Otto-von-Guericke Universität Magdeburg Faculty of Mathematics Summer term 2015

Model Reduction for Dynamical Systems

— Lecture 1 —

Peter Benner Lihong Feng

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

benner@mpi-magdeburg.mpg.de feng@mpi-magdeburg.mpg.de

www.mpi-magdeburg.mpg.de/2909616/mor_ss15





- Model Reduction for Dynamical Systems
- Application Areas
- Motivating Examples

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

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

with

• states
$$x(t) \in \mathbb{R}^n$$

• inputs
$$u(t) \in \mathbb{R}^m$$

• outputs $y(t) \in \mathbb{R}^q$.



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.

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)

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

- 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.

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)

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

- 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.

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 $y(t) \in \mathbb{R}^q$.

 \sum

Reduced-Order Model (ROM)

 $\widehat{\Sigma}$

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

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

٧

Model Reduction for Dynamical Systems Parameter-Dependent Dynamical Systems

Dynamical Systems

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

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

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

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, Time-Invariant (LTI) Systems

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

Model Reduction for Dynamical Systems

Linear, Time-Invariant (LTI) Systems

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

Linear, Time-Invariant Parametric Systems

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

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 ${\sim}1960 \text{ies}$



- Resolving complex 3D geometries \Rightarrow millions of degrees of freedom.
- Analysis of elastic deformations requires many simulation runs for varying external forces, in particular if the model is used in an (elastic) multi-body simulation ((E)MBS).

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

Application Areas Structural Mechanics / Finite Element Modeling

since ${\sim}1960 \text{ies}$



- Resolving complex 3D geometries \Rightarrow millions of degrees of freedom.
- Analysis of elastic deformations requires many simulation runs for varying external forces, in particular if the model is used in an (elastic) multi-body simulation ((E)MBS).

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

Application Areas (Optimal) Control

since ${\sim}1980 \text{ies}$





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

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

 \implies reduce order of plant (*n*) and/or controller (*N*).

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

Application Areas (Optimal) Control

Feedback Controllers

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

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

 $\begin{array}{l} \mbox{Modern (LQG-}/\mathcal{H}_{2^{-}}/\mathcal{H}_{\infty}\text{-}) \mbox{ control} \\ \mbox{design: } N \geq n. \end{array}$



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

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

 \implies reduce order of plant (*n*) and/or controller (*N*).

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

Application Areas (Optimal) Control

since ${\sim}1980 \text{ies}$

Feedback Controllers A feedback controller (dynamic compensator) is a linear system of order N, where • input = output of plant, • output = input of plant. Modern (LQG- $/H_2$ - $/H_\infty$ -) control design: $N \ge n$.



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

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

 \implies reduce order of plant (*n*) and/or controller (*N*).

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

Max Planck Institute Magdeburg

Application Areas (Optimal) Control



v' = E v + F yu = H v + K y

since \sim 1980ies

Feedback Controllers

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

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

 $\begin{array}{l} \mbox{Modern (LQG-}/\mathcal{H}_{2^{-}}/\mathcal{H}_{\infty}\text{-}) \mbox{ control} \\ \mbox{design: } N \geq n. \end{array}$



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

 \implies reduce order of plant (*n*) and/or controller (*N*).

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

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

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/semiconductor equations.

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/semiconductor 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, 45 <i>nm</i> technology
2,300 transistors	> 8, 200, 000 transistors
64 kHz clock speed	> 3 GHz clock speed.

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/semiconductor 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.

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/semiconductor 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).

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/semiconductor equations.

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

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/semiconductor equations.

 \rightsquigarrow 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,
- . . .
- **Current trend:** more and more multi-physics problems, i.e., coupling of several field equations, e.g.,
 - electro-thermal (e.g., bondwire heating in chip design),
 - fluid-structure-interaction,
 - . . .



Peter Benner and Lihong Feng.

Model Order Reduction for Coupled Problems

Applied and Computational Mathematics: An International Journal, 14(1):3-22, 2015. Available from http://www2.mpi-magdeburg.mpg.de/preprints/2015/MPIMD15-02.pdf.

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,
- . . .
- Current trend: more and more multi-physics problems, i.e., coupling of several field equations, e.g.,
 - electro-thermal (e.g., bondwire heating in chip design),
 - fluid-structure-interaction,
 - . . .

Peter Benner and Lihong Feng

Model Order Reduction for Coupled Problems

Applied and Computational Mathematics: An International Journal, 14(1):3-22, 2015. Available from http://www2.mpi-magdeburg.mpg.de/preprints/2015/MPIMD15-02.pdf

Motivating Examples Electro-Thermic Simulation of Integrated Circuit (IC)





- Conservative thermic sub-system in SIMPLORER: voltage ~→ temperature, current ~→ heat flow.
- Original model: n = 270.593, m = q = 2 ⇒ Computing time (on Intel Xeon dualcore 3GHz, 1 Thread):
 - Main computational cost for set-up data $\approx 22 min$.
 - 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.

Motivating Examples Electro-Thermic Simulation of Integrated Circuit (IC)

- Original model: n = 270.593, m = q = 2 ⇒
 Computing time (on Intel Xeon dualcore 3GHz, 1 Thread):
 - Main computational cost for set-up data $\approx 22 min$.
 - 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 , <1 min for reduced system.



Motivating Examples Electro-Thermic Simulation of Integrated Circuit (IC)

- Original model: n = 270.593, m = q = 2 ⇒
 Computing time (on Intel Xeon dualcore 3GHz, 1 Thread):
 - Main computational cost for set-up data $\approx 22 min$.
 - 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 , $<1 \mathrm{min}$ for reduced system.



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

$$egin{aligned} & v(x,0) = 0, & w(x,0) = 0, & x \in [0,1] \\ & v_x(0,t) = -i_0(t), & v_x(1,t) = 0, & t \geq 0, \end{aligned}$$

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]

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

$$w_t(x,t) = hv(x,t) - \gamma w(x,t) + g_y$$

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

$$egin{aligned} & v(x,0) = 0, & w(x,0) = 0, & x \in [0,1] \\ & v_x(0,t) = -i_0(t), & v_x(1,t) = 0, & t \geq 0, \end{aligned}$$

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)



- 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

• Application: inertial navigation.



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

$$\begin{array}{lll} \mathcal{M}(d) &=& \mathcal{M}_1 + d\mathcal{M}_2, \\ \mathcal{D}(\Phi, d, \alpha, \beta) &=& \Phi(\mathcal{D}_1 + d\mathcal{D}_2) + \alpha \mathcal{M}(d) + \beta \mathcal{T}(d), \\ \mathcal{T}(d) &=& \mathcal{T}_1 + \frac{1}{d} \mathcal{T}_2 + d\mathcal{T}_3, \end{array}$$

with

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



Motivating Examples Parametric MOR: Applications in Microsystems/MEMS Design

