Qz \end{aligned}$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

$$\mathcal{H}^*\mathcal{H}u_-(t) = B^T e^{-A^T t} Qz \doteq \sigma^2 u_-(t).$$

$\Longrightarrow u_-(t) = \frac{1}{\sigma^2} B^T e^{-A^T t} Qz \Longrightarrow$ (recalling $z = \int_{-\infty}^0 e^{-A\tau} B u_-(\tau)\, d\tau$)

$$\begin{aligned} z &= \int_{-\infty}^0 e^{-A\tau} B \frac{1}{\sigma^2} B^T e^{-A^T\tau} Qz\, d\tau \\ &= \frac{1}{\sigma^2} \underbrace{\int_0^\infty e^{At} BB^T e^{A^T t}\, dt}_{\equiv P}\; Qz \\ &= \frac{1}{\sigma^2} PQz \end{aligned}$$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Proof: Singular values of $\mathcal{H}$ = square roots of eigenvalues of $\mathcal{H}^*\mathcal{H}$,

$$\mathcal{H}^*\mathcal{H}u_-(t) = B^T e^{-A^T t} Qz \doteq \sigma^2 u_-(t).$$

$\Longrightarrow u_-(t) = \frac{1}{\sigma^2} B^T e^{-A^T t} Qz \Longrightarrow$ (recalling $z = \int_{-\infty}^0 e^{-A\tau} B u_-(\tau)\, d\tau$)

$$\begin{aligned} z &= \int_{-\infty}^0 e^{-A\tau} B \frac{1}{\sigma^2} B^T e^{-A^T \tau} Qz\, d\tau \\ &= \frac{1}{\sigma^2} \underbrace{\int_0^\infty e^{At} BB^T e^{A^T t}\, dt}_{\equiv P}\, Qz \\ &= \frac{1}{\sigma^2} PQz \end{aligned}$$

$\Longleftrightarrow \quad PQz = \sigma^2 z. \quad \square$

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let P, Q be the controllability and observability Gramians of an LTI system Σ. Then the Hankel singular values $\Lambda(PQ)^{\frac{1}{2}} = \{\sigma_1, \ldots, \sigma_n\}$ are the singular values of the Hankel operator associated to Σ.

Theorem

Let the reduced-order system $\hat{\Sigma} : (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ with $r \leq \hat{n}$ be computed by balanced truncation. Then the reduced-order model $\hat{\Sigma}$ is balanced, stable, minimal, and its HSVs are $\sigma_1, \ldots, \sigma_r$.

Balanced Truncation

The Hankel Singular Values are Singular Values!

Theorem

Let the reduced-order system $\hat{\Sigma} : (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ with $r \leq \hat{n}$ be computed by balanced truncation. Then the reduced-order model $\hat{\Sigma}$ is balanced, stable, minimal, and its HSVs are $\sigma_1, \ldots, \sigma_r$.

Proof: Note that in balanced coordinates, the Gramians are diagonal and equal to

$$\mathrm{diag}(\Sigma_1, \Sigma_2) = \mathrm{diag}(\sigma_1, \ldots, \sigma_r, \sigma_{r+1}, \ldots, \sigma_n).$$

Hence, the Gramian satisfies

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} + \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^T + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}^T = 0,$$

whence we obtain the "controllability Lyapunov equation" of the reduced-order system,

$$A_{11}\Sigma_1 + \Sigma_1 A_{11}^T + B_1 B_1^T = 0.$$

The result follows from $\hat{A} = A_{11}, \hat{B} = B_1$, $\Sigma_1 > 0$, the solution theory of Lyapunov equations and the analogous considerations for the observability Gramian. (Minimality is a simple consequence of $\hat{P} = \Sigma_1 = \hat{Q} > 0$.)

Singular Perturbation Approximation (aka Balanced Residualization)

Assume the system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u, \quad y = [\, C_1, \; C_2 \,] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + Du$$

is in balanced coordinates.

Singular Perturbation Approximation (aka Balanced Residualization)

Assume the system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u, \quad y = [\, C_1, \; C_2 \,] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + Du$$

is in balanced coordinates.

Balanced truncation would set $x_2 = 0$ and use (A_{11}, B_1, C_1, D) as reduced-order model, thereby the information present in the remaining model is ignored!

Singular Perturbation Approximation (aka Balanced Residualization)

Assume the system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u, \quad y = [\, C_1, \; C_2 \,] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + Du$$

is in balanced coordinates.

Balanced truncation would set $x_2 = 0$ and use (A_{11}, B_1, C_1, D) as reduced-order model, thereby the information present in the remaining model is ignored!

Particularly, if $G(0) = \hat{G}(0)$ ("zero steady-state error") is required, one can apply the same condensation technique as in Guyan reduction: instead of $x_2 = 0$, set $\dot{x}_2 = 0$. This yields the reduced-order model

$$\begin{aligned} \dot{x}_1 &= (A_{11} - A_{12}A_{22}^{-1}A_{21})x_1 + (B_1 - A_{12}A_{22}^{-1}B_2)u, \\ y &= (C_1 - C_2A_{22}^{-1}A_{21})x_1 + (D - C_2A_{22}^{-1}B_2)u, \end{aligned}$$

with

- the same properties as the reduced-order model w.r.t. stability, minimality, error bound, but $\hat{D} \neq D$;
- zero steady-state error, $G(0) = \hat{G}(0)$ as desired.

Singular Perturbation Approximation (aka Balanced Residualization)

Particularly, if $G(0) = \hat{G}(0)$ ("zero steady-state error") is required, one can apply the same condensation technique as in Guyan reduction: instead of $x_2 = 0$, set $\dot{x}_2 = 0$. This yields the reduced-order model

$$\begin{aligned} \dot{x}_1 &= (A_{11} - A_{12}A_{22}^{-1}A_{21})x_1 + (B_1 - A_{12}A_{22}^{-1}B_2)u, \\ y &= (C_1 - C_2A_{22}^{-1}A_{21})x_1 + (D - C_2A_{22}^{-1}B_2)u, \end{aligned}$$

with

- the same properties as the reduced-order model w.r.t. stability, minimality, error bound, but $\hat{D} \neq D$;
- zero steady-state error, $G(0) = \hat{G}(0)$ as desired.

Note:

- A_{22} invertible as in balanced coordinates, $A_{22}\Sigma_2 + \Sigma_2 A_{22}^T + B_2 B_2^T = 0$ and (A_{22}, B_2) controllable, $\Sigma_2 > 0 \Rightarrow A_{22}$ stable.
- If the original system is not balanced, first compute a minimal realization by applying balanced truncation with $r = \hat{n}$.

Balancing-Related Methods

Basic Principle

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

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

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

Balancing-Related Methods

Basic Principle

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

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

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

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

Balancing-Related Methods

Basic Principle

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

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

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)

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

Balancing-Related Methods

Basic Principle

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

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

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 Methods

Basic Principle

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

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

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

Balancing-Related Methods

Basic Principle

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

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

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 Methods

Properties

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

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

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

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

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

Outline

Solving Large-Scale Matrix Equations

Large-Scale Algebraic Lyapunov and Riccati Equations

Algebraic Riccati equation (ARE) for $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $X \in \mathbb{R}^{n \times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

$G = 0 \Longrightarrow$ Lyapunov equation:

$$0 = \mathcal{L}(X) := A^T X + XA + W.$$

Typical situation in model reduction and optimal control problems for semi-discretized PDEs:

- $n = 10^3 - 10^6$ ($\Longrightarrow 10^6 - 10^{12}$ unknowns!),
- A has sparse representation ($A = -M^{-1}S$ for FEM),
- G, W low-rank with $G, W \in \{BB^T, C^TC\}$, 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!

Solving Large-Scale Matrix Equations

Large-Scale Algebraic Lyapunov and Riccati Equations

Algebraic Riccati equation (ARE) for $A, G = G^T, W = W^T \in \mathbb{R}^{n\times n}$ given and $X \in \mathbb{R}^{n\times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

$G = 0 \Longrightarrow$ Lyapunov equation:

$$0 = \mathcal{L}(X) := A^T X + XA + W.$$

Typical situation in model reduction and optimal control problems for semi-discretized PDEs:

- $n = 10^3 - 10^6$ ($\Longrightarrow 10^6 - 10^{12}$ unknowns!),
- A has sparse representation ($A = -M^{-1}S$ 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!

Solving Large-Scale Matrix Equations

Large-Scale Algebraic Lyapunov and Riccati Equations

Algebraic Riccati equation (ARE) for $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $X \in \mathbb{R}^{n \times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

$G = 0 \Longrightarrow$ Lyapunov equation:

$$0 = \mathcal{L}(X) := A^T X + XA + W.$$

Typical situation in model reduction and optimal control problems for semi-discretized PDEs:

- $n = 10^3 - 10^6$ ($\Longrightarrow 10^6 - 10^{12}$ unknowns!),
- A has sparse representation ($A = -M^{-1}S$ 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!

Solving Large-Scale Matrix Equations

Large-Scale Algebraic Lyapunov and Riccati Equations

Algebraic Riccati equation (ARE) for $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $X \in \mathbb{R}^{n \times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

$G = 0 \Longrightarrow$ Lyapunov equation:

$$0 = \mathcal{L}(X) := A^T X + XA + W.$$

Typical situation in model reduction and optimal control problems for semi-discretized PDEs:

- $n = 10^3 - 10^6$ ($\Longrightarrow 10^6 - 10^{12}$ unknowns!),
- A has sparse representation ($A = -M^{-1}S$ 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!

Solving Large-Scale Matrix Equations

Large-Scale Algebraic Lyapunov and Riccati Equations

Algebraic Riccati equation (ARE) for $A, G = G^T, W = W^T \in \mathbb{R}^{n\times n}$ given and $X \in \mathbb{R}^{n\times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

$G = 0 \Longrightarrow$ Lyapunov equation:

$$0 = \mathcal{L}(X) := A^T X + XA + W.$$

Typical situation in model reduction and optimal control problems for semi-discretized PDEs:

- $n = 10^3 - 10^6$ ($\Longrightarrow 10^6 - 10^{12}$ unknowns!),
- A has sparse representation ($A = -M^{-1}S$ 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!

Solving Large-Scale Matrix Equations

Large-Scale Algebraic Lyapunov and Riccati Equations

Algebraic Riccati equation (ARE) for $A, G = G^T, W = W^T \in \mathbb{R}^{n \times n}$ given and $X \in \mathbb{R}^{n \times n}$ unknown:

$$0 = \mathcal{R}(X) := A^T X + XA - XGX + W.$$

$G = 0 \Longrightarrow$ Lyapunov equation:

$$0 = \mathcal{L}(X) := A^T X + XA + W.$$

Typical situation in model reduction and optimal control problems for semi-discretized PDEs:

- $n = 10^3 - 10^6$ ($\Longrightarrow 10^6 - 10^{12}$ unknowns!),
- A has sparse representation ($A = -M^{-1}S$ 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!

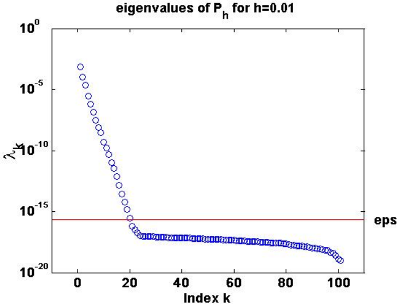
Solving Large-Scale Matrix Equations

Low-Rank Approximation

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: $X = X^T \geq 0 \implies$

$$X = ZZ^T = \sum_{k=1}^{n} \lambda_k z_k z_k^T \approx Z^{(r)}(Z^{(r)})^T = \sum_{k=1}^{r} \lambda_k z_k z_k^T.$$

$\implies$ Goal: compute $Z^{(r)} \in \mathbb{R}^{n \times r}$ directly w/o ever forming X!

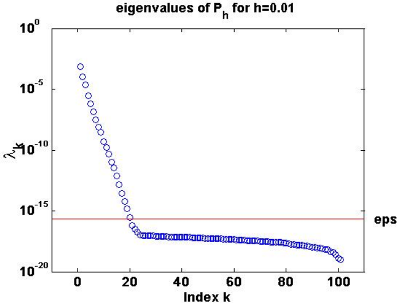
Solving Large-Scale Matrix Equations

Low-Rank Approximation

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: $X = X^T \geq 0 \implies$

$$X = ZZ^T = \sum_{k=1}^{n} \lambda_k z_k z_k^T \approx Z^{(r)}(Z^{(r)})^T = \sum_{k=1}^{r} \lambda_k z_k z_k^T.$$

$\implies$ Goal: compute $Z^{(r)} \in \mathbb{R}^{n \times r}$ directly w/o ever forming X!

Solving Large-Scale Matrix Equations

Linear Matrix Equations

Equations without symmetry

Sylvester equation

$$AX + XB = W$$

discrete Sylvester equation

$$AXB - X = W$$

with data $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times m}$, $W \in \mathbb{R}^{n \times m}$ and unknown $X \in \mathbb{R}^{n \times m}$.

Equations with symmetry

Lyapunov equation

$$AX + XA^T = W$$

Stein equation (discrete Lyapunov equation)

$$AXA^T - X = W$$

with data $A \in \mathbb{R}^{n \times n}$, $W = W^T \in \mathbb{R}^{n \times n}$ and unknown $X \in \mathbb{R}^{n \times n}$.

Here: focus on (Sylvester and) Lyapunov equations; analogous results and methods for discrete versions exist.

Solving Large-Scale Matrix Equations

Linear Matrix Equations

Equations without symmetry

Sylvester equation

$$AX + XB = W$$

discrete Sylvester equation

$$AXB - X = W$$

with data $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times m}$, $W \in \mathbb{R}^{n\times m}$ and unknown $X \in \mathbb{R}^{n\times m}$.

Equations with symmetry

Lyapunov equation

$$AX + XA^T = W$$

Stein equation (discrete Lyapunov equation)

$$AXA^T - X = W$$

with data $A \in \mathbb{R}^{n\times n}$, $W = W^T \in \mathbb{R}^{n\times n}$ and unknown $X \in \mathbb{R}^{n\times n}$.

Here: focus on (Sylvester and) Lyapunov equations; analogous results and methods for discrete versions exist.

Linear Matrix Equations

Solvability

Using the Kronecker (tensor) product, $AX + XB = W$ is equivalent to

$$\left((I_m \otimes A) + \left(B^T \otimes I_n\right)\right) \operatorname{vec}(X) = \operatorname{vec}(W).$$

Hence,

Sylvester equation has a unique solution

$$\Longleftrightarrow$$

$$M := (I_m \otimes A) + \left(B^T \otimes I_n\right) \text{ is invertible.}$$

$$\Longleftrightarrow$$

$$0 \notin \Lambda(M) = \Lambda\left((I_m \otimes A) + (B^T \otimes I_n)\right) = \{\lambda_j + \mu_k, \mid \lambda_j \in \Lambda(A), \ \mu_k \in \Lambda(B)\}.$$

$$\Longleftrightarrow$$

$$\Lambda(A) \cap \Lambda(-B) = \emptyset$$

Corollary

A, B Hurwitz $\Longrightarrow$ Sylvester equation has unique solution.

Linear Matrix Equations

Solvability

Using the Kronecker (tensor) product, $AX + XB = W$ is equivalent to

$$\left((I_m \otimes A) + \left(B^T \otimes I_n\right)\right) \operatorname{vec}(X) = \operatorname{vec}(W).$$

Hence,

Sylvester equation has a unique solution

$$\Longleftrightarrow$$

$M := (I_m \otimes A) + \left(B^T \otimes I_n\right)$ is invertible.

$$\Longleftrightarrow$$

$$0 \notin \Lambda(M) = \Lambda((I_m \otimes A) + (B^T \otimes I_n)) = \{\lambda_j + \mu_k, \mid \lambda_j \in \Lambda(A), \ \mu_k \in \Lambda(B)\}.$$

$$\Longleftrightarrow$$

$$\Lambda(A) \cap \Lambda(-B) = \emptyset$$

Corollary

A, B Hurwitz $\Longrightarrow$ Sylvester equation has unique solution.

Linear Matrix Equations

Solvability

Using the Kronecker (tensor) product, $AX + XB = W$ is equivalent to

$$\left((I_m \otimes A) + \left(B^T \otimes I_n\right)\right) \operatorname{vec}(X) = \operatorname{vec}(W).$$

Hence,

Sylvester equation has a unique solution

$$\Longleftrightarrow$$

$$M := (I_m \otimes A) + \left(B^T \otimes I_n\right) \text{ is invertible.}$$

$$\Longleftrightarrow$$

$$0 \notin \Lambda(M) = \Lambda((I_m \otimes A) + (B^T \otimes I_n)) = \{\lambda_j + \mu_k, \mid \lambda_j \in \Lambda(A), \ \mu_k \in \Lambda(B)\}.$$

$$\Longleftrightarrow$$

$$\Lambda(A) \cap \Lambda(-B) = \emptyset$$

Corollary

A, B Hurwitz $\Longrightarrow$ Sylvester equation has unique solution.

Linear Matrix Equations

Solvability

Using the Kronecker (tensor) product, $AX + XB = W$ is equivalent to

$$\left((I_m \otimes A) + \left(B^T \otimes I_n\right)\right) \operatorname{vec}(X) = \operatorname{vec}(W).$$

Hence,

Sylvester equation has a unique solution

$$\Longleftrightarrow$$

$$M := (I_m \otimes A) + \left(B^T \otimes I_n\right) \text{ is invertible.}$$

$$\Longleftrightarrow$$

$$0 \notin \Lambda(M) = \Lambda\left((I_m \otimes A) + (B^T \otimes I_n)\right) = \{\lambda_j + \mu_k, \mid \lambda_j \in \Lambda(A),\ \mu_k \in \Lambda(B)\}.$$

$$\Longleftrightarrow$$

$$\Lambda(A) \cap \Lambda(-B) = \emptyset$$

Corollary

A, B Hurwitz $\Longrightarrow$ Sylvester equation has unique solution.

Linear Matrix Equations

Solvability

Using the Kronecker (tensor) product, $AX + XB = W$ is equivalent to

$$\left((I_m \otimes A) + \left(B^T \otimes I_n\right)\right) \operatorname{vec}(X) = \operatorname{vec}(W).$$

Hence,

Sylvester equation has a unique solution

$$\Longleftrightarrow$$

$M := (I_m \otimes A) + \left(B^T \otimes I_n\right)$ is invertible.

$$\Longleftrightarrow$$

$$0 \notin \Lambda(M) = \Lambda((I_m \otimes A) + (B^T \otimes I_n)) = \{\lambda_j + \mu_k, \mid \lambda_j \in \Lambda(A), \ \mu_k \in \Lambda(B)\}.$$

$$\Longleftrightarrow$$

$$\Lambda(A) \cap \Lambda(-B) = \emptyset$$

Corollary

A, B Hurwitz $\Longrightarrow$ Sylvester equation has unique solution.

Linear Matrix Equations

Complexity Issues

Solving the Sylvester equation

$$AX + XB = W$$

via the equivalent linear system of equations

$$\left((I_m \otimes A) + \left(B^T \otimes I_n\right)\right) \operatorname{vec}(X) = \operatorname{vec}(W)$$

requires

- LU factorization of $nm \times nm$ matrix; for $n \approx m$, complexity is $\frac{2}{3}n^6$;
- storing $n \cdot m$ unknowns: for $n \approx m$ we have n^4 data for X.

Example

$n = m = 1,000 \Rightarrow$ Gaussian elimination on an Intel core i7 (Westmere, 6 cores, 3.46 GHz $\rightsquigarrow$ 83.2 GFLOP peak) would take > 94 DAYS and 7.3 TB of memory!

Linear Matrix Equations

Complexity Issues

Solving the Sylvester equation

$$AX + XB = W$$

via the equivalent linear system of equations

$$\left((I_m \otimes A) + \left(B^T \otimes I_n\right)\right) \operatorname{vec}(X) = \operatorname{vec}(W)$$

requires

- LU factorization of $nm \times nm$ matrix; for $n \approx m$, complexity is $\frac{2}{3}n^6$;
- storing $n \cdot m$ unknowns: for $n \approx m$ we have n^4 data for X.

Example

$n = m = 1,000 \Rightarrow$ Gaussian elimination on an Intel core i7 (Westmere, 6 cores, 3.46 GHz $\rightsquigarrow$ 83.2 GFLOP peak) would take $>$ 94 DAYS and 7.3 TB of memory!

Numerical Methods for Solving Lyapunov Equations
Traditional Methods

Bartels-Stewart method for Sylvester and Lyapunov equation (`lyap`);
Hessenberg-Schur method for Sylvester equations (`lyap`);
Hammarling's method for Lyapunov equations $AX + XA^T + GG^T = 0$ with A Hurwitz (`lyapchol`).

All based on the fact that if A, B^T are in Schur form, then

$$M = (I_m \otimes A) + \left(B^T \otimes I_n\right)$$

is block-upper triangular. Hence, solve $Mx = b$ by back-substitution.

- Clever implementation of back-substitution process requires $nm(n+m)$ flops.
- For Sylvester eqns., B in Hessenberg form is enough ($\rightsquigarrow$ Hessenberg-Schur method).
- Hammarling's method computes Cholesky factor Y of X directly.
- All methods require Schur decomposition of A and Schur or Hessenberg decomposition of B $\Rightarrow$ need QR algorithm which requires $25n^3$ flops for Schur decomposition.

Not feasible for large-scale problems ($n > 10,000$).

Numerical Methods for Solving Lyapunov Equations
Traditional Methods

Bartels-Stewart method for Sylvester and Lyapunov equation (lyap);
Hessenberg-Schur method for Sylvester equations (lyap);
Hammarling's method for Lyapunov equations $AX + XA^T + GG^T = 0$ with A Hurwitz (lyapchol).

All based on the fact that if A, B^T are in Schur form, then

$$M = (I_m \otimes A) + \left(B^T \otimes I_n\right)$$

is block-upper triangular. Hence, solve $Mx = b$ by back-substitution.

- Clever implementation of back-substitution process requires $nm(n+m)$ flops.
- For Sylvester eqns., B in Hessenberg form is enough ($\rightsquigarrow$ Hessenberg-Schur method).
- Hammarling's method computes Cholesky factor Y of X directly.
- All methods require Schur decomposition of A and Schur or Hessenberg decomposition of B $\Rightarrow$ need QR algorithm which requires $25n^3$ flops for Schur decomposition.

Not feasible for large-scale problems ($n > 10,000$).

Numerical Methods for Solving Lyapunov Equations
Traditional Methods

Bartels-Stewart method for Sylvester and Lyapunov equation (`lyap`);
Hessenberg-Schur method for Sylvester equations (`lyap`);
Hammarling's method for Lyapunov equations $AX + XA^T + GG^T = 0$ with A Hurwitz (`lyapchol`).

All based on the fact that if A, B^T are in Schur form, then

$$M = (I_m \otimes A) + \left(B^T \otimes I_n\right)$$

is block-upper triangular. Hence, solve $Mx = b$ by back-substitution.

- Clever implementation of back-substitution process requires $nm(n+m)$ flops.
- For Sylvester eqns., B in Hessenberg form is enough ($\rightsquigarrow$ Hessenberg-Schur method).
- Hammarling's method computes Cholesky factor Y of X directly.
- All methods require Schur decomposition of A and Schur or Hessenberg decomposition of B $\Rightarrow$ need QR algorithm which requires $25n^3$ flops for Schur decomposition.

Not feasible for large-scale problems ($n > 10,000$).

Numerical Methods for Solving Lyapunov Equations
The Sign Function Method

Definition

For $Z \in \mathbb{R}^{n \times n}$ with $\Lambda(Z) \cap \imath\mathbb{R} = \emptyset$ and Jordan canonical form

$$Z = S \begin{bmatrix} J^+ & 0 \\ 0 & J^- \end{bmatrix} S^{-1}$$

the matrix sign function is

$$\mathrm{sign}(Z) := S \begin{bmatrix} I_k & 0 \\ 0 & -I_{n-k} \end{bmatrix} S^{-1}.$$

Numerical Methods for Solving Lyapunov Equations
The Sign Function Method

Definition

For $Z \in \mathbb{R}^{n \times n}$ with $\Lambda(Z) \cap \imath\mathbb{R} = \emptyset$ and Jordan canonical form

$$Z = S \begin{bmatrix} J^+ & 0 \\ 0 & J^- \end{bmatrix} S^{-1}$$

the matrix sign function is

$$\operatorname{sign}(Z) := S \begin{bmatrix} I_k & 0 \\ 0 & -I_{n-k} \end{bmatrix} S^{-1}.$$

Lemma

Let $T \in \mathbb{R}^{n \times n}$ be nonsingular and Z as before, then

$$\operatorname{sign}\left(TZT^{-1}\right) = T \operatorname{sign}(Z)\, T^{-1}$$

Numerical Methods for Solving Lyapunov Equations
The Sign Function Method

Computation of $\operatorname{sign}(Z)$

$\operatorname{sign}(Z)$ is root of $I_n \Longrightarrow$ use Newton's method to compute it:

$$Z_0 \leftarrow Z, \qquad Z_{j+1} \leftarrow \frac{1}{2}\left(c_j Z_j + \frac{1}{c_j} Z_j^{-1}\right), \qquad j = 1, 2, \ldots$$

$$\Longrightarrow \quad \operatorname{sign}(Z) = \lim_{j\to\infty} Z_j.$$

$c_j > 0$ is scaling parameter for convergence acceleration and rounding error minimization, e.g.

$$c_j = \sqrt{\frac{\|Z_j^{-1}\|_F}{\|Z_j\|_F}},$$

based on "equilibrating" the norms of the two summands [Higham '86].

Solving Lyapunov Equations with the Matrix Sign Function Method

Key observation:
If $X \in \mathbb{R}^{n\times n}$ is a solution of $AX + XA^T + W = 0$, then

$$\underbrace{\begin{bmatrix} I_n & -X \\ 0 & I_n \end{bmatrix}}_{=T^{-1}} \underbrace{\begin{bmatrix} A & W \\ 0 & -A^T \end{bmatrix}}_{=:H} \underbrace{\begin{bmatrix} I_n & X \\ 0 & I_n \end{bmatrix}}_{=:T} = \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix}.$$

Hence, if A is Hurwitz (i.e., asymptotically stable), then

$$\begin{aligned} \operatorname{sign}(H) &= \operatorname{sign}\left(T \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix} T^{-1} \right) = T \operatorname{sign}\left(\begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix} \right) T^{-1} \\ &= \begin{bmatrix} -I_n & 2X \\ 0 & I_n \end{bmatrix}. \end{aligned}$$

Solving Lyapunov Equations with the Matrix Sign Function Method

Key observation:
If $X \in \mathbb{R}^{n\times n}$ is a solution of $AX + XA^T + W = 0$, then

$$\underbrace{\begin{bmatrix} I_n & -X \\ 0 & I_n \end{bmatrix}}_{=T^{-1}} \underbrace{\begin{bmatrix} A & W \\ 0 & -A^T \end{bmatrix}}_{=:H} \underbrace{\begin{bmatrix} I_n & X \\ 0 & I_n \end{bmatrix}}_{=:T} = \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix}.$$

Hence, if A is Hurwitz (i.e., asymptotically stable), then

$$\begin{aligned} \operatorname{sign}(H) &= \operatorname{sign}\left(T \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix} T^{-1} \right) = T \operatorname{sign}\left(\begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix} \right) T^{-1} \\ &= \begin{bmatrix} -I_n & 2X \\ 0 & I_n \end{bmatrix}. \end{aligned}$$

Solving Lyapunov Equations with the Matrix Sign Function Method

Apply sign function iteration $Z \leftarrow \frac{1}{2}(Z + Z^{-1})$ to $H = \begin{bmatrix} A & W \\ 0 & -A^T \end{bmatrix}$:

$$H + H^{-1} = \begin{bmatrix} A & W \\ 0 & -A^T \end{bmatrix} + \begin{bmatrix} A^{-1} & A^{-1}WA^{-T} \\ 0 & -A^{-T} \end{bmatrix}$$

$\Longrightarrow$ Sign function iteration for Lyapunov equation:

$$\begin{aligned} A_0 &\leftarrow A, \quad A_{j+1} \leftarrow \tfrac{1}{2}\left(A_j + A_j^{-1}\right), \\ W_0 &\leftarrow G, \quad W_{j+1} \leftarrow \tfrac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right), \end{aligned} \qquad j = 0, 1, 2, \ldots.$$

Define $A_\infty := \lim_{j\to\infty} A_j$, $W_\infty := \lim_{j\to\infty} W_j$.

Theorem

If A is Hurwitz, then

$$A_\infty = -I_n \qquad \text{and} \qquad X = \frac{1}{2} W_\infty.$$

Solving Lyapunov Equations with the Matrix Sign Function Method

Factored form

Recall sign function iteration for $AX + XA^T + W = 0$:

$$\begin{aligned} A_0 &\leftarrow A, \quad A_{j+1} \leftarrow \tfrac{1}{2}\left(A_j + A_j^{-1}\right), \\ W_0 &\leftarrow G, \quad W_{j+1} \leftarrow \tfrac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right), \end{aligned} \qquad j = 0, 1, 2, \ldots.$$

Now consider the second iteration for $W = BB^T$, starting with $W_0 = BB^T =: B_0 B_0^T$:

$$\begin{aligned} \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right) &= \frac{1}{2}\left(B_j B_j^T + A_j^{-1} B_j B_j^T A_j^{-T}\right) \\ &= \frac{1}{2}\left[B_j \quad A_j^{-1} B_j\right]\left[B_j \quad A_j^{-1} B_j\right]^T. \end{aligned}$$

Hence, obtain factored iteration

$$B_{j+1} \leftarrow \frac{1}{\sqrt{2}}\left[B_j \quad A_j^{-1} B_j\right]$$

with $S := \frac{1}{\sqrt{2}} \lim_{j\to\infty} B_j$ and $X = SS^T$.

Solving Lyapunov Equations with the Matrix Sign Function Method
Factored form

Recall sign function iteration for $AX + XA^T + W = 0$:

$$\begin{aligned} A_0 &\leftarrow A, & A_{j+1} &\leftarrow \tfrac{1}{2}\left(A_j + A_j^{-1}\right), \\ W_0 &\leftarrow G, & W_{j+1} &\leftarrow \tfrac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right), \end{aligned} \qquad j = 0, 1, 2, \ldots.$$

Now consider the second iteration for $W = BB^T$, starting with $W_0 = BB^T =: B_0 B_0^T$:

$$\begin{aligned} \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right) &= \frac{1}{2}\left(B_j B_j^T + A_j^{-1} B_j B_j^T A_j^{-T}\right) \\ &= \frac{1}{2}\left[B_j \quad A_j^{-1} B_j\right]\left[B_j \quad A_j^{-1} B_j\right]^T. \end{aligned}$$

Hence, obtain factored iteration

$$B_{j+1} \leftarrow \frac{1}{\sqrt{2}}\left[B_j \quad A_j^{-1} B_j\right]$$

with $S := \frac{1}{\sqrt{2}} \lim_{j\to\infty} B_j$ and $X = SS^T$.

Solving Lyapunov Equations with the Matrix Sign Function Method
Factored form

Recall sign function iteration for $AX + XA^T + W = 0$:

$$\begin{array}{ll} A_0 \leftarrow A, \quad A_{j+1} \leftarrow \frac{1}{2}\left(A_j + A_j^{-1}\right), \\ W_0 \leftarrow G, \quad W_{j+1} \leftarrow \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right), \end{array} \qquad j = 0, 1, 2, \ldots.$$

Now consider the second iteration for $W = BB^T$, starting with $W_0 = BB^T =: B_0 B_0^T$:

$$\begin{aligned} \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right) &= \frac{1}{2}\left(B_j B_j^T + A_j^{-1} B_j B_j^T A_j^{-T}\right) \\ &= \frac{1}{2}\left[B_j \quad A_j^{-1} B_j\right]\left[B_j \quad A_j^{-1} B_j\right]^T. \end{aligned}$$

Hence, obtain factored iteration

$$B_{j+1} \leftarrow \frac{1}{\sqrt{2}}\left[B_j \quad A_j^{-1} B_j\right]$$

with $S := \frac{1}{\sqrt{2}} \lim_{j\to\infty} B_j$ and $X = SS^T$.

Solving Lyapunov Equations with the Matrix Sign Function Method

Factored form

Recall sign function iteration for $AX + XA^T + W = 0$:

$$\begin{array}{ll} A_0 \leftarrow A, \quad A_{j+1} \leftarrow \frac{1}{2}\left(A_j + A_j^{-1}\right), & \\ W_0 \leftarrow G, \quad W_{j+1} \leftarrow \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right), & \end{array} \qquad j = 0, 1, 2, \ldots.$$

Now consider the second iteration for $W = BB^T$, starting with $W_0 = BB^T =: B_0 B_0^T$:

$$\begin{aligned} \frac{1}{2}\left(W_j + A_j^{-1} W_j A_j^{-T}\right) &= \frac{1}{2}\left(B_j B_j^T + A_j^{-1} B_j B_j^T A_j^{-T}\right) \\ &= \frac{1}{2}\begin{bmatrix} B_j & A_j^{-1} B_j \end{bmatrix} \begin{bmatrix} B_j & A_j^{-1} B_j \end{bmatrix}^T. \end{aligned}$$

Hence, obtain factored iteration

$$B_{j+1} \leftarrow \frac{1}{\sqrt{2}} \begin{bmatrix} B_j & A_j^{-1} B_j \end{bmatrix}$$

with $S := \frac{1}{\sqrt{2}} \lim_{j\to\infty} B_j$ and $X = SS^T$.

Solving Lyapunov Equations with the Matrix Sign Function Method

Factored form **[B./Quintana-Ortí '97]**

Factored sign function iteration for $A(SS^T) + (SS^T)A^T + BB^T = 0$

$$\begin{array}{ll} A_0 \leftarrow A, & A_{j+1} \leftarrow \frac{1}{2}\left(A_j + A_j^{-1}\right), \\ B_0 \leftarrow B, & B_{j+1} \leftarrow \frac{1}{\sqrt{2}}\left[B_j \quad A_j^{-1}B_j\right], \end{array} \qquad j = 0, 1, 2, \ldots.$$

Remarks:

- To get both Gramians, run in parallel

$$C_{j+1} \leftarrow \frac{1}{\sqrt{2}}\begin{bmatrix} C_j \\ C_j A_j^{-1} \end{bmatrix}.$$

- To avoid growth in numbers of columns of B_j (or rows of C_j): column compression by RRLQ or truncated SVD.
- Several options to incorporate scaling, e.g., scale "A"-iteration only.
- Simple stopping cirterion: $\|A_j + I_n\|_F \leq tol$.

Numerical Methods for Solving Lyapunov Equations
The ADI Method

Recall Peaceman Rachford ADI:
Consider $Au = s$ where $A \in \mathbb{R}^{n \times n}$ spd, $s \in \mathbb{R}^n$. ADI Iteration Idea:
Decompose $A = H + V$ with $H, V \in \mathbb{R}^{n \times n}$ such that

$$\begin{array}{c}(H + pI)v = r \\ (V + pI)w = t\end{array}$$

can be solved easily/efficiently.

ADI Iteration

If H, V spd $\Rightarrow \exists p_k,\ k = 1, 2, \ldots$ such that

$$\begin{array}{rcl} u_0 & = & 0 \\ (H + p_k I)u_{k-\frac{1}{2}} & = & (p_k I - V)u_{k-1} + s \\ (V + p_k I)u_k & = & (p_k I - H)u_{k-\frac{1}{2}} + s \end{array}$$

converges to $u \in \mathbb{R}^n$ solving $Au = s$.

Numerical Methods for Solving Lyapunov Equations
The ADI Method

Recall Peaceman Rachford ADI:
Consider $Au = s$ where $A \in \mathbb{R}^{n \times n}$ spd, $s \in \mathbb{R}^n$. ADI Iteration Idea:
Decompose $A = H + V$ with $H, V \in \mathbb{R}^{n \times n}$ such that

$$(H + pI)v = r$$
$$(V + pI)w = t$$

can be solved easily/efficiently.

ADI Iteration

If H, V spd $\Rightarrow \exists p_k,\ k = 1, 2, \ldots$ such that

$$\begin{aligned} u_0 &= 0 \\ (H + p_k I)u_{k-\frac{1}{2}} &= (p_k I - V)u_{k-1} + s \\ (V + p_k I)u_k &= (p_k I - H)u_{k-\frac{1}{2}} + s \end{aligned}$$

converges to $u \in \mathbb{R}^n$ solving $Au = s$.

Numerical Methods for Solving Lyapunov Equations

The Lyapunov operator

$$\mathcal{L}: \quad P \quad \mapsto \quad AX + XA^T$$

can be decomposed into the linear operators

$$\mathcal{L}_H : X \mapsto AX, \qquad \mathcal{L}_V : X \mapsto XA^T.$$

In analogy to the standard ADI method we find the

ADI iteration for the Lyapunov equation [Wachspress '88]

$$\begin{aligned} X_0 &= 0 \\ (A + p_k I)X_{k-\frac{1}{2}} &= -W - X_{k-1}(A^T - p_k I) \\ (A + p_k I)X_k^T &= -W - X_{k-\frac{1}{2}}^T(A^T - p_k I). \end{aligned}$$

Numerical Methods for Solving Lyapunov Equations
Low-Rank ADI

Consider $AX + XA^T = -BB^T$ for stable A; $B \in \mathbb{R}^{n \times m}$ with $m \ll n$.

ADI iteration for the Lyapunov equation [Wachspress '95]

For $k = 1, \ldots, k_{\max}$

$$\begin{aligned} X_0 &= 0 \\ (A + p_k I)X_{k-\frac{1}{2}} &= -BB^T - X_{k-1}(A^T - p_k I) \\ (A + p_k I)X_k^T &= -BB^T - X_{k-\frac{1}{2}}^T(A^T - p_k I) \end{aligned}$$

Rewrite as one step iteration and factorize $X_k = Z_k Z_k^T$, $k = 0, \ldots, k_{\max}$

$$\begin{aligned} Z_0 Z_0^T &= 0 \\ Z_k Z_k^T &= -2p_k (A + p_k I)^{-1} BB^T (A + p_k I)^{-T} \\ &\quad + (A + p_k I)^{-1}(A - p_k I) Z_{k-1} Z_{k-1}^T (A - p_k I)^T (A + p_k I)^{-T} \end{aligned}$$

. . . ⇝ low-rank Cholesky factor ADI

[Penzl '97/'00, Li/White '99/'02, B./Li/Penzl '99/'08, Gugercin/Sorensen/Antoulas '03]

Numerical Methods for Solving Lyapunov Equations
Low-Rank ADI

Consider $AX + XA^T = -BB^T$ for stable A; $B \in \mathbb{R}^{n\times m}$ with $m \ll n$.

ADI iteration for the Lyapunov equation [Wachspress '95]

For $k = 1, \ldots, k_{\max}$

$$\begin{aligned} X_0 &= 0 \\ (A + p_k I)X_{k-\frac{1}{2}} &= -BB^T - X_{k-1}(A^T - p_k I) \\ (A + p_k I)X_k^T &= -BB^T - X_{k-\frac{1}{2}}^T(A^T - p_k I) \end{aligned}$$

Rewrite as one step iteration and factorize $X_k = Z_k Z_k^T$, $k = 0, \ldots, k_{\max}$

$$\begin{aligned} Z_0 Z_0^T &= 0 \\ Z_k Z_k^T &= -2p_k(A + p_k I)^{-1}BB^T(A + p_k I)^{-T} \\ &\quad +(A + p_k I)^{-1}(A - p_k I)Z_{k-1}Z_{k-1}^T(A - p_k I)^T(A + p_k I)^{-T} \end{aligned}$$

$\ldots \rightsquigarrow$ low-rank Cholesky factor ADI

[Penzl '97/'00, Li/White '99/'02, B./Li/Penzl '99/'08, Gugercin/Sorensen/Antoulas '03]

Numerical Methods for Solving Lyapunov Equations
Low-Rank ADI

Consider $AX + XA^T = -BB^T$ for stable A; $B \in \mathbb{R}^{n\times m}$ with $m \ll n$.

ADI iteration for the Lyapunov equation [Wachspress '95]

For $k = 1, \ldots, k_{\max}$

$$\begin{aligned} X_0 &= 0 \\ (A + p_k I)X_{k-\frac{1}{2}} &= -BB^T - X_{k-1}(A^T - p_k I) \\ (A + p_k I)X_k^T &= -BB^T - X_{k-\frac{1}{2}}^T(A^T - p_k I) \end{aligned}$$

Rewrite as one step iteration and factorize $X_k = Z_k Z_k^T$, $k = 0, \ldots, k_{\max}$

$$\begin{aligned} Z_0 Z_0^T &= 0 \\ Z_k Z_k^T &= -2p_k (A + p_k I)^{-1} BB^T (A + p_k I)^{-T} \\ &\quad + (A + p_k I)^{-1}(A - p_k I) Z_{k-1} Z_{k-1}^T (A - p_k I)^T (A + p_k I)^{-T} \end{aligned}$$

. . . $\rightsquigarrow$ low-rank Cholesky factor ADI

[Penzl '97/'00, Li/White '99/'02, B./Li/Penzl '99/'08, Gugercin/Sorensen/Antoulas '03]

Solving Large-Scale Matrix Equations
Numerical Methods for Solving Lyapunov Equations

$$Z_k = [\sqrt{-2p_k}(A + p_k I)^{-1}B,\ (A + p_k I)^{-1}(A - p_k I)Z_{k-1}]$$

[PENZL '00]

Observing that $(A - p_i I)$, $(A + p_k I)^{-1}$ commute, we rewrite $Z_{k_{\max}}$ as

$$Z_{k_{\max}} = [z_{k_{\max}},\ P_{k_{\max}-1} z_{k_{\max}},\ P_{k_{\max}-2}(P_{k_{\max}-1} z_{k_{\max}}),\ \ldots,\ P_1(P_2 \cdots P_{k_{\max}-1} z_{k_{\max}})],$$

[LI/WHITE '02]

where

$$z_{k_{\max}} = \sqrt{-2p_{k_{\max}}}(A + p_{k_{\max}} I)^{-1} B$$

and

$$P_i := \frac{\sqrt{-2p_i}}{\sqrt{-2p_{i+1}}}\left[I - (p_i + p_{i+1})(A + p_i I)^{-1}\right].$$

Solving Large-Scale Matrix Equations

Numerical Methods for Solving Lyapunov Equations

$$Z_k = [\sqrt{-2p_k}(A + p_k I)^{-1}B,\ (A + p_k I)^{-1}(A - p_k I)Z_{k-1}]$$

[Penzl '00]

Observing that $(A - p_i I)$, $(A + p_k I)^{-1}$ commute, we rewrite $Z_{k_{\max}}$ as

$$Z_{k_{\max}} = [z_{k_{\max}},\ P_{k_{\max}-1}z_{k_{\max}},\ P_{k_{\max}-2}(P_{k_{\max}-1}z_{k_{\max}}),\ \ldots,\ P_1(P_2 \cdots P_{k_{\max}-1}z_{k_{\max}})],$$

[Li/White '02]

where

$$z_{k_{\max}} = \sqrt{-2p_{k_{\max}}}(A + p_{k_{\max}} I)^{-1}B$$

and

$$P_i := \frac{\sqrt{-2p_i}}{\sqrt{-2p_{i+1}}}\left[I - (p_i + p_{i+1})(A + p_i I)^{-1}\right].$$

Numerical Methods for Solving Lyapunov Equations

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

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

$$V_1 \leftarrow \sqrt{-2\,\mathrm{re}\,p_1}(A + p_1 I)^{-1}B, \qquad Z_1 \leftarrow V_1$$

FOR $k = 2, 3, \ldots$

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

$$Z_k \leftarrow \left[\begin{array}{cc} Z_{k-1} & V_k \end{array}\right]$$

$$Z_k \leftarrow \mathrm{rrlq}(Z_k, \tau) \qquad \text{column compression}$$

At convergence, $Z_{k_{\max}} Z_{k_{\max}}^T \approx X$, where (without column compression)

$$Z_{k_{\max}} = \left[\begin{array}{ccc} V_1 & \ldots & V_{k_{\max}} \end{array}\right], \quad V_k = \left[\;\right] \in \mathbb{C}^{n \times m}.$$

Note: Implementation in real arithmetic possible by combining two steps [B./Li/Penzl '99/'08] or using new idea employing the relation of 2 consecutive complex factors [B./Kürschner/Saak '11].

Numerical Methods for Solving Lyapunov Equations

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

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

$$V_1 \leftarrow \sqrt{-2\,\mathrm{re}\,p_1}(A + p_1 I)^{-1}B, \qquad Z_1 \leftarrow V_1$$

FOR $k = 2, 3, \ldots$

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

$$Z_k \leftarrow \begin{bmatrix} Z_{k-1} & V_k \end{bmatrix}$$

$$Z_k \leftarrow \mathrm{rrlq}(Z_k, \tau) \qquad \text{column compression}$$

At convergence, $Z_{k_{\max}} Z_{k_{\max}}^T \approx X$, where (without column compression)

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

Note: Implementation in real arithmetic possible by combining two steps [B./Li/Penzl '99/'08] or using new idea employing the relation of 2 consecutive complex factors [B./Kürschner/Saak '11].

Numerical Results for ADI

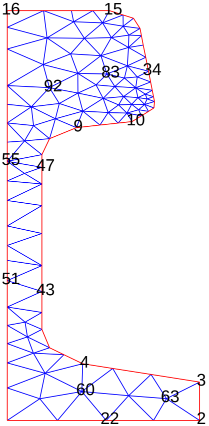
Optimal Cooling of Steel Profiles

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

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

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

- 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.

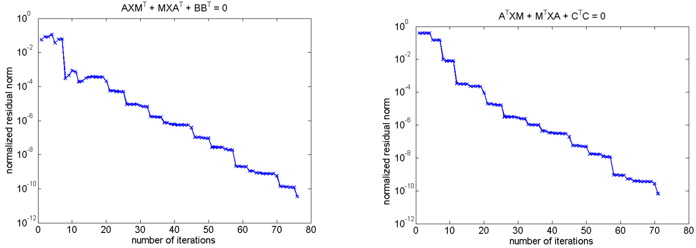
Numerical Results for ADI

Optimal Cooling of Steel Profiles

- Solve dual Lyapunov equations needed for balanced truncation, i.e.,

$$APM^T + MPA^T + BB^T = 0, \quad A^TQM + M^TQA + C^TC = 0,$$

  for $n = 79,841$.
- 25 shifts chosen by Penzl heuristic from 50/25 Ritz values of A of largest/smallest magnitude, no column compression performed.
- No factorization of mass matrix required.
- Computations done on Core2Duo at 2.8GHz with 3GB RAM and 32Bit-MATLAB.



CPU times: 626 / 356 sec.

Numerical Results for ADI

Scaling / Mesh Independence **Computations by Martin Köhler '10**

- $A \in \mathbb{R}^{n \times n} \equiv$ FDM matrix for 2D heat equation on $[0, 1]^2$ (Lyapack benchmark `demo_l1`, $m = 1$).
- 16 shifts chosen by Penzl heuristic from 50/25 Ritz values of A of largest/smallest magnitude.
- Computations on 2 dual core Intel Xeon 5160 with 16 GB RAM using M.E.S.S. (http://svncsc.mpi-magdeburg.mpg.de/trac/messtrac/).

Numerical Results for ADI

Scaling / Mesh Independence

Computations by Martin Köhler '10

- $A \in \mathbb{R}^{n \times n} \equiv$ FDM matrix for 2D heat equation on $[0,1]^2$ (LYAPACK benchmark `demo_l1`, $m = 1$).
- 16 shifts chosen by Penzl heuristic from 50/25 Ritz values of A of largest/smallest magnitude.
- Computations on 2 dual core Intel Xeon 5160 with 16 GB RAM using M.E.S.S. (http://svncsc.mpi-magdeburg.mpg.de/trac/messtrac/).

CPU Times

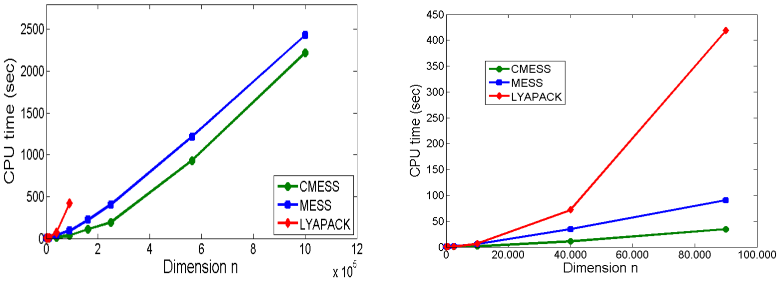
n	M.E.S.S.[1] (C)	LyaPack	M.E.S.S. (MATLAB)
100	0.023	0.124	0.158
625	0.042	0.104	0.227
2,500	0.159	0.702	0.989
10,000	0.965	6.22	5.644
40,000	11.09	71.48	34.55
90,000	34.67	418.5	90.49
160,000	109.3	out of memory	219.9
250,000	193.7	out of memory	403.8
562,500	930.1	out of memory	1216.7
1,000,000	2220.0	out of memory	2428.6

Numerical Results for ADI

Scaling / Mesh Independence

Computations by Martin Köhler '10

- $A \in \mathbb{R}^{n \times n} \equiv$ FDM matrix for 2D heat equation on $[0,1]^2$ (LYAPACK benchmark `demo_l1`, $m = 1$).
- 16 shifts chosen by Penzl heuristic from 50/25 Ritz values of A of largest/smallest magnitude.
- Computations on 2 dual core Intel Xeon 5160 with 16 GB RAM using M.E.S.S. (http://svncsc.mpi-magdeburg.mpg.de/trac/messtrac/).



Note: for $n = 1{,}000{,}000$, first sparse LU needs $\sim 1{,}100$ sec., using UMFPACK this reduces to 30 sec.

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 $\operatorname{range}(Z)$, $Z \in \mathbb{R}^{n \times r}$, for subspace $\mathcal{Z} \subset \mathbb{R}^n$, $\dim \mathcal{Z} = r$.
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 $m = 1$:

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

  [Saad '90, Jaimoukha/Kasenally '94, Jbilou '02–'08].

- K-PIK [Simoncini '07],

$$\mathcal{Z} = \mathcal{K}(A, B, r) \cup \mathcal{K}(A^{-1}, B, r).$$

- Rational Krylov [Druskin/Simoncini '11] ($\rightsquigarrow$ exercises).

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 $\mathrm{range}(Z)$, $Z \in \mathbb{R}^{n \times r}$, for subspace $\mathcal{Z} \subset \mathbb{R}^n$, $\dim \mathcal{Z} = r$.
2. Set $\hat{A} := Z^T A Z$, $\hat{B} := Z^T B$.
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 $m = 1$:

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

[Saad '90, Jaimoukha/Kasenally '94, Jbilou '02–'08].

- K-PIK [Simoncini '07],

$$\mathcal{Z} = \mathcal{K}(A, B, r) \cup \mathcal{K}(A^{-1}, B, r).$$

- Rational Krylov [Druskin/Simoncini '11] ($\rightsquigarrow$ exercises).

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 $\mathrm{range}(Z)$, $Z \in \mathbb{R}^{n \times r}$, for subspace $\mathcal{Z} \subset \mathbb{R}^n$, $\dim \mathcal{Z} = r$.
2. Set $\hat{A} := Z^T A Z$, $\hat{B} := Z^T B$.
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 $m = 1$:

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

  [Saad '90, Jaimoukha/Kasenally '94, Jbilou '02–'08].

- K-PIK [Simoncini '07],

$$\mathcal{Z} = \mathcal{K}(A, B, r) \cup \mathcal{K}(A^{-1}, B, r).$$

- Rational Krylov [Druskin/Simoncini '11] ($\rightsquigarrow$ exercises).

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 $\mathrm{range}(Z)$, $Z \in \mathbb{R}^{n \times r}$, for subspace $\mathcal{Z} \subset \mathbb{R}^n$, $\dim \mathcal{Z} = r$.
2. Set $\hat{A} := Z^T A Z$, $\hat{B} := Z^T B$.
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]:

$$\mathcal{Z} = \mathrm{colspan}\left[\; V_1, \quad \ldots, \quad V_r \;\right].$$

Note:

1. ADI subspace is rational Krylov subspace [J.-R. Li/White '02].
2. Similar approach: ADI-preconditioned global Arnoldi method [Jbilou '08].

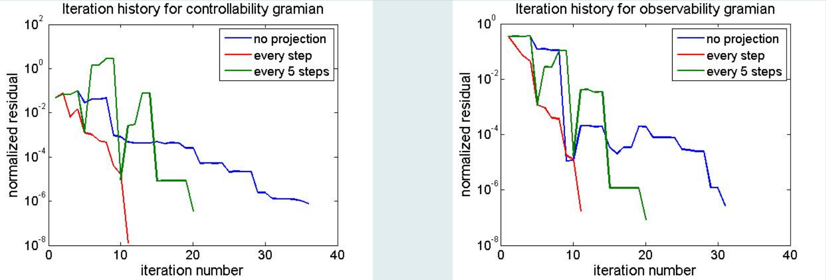
Numerical Methods for Solving Lyapunov Equations

Numerical examples for Galerkin-ADI

FEM semi-discretized control problem for parabolic PDE:

- optimal cooling of rail profiles,
- $n = 20,209,\ m = 7,\ q = 6.$

Good ADI shifts



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

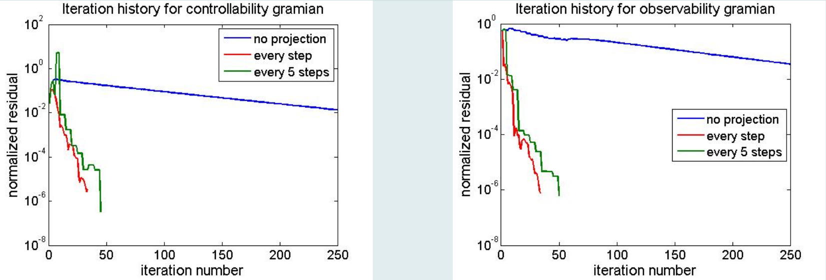
Computations by Jens Saak '10.

Numerical Methods for Solving Lyapunov Equations
Numerical examples for Galerkin-ADI

FEM semi-discretized control problem for parabolic PDE:

- optimal cooling of rail profiles,
- $n = 20,209,\ m = 7,\ q = 6.$

Bad ADI shifts



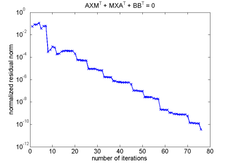
CPU times: 368s (projection every 5th ADI step) vs. 1207s (no projection).

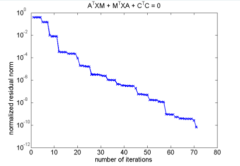
Computations by Jens Saak '10.

Numerical Methods for Solving Lyapunov Equations

Numerical examples for Galerkin-ADI: optimal cooling of rail profiles, $n = 79,841$.

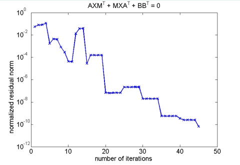
M.E.S.S. w/o Galerkin projection and column compression

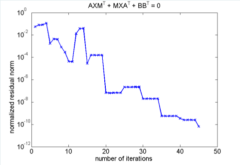




Rank of solution factors: 532 / 426

M.E.S.S. with Galerkin projection and column compression



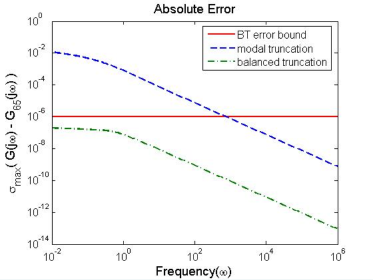


Rank of solution factors: 269 / 205

Solving Large-Scale Matrix Equations

Numerical example for BT: Optimal Cooling of Steel Profiles

$n = 1,357$, Absolute Error

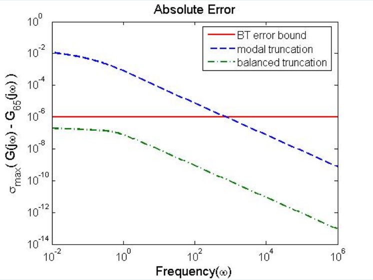


- BT model computed with sign function method,
- MT w/o static condensation, same order as BT model.

Solving Large-Scale Matrix Equations

Numerical example for BT: Optimal Cooling of Steel Profiles

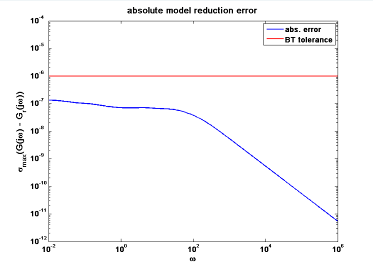
$n = 1,357$, Absolute Error



– BT model computed with sign function method,

– MT w/o static condensation, same order as BT model.

$n = 79,841$, Absolute Error



– BT model computed using M.E.S.S. in MATLAB,

– dualcore, computation time: <10 min.

Solving Large-Scale Matrix Equations

Numerical example for BT: Microgyroscope (Butterfly Gyro)

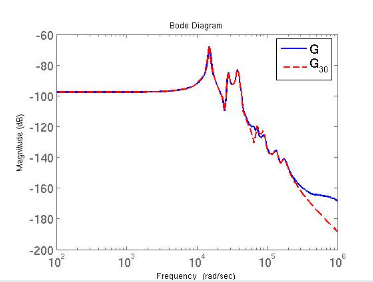
- FEM discretization of structure dynamical model using quadratic tetrahedral elements (ANSYS-SOLID187)
  $\rightsquigarrow$ $n = 34,722$, $m = 1$, $q = 12$.
- Reduced model computed using SpaRed, $r = 30$.

Solving Large-Scale Matrix Equations

Numerical example for BT: Microgyroscope (Butterfly Gyro)

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

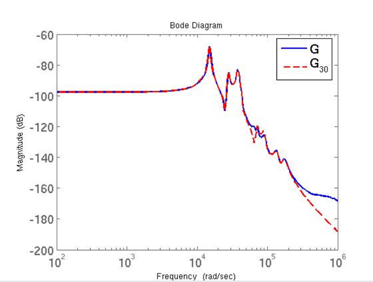
Frequency Repsonse Analysis



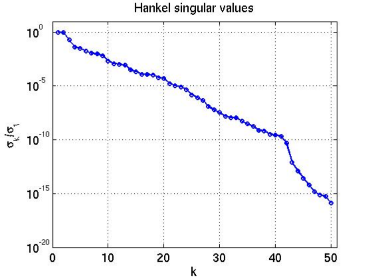
Solving Large-Scale Matrix Equations
Numerical example for BT: Microgyroscope (Butterfly Gyro)

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

Frequency Repsonse Analysis



Hankel Singular Values



Solving Large-Scale Algebraic Riccati Equations

Theory **[Lancaster/Rodman '95]**

Theorem

Consider the (continuous-time) algebraic Riccati equation (ARE)

$$0 = \mathcal{R}(X) = C^T C + A^T X + XA - XBB^T X,$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{q\times n}$, (A, B) stabilizable, (A, C) detectable.
Then:

(a) There exists a unique stabilizing $X_* \in \{X \in \mathbb{R}^{n\times n} \,|\, \mathcal{R}(X) = 0\}$, i.e., $\Lambda(A - BB^T X_*) \in \mathbb{C}^-$.

(b) $X_* = X_*^T \geq 0$ and $X_* \geq X$ for all $X \in \{X \in \mathbb{R}^{n\times n} \,|\, \mathcal{R}(X) = 0\}$.

(c) If (A, C) observable, then $X_* > 0$.

(d) $\operatorname{span}\left\{\begin{bmatrix} I_n \\ -X_* \end{bmatrix}\right\}$ is the unique invariant subspace of the Hamiltonian matrix

$$H = \begin{bmatrix} A & BB^T \\ C^T C & -A^T \end{bmatrix}$$

corresponding to $\Lambda(H) \cap \mathbb{C}^-$.

Solving Large-Scale Algebraic Riccati Equations

Numerical Methods **[Bini/Iannazzo/Meini '12]**

Numerical Methods (incomplete list)

- Invariant subspace methods ($\rightsquigarrow$ eigenproblem for Hamiltonian matrix):
  - Schur vector method (`care`) [Laub '79]
  - Hamiltonian SR algorithm [Bunse-Gerstner/Mehrmann '86]
  - Symplectic URV-based method [B./Mehrmann/Xu '97/'98, Chu/Liu/Mehrmann '07]
- Spectral projection methods
  - Sign function method [Roberts '71, Byers '87]
  - Disk function method [Bai/Demmel/Gu '94, B. '97]
- (rational, global) Krylov subspace techniques [Jaimoukha/Kasenally '94, Jbilou '03/'06, Heyouni/Jbilou '09]
- Newton's method
  - Kleinman iteration [Kleinman '68]
  - Line search acceleration [B./Byers '98]
  - Newton-ADI [B./J.-R. Li/Penzl '99/'08]
  - Inexact Newton [Feitzinger/Hylla/Sachs '09]

Solving Large-Scale Algebraic Riccati Equations

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

Solving Large-Scale Algebraic Riccati Equations

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

Solving Large-Scale Algebraic Riccati Equations

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

Solving Large-Scale Algebraic Riccati Equations

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

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:

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

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

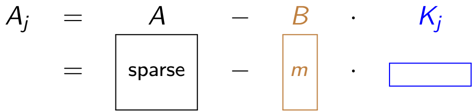
$$(A - BK_j + p_k^{(j)} I)^{-1} = (I_n + (A + p_k^{(j)} I)^{-1} B(I_m - K_j(A + p_k^{(j)} I)^{-1} B)^{-1} K_j)(A + p_k^{(j)} I)^{-1}.$$

- 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:



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

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

- 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:

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

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

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

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

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

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

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

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

Low-Rank Newton-ADI for AREs

Re-write Newton's method for AREs

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

$$\Longleftrightarrow$$

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

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

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

Factored Newton Iteration [B./Li/Penzl 1999/2008]

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 1999/2008]

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

Feedback Iteration

Optimal feedback

$$K_* = B^T X_* = B^T Z_* Z_*^T$$

can be computed by direct feedback iteration:

- jth Newton iteration:

$$K_j = B^T Z_j Z_j^T = \sum_{k=1}^{k_{\max}} (B^T V_{j,k}) V_{j,k}^T \quad \xrightarrow{j\to\infty} \quad K_* = B^T Z_* Z_*^T$$

- K_j can be updated in ADI iteration, no need to even form Z_j, need only fixed workspace for $K_j \in \mathbb{R}^{m\times n}$!

Related to earlier work by [Banks/Ito 1991].

Solving Large-Scale Matrix Equations

Galerkin-Newton-ADI

Basic ideas

- Hybrid method of Galerkin projection methods for AREs [JAIMOUKHA/KASENALLY '94, JBILOU '06, HEYOUNI/JBILOU '09] and Newton-ADI, i.e., use column space of current Newton iterate for projection, solve projected ARE, and prolongate.
- Independence of good parameters observed for Galerkin-ADI applied to Lyapunov equations $\rightsquigarrow$ fix ADI parameters for all Newton iterations.

Solving Large-Scale Matrix Equations

Galerkin-Newton-ADI

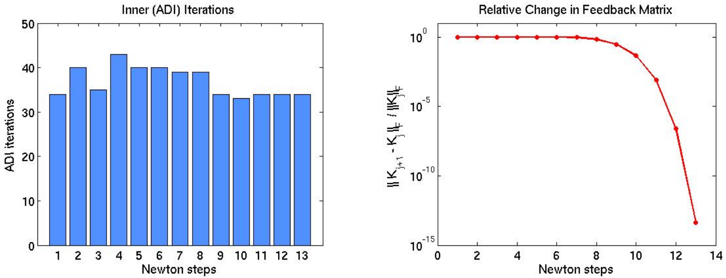
Basic ideas

- Hybrid method of Galerkin projection methods for AREs
  [JAIMOUKHA/KASENALLY '94, JBILOU '06, HEYOUNI/JBILOU '09]
  and Newton-ADI, i.e., use column space of current Newton iterate for projection, solve projected ARE, and prolongate.
- Independence of good parameters observed for Galerkin-ADI applied to Lyapunov equations $\rightsquigarrow$ fix ADI parameters for all Newton iterations.

Numerical Results

LQR Problem for 2D Geometry

- Linear 2D heat equation with homogeneous Dirichlet boundary and point control/observation.
- FD discretization on uniform 150×150 grid.
- $n = 22.500$, $m = p = 1$, 10 shifts for ADI iterations.
- Convergence of large-scale matrix equation solvers:



Numerical Results

Newton-ADI vs. Newton-ADI-Gelerkin

- FDM for 2D heat/convection-diffusion equations on $[0,1]^2$ (LYAPACK benchmarks, $m = p = 1$) $\rightsquigarrow$ symmetric/nonsymmetric $A \in \mathbb{R}^{n \times n}$, $n = 10,000$.
- 15 shifts chosen by Penzl's heuristic from 50/25 Ritz/harmonic Ritz values of A.
- Computations using Intel Core 2 Quad CPU of type Q9400 at 2.66GHz with 4 GB RAM and 64Bit-MATLAB.

Numerical Results

Newton-ADI vs. Newton-ADI-Gelerkin

- FDM for 2D heat/convection-diffusion equations on $[0,1]^2$ (LYAPACK benchmarks, $m = p = 1$) $\rightsquigarrow$ symmetric/nonsymmetric $A \in \mathbb{R}^{n \times n}$, $n = 10,000$.
- 15 shifts chosen by Penzl's heuristic from 50/25 Ritz/harmonic Ritz values of A.
- Computations using Intel Core 2 Quad CPU of type Q9400 at 2.66GHz with 4 GB RAM and 64Bit-MATLAB.

Newton-ADI

step	rel. change	rel. residual	ADI
1	1	9.99e–01	200
2	9.99e–01	3.41e+01	23
3	5.25e–01	6.37e+00	20
4	5.37e–01	1.52e+00	20
5	7.03e–01	2.64e–01	23
6	5.57e–01	1.56e–02	23
7	6.59e–02	6.30e–05	23
8	4.02e–04	9.68e–10	23
9	8.45e–09	1.09e–11	23
10	1.52e–14	1.09e–11	23

CPU time: 76.9 sec.

Numerical Results

Newton-ADI vs. Newton-ADI-Gelerkin

- FDM for 2D heat/convection-diffusion equations on $[0,1]^2$ (LYAPACK benchmarks, $m = p = 1$) $\rightsquigarrow$ symmetric/nonsymmetric $A \in \mathbb{R}^{n \times n}$, $n = 10,000$.
- 15 shifts chosen by Penzl's heuristic from 50/25 Ritz/harmonic Ritz values of A.
- Computations using Intel Core 2 Quad CPU of type Q9400 at 2.66GHz with 4 GB RAM and 64Bit-MATLAB.

Newton-ADI

step	rel. change	rel. residual	ADI
1	1	9.99e–01	200
2	9.99e–01	3.41e+01	23
3	5.25e–01	6.37e+00	20
4	5.37e–01	1.52e+00	20
5	7.03e–01	2.64e–01	23
6	5.57e–01	1.56e–02	23
7	6.59e–02	6.30e–05	23
8	4.02e–04	9.68e–10	23
9	8.45e–09	1.09e–11	23
10	1.52e–14	1.09e–11	23

CPU time: 76.9 sec.

Newton-Galerkin-ADI

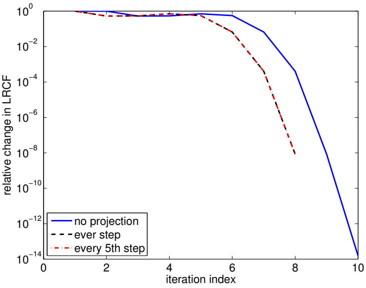
step	rel. change	rel. residual	ADI
1	1	3.56e–04	20
2	5.25e–01	6.37e+00	10
3	5.37e–01	1.52e+00	6
4	7.03e–01	2.64e–01	10
5	5.57e–01	1.57e–02	10
6	6.59e–02	6.30e–05	10
7	4.03e–04	9.79e–10	10
8	8.45e–09	1.43e–15	10

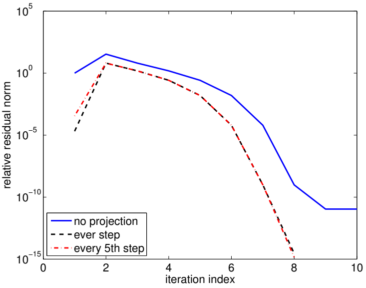
CPU time: 38.0 sec.

Numerical Results

Newton-ADI vs. Newton-ADI-Gelerkin

- FDM for 2D heat/convection-diffusion equations on $[0,1]^2$ (LYAPACK benchmarks, $m = p = 1$) $\rightsquigarrow$ symmetric/nonsymmetric $A \in \mathbb{R}^{n\times n}$, $n = 10,000$.
- 15 shifts chosen by Penzl's heuristic from 50/25 Ritz/harmonic Ritz values of A.
- Computations using Intel Core 2 Quad CPU of type Q9400 at 2.66GHz with 4 GB RAM and 64Bit-MATLAB.





Numerical Results

Newton-ADI vs. Newton-ADI-Gelerkin

- FDM for 2D heat/convection-diffusion equations on $[0,1]^2$ (LYAPACK benchmarks, $m = p = 1$) $\rightsquigarrow$ symmetric/nonsymmetric $A \in \mathbb{R}^{n \times n}$, $n = 10,000$.
- 15 shifts chosen by Penzl's heuristic from 50/25 Ritz/harmonic Ritz values of A.
- Computations using Intel Core 2 Quad CPU of type Q9400 at 2.66GHz with 4 GB RAM and 64Bit-MATLAB.

Newton-ADI

step	rel. change	rel. residual	ADI
1	1	9.99e–01	200
2	9.99e–01	3.56e+01	60
3	3.11e–01	3.72e+00	39
4	2.88e–01	9.62e–01	40
5	3.41e–01	1.68e–01	45
6	1.22e–01	5.25e–03	42
7	3.88e–03	2.96e–06	47
8	2.30e–06	6.09e–13	47

CPU time: 185.9 sec.

Numerical Results

Newton-ADI vs. Newton-ADI-Gelerkin

- FDM for 2D heat/convection-diffusion equations on $[0,1]^2$ (LYAPACK benchmarks, $m = p = 1$) $\rightsquigarrow$ symmetric/nonsymmetric $A \in \mathbb{R}^{n \times n}$, $n = 10,000$.
- 15 shifts chosen by Penzl's heuristic from 50/25 Ritz/harmonic Ritz values of A.
- Computations using Intel Core 2 Quad CPU of type Q9400 at 2.66GHz with 4 GB RAM and 64Bit-MATLAB.

Newton-ADI

step	rel. change	rel. residual	ADI
1	1	9.99e–01	200
2	9.99e–01	3.56e+01	60
3	3.11e–01	3.72e+00	39
4	2.88e–01	9.62e–01	40
5	3.41e–01	1.68e–01	45
6	1.22e–01	5.25e–03	42
7	3.88e–03	2.96e–06	47
8	2.30e–06	6.09e–13	47

CPU time: 185.9 sec.

Newton-Galerkin-ADI

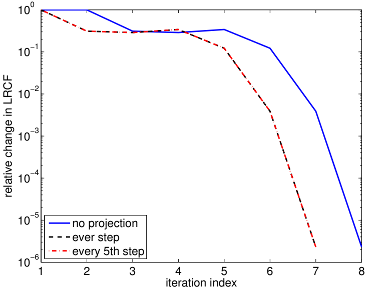
step	rel. change	rel. residual	ADI it.
1	1	1.78e–02	35
2	3.11e–01	3.72e+00	15
3	2.88e–01	9.62e–01	20
4	3.41e–01	1.68e–01	15
5	1.22e–01	5.25e–03	20
6	3.89e–03	2.96e–06	15
7	2.30e–06	6.14e–13	20

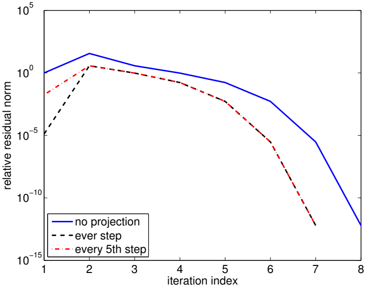
CPU time: 75.7 sec.

Numerical Results

Newton-ADI vs. Newton-ADI-Gelerkin

- FDM for 2D heat/convection-diffusion equations on $[0,1]^2$ (LYAPACK benchmarks, $m = p = 1$) $\rightsquigarrow$ symmetric/nonsymmetric $A \in \mathbb{R}^{n \times n}$, $n = 10,000$.
- 15 shifts chosen by Penzl's heuristic from 50/25 Ritz/harmonic Ritz values of A.
- Computations using Intel Core 2 Quad CPU of type Q9400 at 2.66GHz with 4 GB RAM and 64Bit-MATLAB.





Numerical Results

Example: LQR Problem for 3D Geometry

Control problem for 3d Convection-Diffusion Equation

- FDM for 3D convection-diffusion equation on $[0,1]^3$
- proposed in [SIMONCINI '07], $q = p = 1$
- non-symmetric $A \in \mathbb{R}^{n \times n}$, $n = 10\,648$

Test system:

INTEL Xeon 5160 3.00GHz ; 16 GB RAM; 64Bit-MATLAB (R2010a) using threaded BLAS; stopping tolerance: 10^{-10}

Numerical Results

Example: LQR Problem for 3D Geometry

Newton-ADI

NWT	rel. change	rel. residual	ADI
1	$1.0 \cdot 10^{0}$	$9.3 \cdot 10^{-01}$	100
2	$3.7 \cdot 10^{-02}$	$9.6 \cdot 10^{-02}$	94
3	$1.4 \cdot 10^{-02}$	$1.1 \cdot 10^{-03}$	98
4	$3.5 \cdot 10^{-04}$	$1.0 \cdot 10^{-07}$	97
5	$6.4 \cdot 10^{-08}$	$1.3 \cdot 10^{-10}$	97
6	$7.5 \cdot 10^{-16}$	$1.3 \cdot 10^{-10}$	97

CPU time: 4 805.8 sec.

NG-ADI inner= 5, outer= 1

NWT	rel. change	rel. residual	ADI
1	$1.0 \cdot 10^{0}$	$5.0 \cdot 10^{-11}$	80

CPU time: 497.6 sec.

NG-ADI inner= 1, outer= 1

NWT	rel. change	rel. residual	ADI
1	$1.0 \cdot 10^{0}$	$7.4 \cdot 10^{-11}$	71

CPU time: 856.6 sec.

NG-ADI inner= 0, outer= 1

NWT	rel. change	rel. residual	ADI
1	$1.0 \cdot 10^{0}$	$6.5 \cdot 10^{-13}$	100

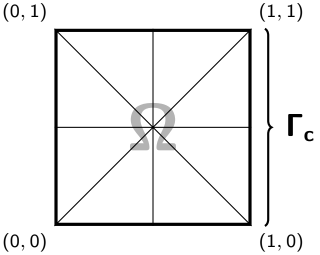
CPU time: 506.6 sec.

Test system:

INTEL Xeon 5160 3.00GHz ; 16 GB RAM; 64Bit-MATLAB (R2010a) using threaded BLAS; stopping tolerance: 10^{-10}

Numerical Results

Scaling of CPU times / Mesh Independence



$$\begin{aligned} \partial_t x(\xi, t) &= \Delta x(\xi, t) && \text{in } \Omega \\ \partial_\nu x &= b(\xi) \cdot u(t) - x && \text{on } \Gamma_c \\ \partial_\nu x &= -x && \text{on } \partial\Omega \setminus \Gamma_c \\ x(\xi, 0) &= 1 \end{aligned}$$

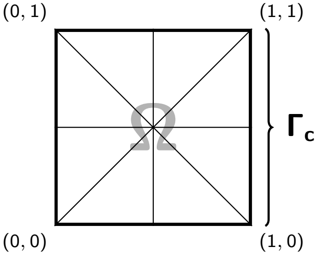
Note:

Here $b(\xi) = 4\,(1 - \xi_2)\,\xi_2$ for $\xi \in \Gamma_c$ and 0 otherwise, thus $\forall t \in \mathbb{R}_{>0}$, we have $u(t) \in \mathbb{R}$.

$$\Rightarrow B_h = M_{\Gamma,h} \cdot b.$$

Numerical Results

Scaling of CPU times / Mesh Independence



$$\begin{aligned} \partial_t x(\xi,t) &= \Delta x(\xi,t) && \text{in } \Omega \\ \partial_\nu x &= b(\xi)\cdot u(t) - x && \text{on } \Gamma_c \\ \partial_\nu x &= -x && \text{on } \partial\Omega \setminus \Gamma_c \\ x(\xi,0) &= 1 \end{aligned}$$

Consider: output equation $y = Cx$, where

$$\begin{aligned} C : \mathcal{L}^2(\Omega) &\to \mathbb{R} \\ x(\xi,t) &\mapsto y(t) = \int_\Omega x(\xi,t)\,d\xi \end{aligned} \Rightarrow C_h = \underline{1} \cdot M_h.$$

Numerical Results

Scaling of CPU times / Mesh Independence

Simplified Low Rank Newton-Galerkin ADI

- generalized state space form implementation
- Penzl shifts (16/50/25) with respect to initial matrices
- projection acceleration in every outer iteration step
- projection acceleration in every 5-th inner iteration step

Test system:

INTEL Xeon 5160 @ 3.00 GHz; 16 GB RAM; 64Bit-MATLAB (R2010a) using threaded BLAS,
stopping criterion tolerances: 10^{-10}

Numerical Results

Scaling of CPU times / Mesh Independence

Computation Times

discretization level	problem size	time in seconds
3	81	$4.87 \cdot 10^{-2}$
4	289	$2.81 \cdot 10^{-1}$
5	1 089	$5.87 \cdot 10^{-1}$
6	4 225	2.63
7	16 641	$2.03 \cdot 10^{+1}$
8	66 049	$1.22 \cdot 10^{+2}$
9	263 169	$1.05 \cdot 10^{+3}$
10	1 050 625	$1.65 \cdot 10^{+4}$
11	4 198 401	$1.35 \cdot 10^{+5}$

Test system:

INTEL Xeon 5160 @ 3.00 GHz; 16 GB RAM; 64Bit-MATLAB (R2010a) using threaded BLAS,
stopping criterion tolerances: 10^{-10}

Solving Large-Scale Matrix Equations
Software

Lyapack [Penzl 2000]

MATLAB toolbox for solving

- Lyapunov equations and algebraic Riccati equations,
- model reduction and LQR problems.

Main work horse: Low-rank ADI and Newton-ADI iterations.

Solving Large-Scale Matrix Equations
Software

Lyapack [Penzl 2000]

MATLAB toolbox for solving

- Lyapunov equations and algebraic Riccati equations,
- model reduction and LQR problems.

Main work horse: Low-rank ADI and Newton-ADI iterations.

M.E.S.S. – Matrix Equations Sparse Solvers [B./Köhler/Saak '08–]

- Extended and revised version of LYAPACK.
- Includes solvers for large-scale differential Riccati equations (based on Rosenbrock and BDF methods).
- Many algorithmic improvements:
  - new ADI parameter selection,
  - column compression based on RRQR,
  - more efficient use of direct solvers,
  - treatment of generalized systems without factorization of the mass matrix,
  - new ADI versions avoiding complex arithmetic etc.
- C and MATLAB versions.

Solving Large-Scale Matrix Equations
Software

Lyapack [Penzl 2000]

MATLAB toolbox for solving

- Lyapunov equations and algebraic Riccati equations,
- model reduction and LQR problems.

Main work horse: Low-rank ADI and Newton-ADI iterations.

M.E.S.S. – Matrix Equations Sparse Solvers [B./Köhler/Saak '08–]

- Extended and revised version of LYAPACK.
- Includes solvers for large-scale differential Riccati equations (based on Rosenbrock and BDF methods).
- Many algorithmic improvements:
  - new ADI parameter selection,
  - column compression based on RRQR,
  - more efficient use of direct solvers,
  - treatment of generalized systems without factorization of the mass matrix,
  - new ADI versions avoiding complex arithmetic etc.
- C and MATLAB versions.

Solving Large-Scale Matrix Equations
Software

Lyapack [Penzl 2000]

MATLAB toolbox for solving

- Lyapunov equations and algebraic Riccati equations,
- model reduction and LQR problems.

Main work horse: Low-rank ADI and Newton-ADI iterations.

M.E.S.S. – Matrix Equations Sparse Solvers [B./Köhler/Saak '08–]

- Extended and revised version of LYAPACK.
- Includes solvers for large-scale differential Riccati equations (based on Rosenbrock and BDF methods).
- Many algorithmic improvements:
  - new ADI parameter selection,
  - column compression based on RRQR,
  - more efficient use of direct solvers,
  - treatment of generalized systems without factorization of the mass matrix,
  - new ADI versions avoiding complex arithmetic etc.
- C and MATLAB versions.

Topics Not Covered

- Extensions to bilinear and stochastic systems.
- Rational interpolation methods for nonlinear systems.
- Other MOR techniques like POD, RB.
- MOR methods for discrete-time systems.
- Extensions to descriptor systems $E\dot{x} = Ax + Bu$, E singular.
- Parametric model reduction:

$$\dot{x} = A(p)x + B(p)u, \quad y = C(p)x,$$

  where $p \in \mathbb{R}^d$ is a free parameter vector; parameters should be preserved in the reduced-order model.

Further Reading — Model Order Reduction

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, 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.

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

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

8. W.H.A. Schilders, H.A. van der Vorst, and J. Rommes (editors).
Model Order Reduction: Theory, Research Aspects and Applications.
Mathematics in Industry, Vol. 13,
Springer-Verlag, Berlin/Heidelberg, 2008.

9. P. Benner, J. ter Maten, and M. Hinze (editors).
Model Reduction for Circuit Simulation.
Lecture Notes in Electrical Engineering, Vol. 74,
Springer-Verlag, Dordrecht, 2011.

Further Reading — Matrix Equations

1. V. Mehrmann.
The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution.
Number 163 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Heidelberg, July 1991.

2. P. Lancaster and L. Rodman.
The Algebraic Riccati Equation.
Oxford University Press, Oxford, 1995.

3. P. Benner.
Computational methods for linear-quadratic optimization
RENDICONTI DEL CIRCOLO MATEMATICO DI PALERMO, Supplemento, Serie II, 58:21–56, 1999.

4. T. Penzl.
LYAPACK Users Guide.
Technical Report SFB393/00-33, Sonderforschungsbereich 393 *Numerische Simulation auf massiv parallelen Rechnern*, TU Chemnitz, 09107 Chemnitz, FRG, 2000.
Available from `http://www.tu-chemnitz.de/sfb393/sfb00pr.html`.

5. H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank.
Matrix Riccati Equations in Control and Systems Theory.
Birkhäuser, Basel, Switzerland, 2003.

6. P. Benner.
Solving large-scale control problems.
IEEE CONTROL SYSTEMS MAGAZINE, 24(1):44–59, 2004.

7. D. Bini, B. Iannazzo, and B. Meini.
Numerical Solution of Algebraic Riccati Equations.
SIAM, Philadelphia, PA, 2012.

8. P. Benner and J. Saak.
Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey.
GAMM-MITTEILUNGEN, 36(1):32-–52, 2013.

9. V. Simoncini.
Computational methods for linear matrix equations (survey article).
March 2013.
`http://www.dm.unibo.it/~simoncin/matrixeq.pdf`.