MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

20 YEARS
1998-2018

System-Theoretic Model Reduction for Nonlinear (Parametric) Systems

Peter Benner

Joint work with
Pawan Goyal (MPI, Magdeburg, Germany)
Igor Duff Pontes (MPI, Magdeburg, Germany)

ICIAM, Valencia, Spain
July 19, 2019

Outine

1. Introduction
2. Nonlinear Systems
3. Balanced Truncation
4. Interpolation-Based Method
5. Numerical Example
6. Outlook

Physical systems

Introduction

–Reduced-order modeling motivation–

Physical systems

Circuits laws, mechanics, etc.

Partial differential equations

e.g., Navier-Stokes equations

$$\dot{v} + (v \cdot \nabla)v - \frac{1}{\mathrm{Re}}\Delta v + \nabla p = 0,$$
$$\nabla \cdot v = 0.$$

Introduction

–Reduced-order modeling motivation–

Physical systems

Using expert knowledge, experimental data

Circuits laws, mechanics, etc.

Differential (algebraic) equations

$$E\dot{x}(t) = f(x(t), u(t)),$$
$$y(t) = g(x(t), u(t)).$$

Partial differential equations

e.g., Navier-Stokes equations

$$\dot{v} + (v \cdot \nabla)v - \frac{1}{\mathrm{Re}}\Delta v + \nabla p = 0,$$
$$\nabla \cdot v = 0.$$

Introduction

–Reduced-order modeling motivation–

Physical systems

Using expert knowledge, experimental data

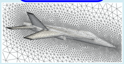
Circuits laws, mechanics, etc.

Partial differential equations

e.g., Navier-Stokes equations

$$\dot{v} + (v \cdot \nabla)v - \frac{1}{\mathrm{Re}}\Delta v + \nabla p = 0,$$
$$\nabla \cdot v = 0.$$

Discretize



Differential (algebraic) equations

$$E\dot{x}(t) = f(x(t), u(t)),$$
$$y(t) = g(x(t), u(t)).$$

Physical systems

Using expert knowledge, experimental data

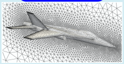
Circuits laws, mechanics, etc.

Partial differential equations

e.g., Navier-Stokes equations

$$\dot{v} + (v \cdot \nabla)v - \frac{1}{\mathsf{Re}}\Delta v + \nabla p = 0,$$
$$\nabla \cdot v = 0.$$

Discretize



Differential (algebraic) equations

$$E\dot{x}(t) = f(x(t), u(t)),$$
$$y(t) = g(x(t), u(t)).$$

simulations, control, optimization, UQ, etc.

Physical systems

Using expert knowledge, experimental data

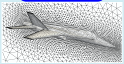
Circuits laws, mechanics, etc.

Partial differential equations

e.g., Navier-Stokes equations

$$\dot{v} + (v \cdot \nabla)v - \frac{1}{\mathsf{Re}}\Delta v + \nabla p = 0,$$
$$\nabla \cdot v = 0.$$

Discretize



Differential (algebraic) equations

$$E\dot{x}(t) = f(x(t), u(t)),$$
$$y(t) = g(x(t), u(t)).$$

simulations, control, optimization, UQ, etc.

Large-scale system

Introduction

–Reduced-order modeling motivation–

Physical systems

Using expert knowledge, experimental data

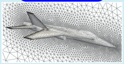
Circuits laws, mechanics, etc.

Partial differential equations

e.g., Navier-Stokes equations

$$\dot{v} + (v \cdot \nabla)v - \frac{1}{\mathrm{Re}}\Delta v + \nabla p = 0,$$
$$\nabla \cdot v = 0.$$

Discretize



Differential (algebraic) equations

$$E\dot{x}(t) = f(x(t), u(t)),$$
$$y(t) = g(x(t), u(t)).$$

~~simulations, control, optimization, UQ, etc.~~

Large-scale system

Physical systems

Using expert knowledge, experimental data

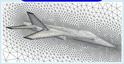
Circuits laws, mechanics, etc.

Partial differential equations

e.g., Navier-Stokes equations

$$\dot{v} + (v \cdot \nabla)v - \frac{1}{\mathrm{Re}}\Delta v + \nabla p = 0,$$
$$\nabla \cdot v = 0.$$

Discretize



Differential (algebraic) equations

$$E\dot{x}(t) = f(x(t), u(t)),$$
$$y(t) = g(x(t), u(t)).$$

~~simulations, control, optimization, UQ, etc.~~

Large-scale system

Goal

Reduce numerical complexity

Physical systems

Using expert knowledge, experimental data

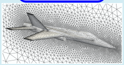
Circuits laws, mechanics, etc.

Partial differential equations

e.g., Navier-Stokes equations

$$\dot{v} + (v \cdot \nabla)v - \frac{1}{\mathrm{Re}}\Delta v + \nabla p = 0,$$
$$\nabla \cdot v = 0.$$

Discretize



Differential (algebraic) equations

$$E\dot{x}(t) = f(x(t), u(t)),$$
$$y(t) = g(x(t), u(t)).$$

~~simulations, control, optimization, UQ, etc.~~

Large-scale system

Goal

Reduce numerical complexity

MOR

A solution

Reduced-order model

Introduction

–Reduced-order modeling motivation–

Physical systems

Using expert knowledge, experimental data

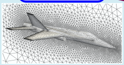
Circuits laws, mechanics, etc.

Partial differential equations

e.g., Navier-Stokes equations

$$\dot{v} + (v \cdot \nabla)v - \frac{1}{\mathrm{Re}}\Delta v + \nabla p = 0,$$
$$\nabla \cdot v = 0.$$

Discretize



Differential (algebraic) equations

$$E\dot{x}(t) = f(x(t), u(t)),$$
$$y(t) = g(x(t), u(t)).$$

simulations, control, optimization, etc.

Large-scale system

Goal

Reduce numerical complexity

MOR

A solution

Reduced-order model

Nonlinear Systems

Nonlinear systems

In this talk, we consider nonlinear systems with polynomial terms:

$$E\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t), \quad x(0) = 0.$$

Nonlinear systems

In this talk, we consider nonlinear systems with polynomial terms:

$$
\begin{aligned}
E\dot{x}(t) &= Ax(t) + Bu(t) + \sum_{\xi=2}^{d} H_\xi x^{\textcircled{\xi}}(t) + \sum_{\eta=2}^{d} N_\eta \left(u(t) \otimes x^{\textcircled{\eta}}(t)\right),\\
y(t) &= Cx(t), \quad x(0) = 0.
\end{aligned}
$$

Nonlinear Systems

Nonlinear systems

In this talk, we consider nonlinear systems with polynomial terms:

$$E\dot{x}(t) = Ax(t) + Bu(t) + \sum_{\xi=2}^{d} H_\xi x^{\textcircled{\xi}}(t) + \sum_{\eta=2}^{d} N_\eta \left(u(t) \otimes x^{\textcircled{\eta}}(t)\right),$$
$$y(t) = Cx(t), \quad x(0) = 0.$$

- d is the **degree of the polynomial term** in the system,
- (generalized) states $x(t) \in \mathbb{R}^n$, $x^{\textcircled{g}} := \underbrace{x(t) \otimes \cdots \otimes x(t)}_{g-\text{times}}$,
- inputs (controls) $u(t) \in \mathbb{R}^m$,
- outputs (measurements, quantities of interest) $y(t) \in \mathbb{R}^q$.

Nonlinear Systems

Nonlinear systems

In this talk, we consider nonlinear systems with polynomial terms:

$$E\dot{x}(t) = Ax(t) + Bu(t) + \sum_{\xi=2}^{d} H_\xi x^{\textcircled{\xi}}(t) + \sum_{\eta=2}^{d} N_\eta \left(u(t) \otimes x^{\textcircled{\eta}}(t)\right),$$
$$y(t) = Cx(t), \quad x(0) = 0.$$

- d is the **degree of the polynomial term** in the system,
- (generalized) states $x(t) \in \mathbb{R}^n$, $x^{\textcircled{g}} := \underbrace{x(t) \otimes \cdots \otimes x(t)}_{g-\text{times}}$,
- inputs (controls) $u(t) \in \mathbb{R}^m$,
- outputs (measurements, quantities of interest) $y(t) \in \mathbb{R}^q$.

- It is possible to rewrite a large class of nonlinear systems into polynomial systems,

Nonlinear Systems

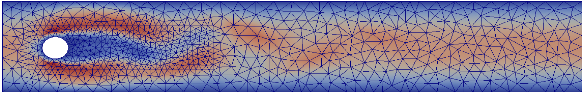
Nonlinear systems

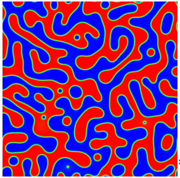
In this talk, we consider nonlinear systems with polynomial terms:

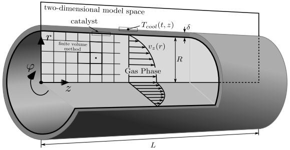
$$E\dot{x}(t) = Ax(t) + Bu(t) + \sum_{\xi=2}^{d} H_\xi x^{\textcircled{\scriptsize{ξ}}}(t) + \sum_{\eta=2}^{d} N_\eta \left(u(t) \otimes x^{\textcircled{\scriptsize{η}}}(t)\right),$$
$$y(t) = Cx(t), \quad x(0) = 0.$$

- d is the **degree of the polynomial term** in the system,
- (generalized) states $x(t) \in \mathbb{R}^n$, $x^{\textcircled{\scriptsize{g}}} := \underbrace{x(t) \otimes \cdots \otimes x(t)}_{g-\text{times}}$,
- inputs (controls) $u(t) \in \mathbb{R}^m$,
- outputs (measurements, quantities of interest) $y(t) \in \mathbb{R}^q$.

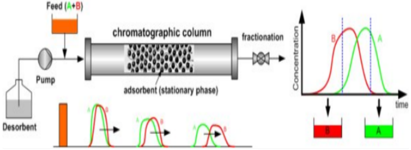
- It is possible to rewrite a large class of nonlinear systems into polynomial systems,
- via McCormick Relaxation $\rightsquigarrow$ **no approximation**. [McCormick '76, Gu '09]



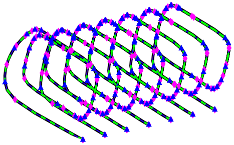

*



Nonlinear systems



Batch Chromatography



---

* Courtesy of [Hawick/Playne '10]

Transformation of Nonlinear Systems

A nonlinear system

$$\dot{x}_1(t) = -x_1(t) + x_2^3(t) + e^{-x_2(t)}, \tag{1a}$$

$$\dot{x}_2(t) = -x_1(t) + u(t). \tag{1b}$$

A nonlinear system

$$\dot{x}_1(t) = -x_1(t) + x_2^3(t) + e^{-x_2(t)}, \tag{1a}$$

$$\dot{x}_2(t) = -x_1(t) + u(t). \tag{1b}$$

- To write the system (1) in the polynomial form, we define $z_1(t) := e^{-x_2(t)}$.

Transformation of Nonlinear Systems

A nonlinear system

$$\dot{x}_1(t) = -x_1(t) + x_2^3(t) + e^{-x_2(t)}, \tag{1a}$$
$$\dot{x}_2(t) = -x_1(t) + u(t). \tag{1b}$$

- To write the system (1) in the polynomial form, we define $z_1(t) := e^{-x_2(t)}$.
- Then, we differentiate $z_1(t)$:

$$\dot{z}_1(t) = -e^{-x_2(t)} \cdot \dot{x}_2(t) = -z_1(t)\left(-x_1(t) + u(t)\right).$$

A nonlinear system

$$\dot{x}_1(t) = -x_1(t) + x_2^3(t) + e^{-x_2(t)}, \tag{1a}$$
$$\dot{x}_2(t) = -x_1(t) + u(t). \tag{1b}$$

- To write the system (1) in the polynomial form, we define $z_1(t) := e^{-x_2(t)}$.
- Then, we differentiate $z_1(t)$:

$$\dot{z}_1(t) = -e^{-x_2(t)} \cdot \dot{x}_2(t) = -z_1(t)\left(-x_1(t) + u(t)\right).$$

An equivalent polynomial system

$$\dot{x}_1(t) = -x_1(t) + x_2^3(t) + z_1(t),$$

Transformation of Nonlinear Systems

A nonlinear system

$$\dot{x}_1(t) = -x_1(t) + x_2^3(t) + e^{-x_2(t)}, \tag{1a}$$
$$\dot{x}_2(t) = -x_1(t) + u(t). \tag{1b}$$

- To write the system (1) in the polynomial form, we define $z_1(t) := e^{-x_2(t)}$.
- Then, we differentiate $z_1(t)$:

$$\dot{z}_1(t) = -e^{-x_2(t)} \cdot \dot{x}_2(t) = -z_1(t)\left(-x_1(t) + u(t)\right).$$

An equivalent polynomial system

$$\dot{x}_1(t) = -x_1(t) + x_2^3(t) + z_1(t),$$
$$\dot{x}_2(t) = -x_1(t) + u(t),$$

A nonlinear system

$$\dot{x}_1(t) = -x_1(t) + x_2^3(t) + e^{-x_2(t)}, \tag{1a}$$
$$\dot{x}_2(t) = -x_1(t) + u(t). \tag{1b}$$

- To write the system (1) in the polynomial form, we define $z_1(t) := e^{-x_2(t)}$.
- Then, we differentiate $z_1(t)$:

$$\dot{z}_1(t) = -e^{-x_2(t)} \cdot \dot{x}_2(t) = -z_1(t)\left(-x_1(t) + u(t)\right).$$

An equivalent polynomial system

$$\begin{aligned}
\dot{x}_1(t) &= -x_1(t) + x_2^3(t) + z_1(t),\\
\dot{x}_2(t) &= -x_1(t) + u(t),\\
\dot{z}_1(t) &= x_1(t)z_1(t) - z_1(t)u(t).
\end{aligned}$$

Construction of ROMs

Full-order system

$$E\dot{x}(t) = Ax(t) + \sum_{\xi=2}^{d} H_\xi x^{\textcircled{\xi}}(t)(t) + \sum_{\eta=1}^{d} N_\eta \left(u(t) \otimes x^{\textcircled{\eta}}(t)\right) + Bu(t),$$

$$y(t) = Cx(t), \quad x(0) = 0,$$

Construction of ROMs

Full-order system

$$E\dot{x}(t) = Ax(t) + \sum_{\xi=2}^{d} H_\xi x^{\textcircled{\xi}}(t)(t) + \sum_{\eta=1}^{d} N_\eta \left(u(t) \otimes x^{\textcircled{\eta}}(t)\right) + Bu(t),$$

$$y(t) = Cx(t), \quad x(0) = 0,$$

Petrov-Galerkin projection

Reduced-order system

$$\hat{E}\dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \sum_{\xi=2}^{d} \hat{H}_\xi \hat{x}^{\textcircled{\xi}}(t)(t) + \sum_{\eta=1}^{d} \hat{N}_\eta \left(u(t) \otimes \hat{x}^{\textcircled{\eta}}(t)\right) + \hat{B}u(t),$$

$$\hat{y}(t) = \hat{C}\hat{x}(t), \quad \hat{x}(0) = 0,$$

$$\hat{E} = \mathbf{W}^T E V, \quad \hat{A} = \mathbf{W}^T A\mathbf{V}, \quad \hat{H}_\xi = \mathbf{W}^T H_\xi \mathbf{V}^{\textcircled{\xi}}, \quad \xi \in \{2, \ldots, d\},$$
$$\hat{B} = \mathbf{W}^T B, \quad \hat{C} = C\mathbf{V}, \quad \hat{N}_\eta = \mathbf{W}^T N_\eta \mathbf{V}^{\textcircled{\eta}}, \quad \eta \in \{1, \ldots, d\}.$$

Existing Approaches

Snapshot-based methods

- Proper orthogonal decomposition, e.g., [Volkwein '08]
- Reduced basis methods, e.g., [Quarteroni et al. '16]
- Non-intrusive reduced-order modeling. [Peherstorfer/Willcox '16]

Existing Approaches

Snapshot-based methods

- Proper orthogonal decomposition, e.g., [Volkwein '08]
- Reduced basis methods, e.g., [Quarteroni et al. '16]
- Non-intrusive reduced-order modeling. [Peherstorfer/Willcox '16]

System-theoretic methods

Do not require simulations of full-order systems, but rather need a state-space representation and concepts from control theory.

Existing Approaches

Snapshot-based methods

- Proper orthogonal decomposition, e.g., [Volkwein '08]
- Reduced basis methods, e.g., [Quarteroni et al. '16]
- Non-intrusive reduced-order modeling. [Peherstorfer/Willcox '16]

System-theoretic methods

Do not require simulations of full-order systems, but rather need a state-space representation and concepts from control theory.

- For **order** 2 polynomial systems (known as **quadratic-bilinear** systems)
  - Balanced truncation [B./Goyal '17]
  - Interpolation-based methods [Gu '11, B./Breiten '15, B./Goyal/Gugercin '18, Cao '19]

Existing Approaches

Snapshot-based methods

- Proper orthogonal decomposition, e.g., [Volkwein '08]
- Reduced basis methods, e.g., [Quarteroni et al. '16]
- Non-intrusive reduced-order modeling. [Peherstorfer/Willcox '16]

System-theoretic methods

Do not require simulations of full-order systems, but rather need a state-space representation and concepts from control theory.

- For **order** 2 polynomial systems (known as **quadratic-bilinear** systems)
  - Balanced truncation [B./Goyal '17]
  - Interpolation-based methods [Gu '11, B./Breiten '15, B./Goyal/Gugercin '18, Cao '19]
- **In this talk**, for **order** $d \geq 3$ polynomial systems, we present extensions of

Existing Approaches

Snapshot-based methods

- Proper orthogonal decomposition, e.g., [Volkwein '08]
- Reduced basis methods, e.g., [Quarteroni et al. '16]
- Non-intrusive reduced-order modeling. [Peherstorfer/Willcox '16]

System-theoretic methods

Do not require simulations of full-order systems, but rather need a state-space representation and concepts from control theory.

- For **order** 2 polynomial systems (known as **quadratic-bilinear** systems)
  - Balanced truncation [B./Goyal '17]
  - Interpolation-based methods [Gu '11, B./Breiten '15, B./Goyal/Gugercin '18, Cao '19]
- **In this talk**, for **order** $d \geq 3$ polynomial systems, we present extensions of
  - Balanced truncation

Existing Approaches

Snapshot-based methods

- Proper orthogonal decomposition, e.g., [Volkwein '08]
- Reduced basis methods, e.g., [Quarteroni et al. '16]
- Non-intrusive reduced-order modeling. [Peherstorfer/Willcox '16]

System-theoretic methods

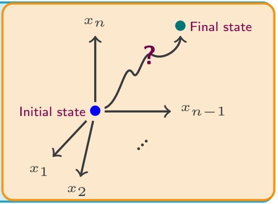
Do not require simulations of full-order systems, but rather need a state-space representation and concepts from control theory.

- For **order** 2 polynomial systems (known as **quadratic-bilinear** systems)
  - Balanced truncation [B./Goyal '17]
  - Interpolation-based methods [Gu '11, B./Breiten '15, B./Goyal/Gugercin '18, Cao '19]
- **In this talk**, for **order** $d \geq 3$ polynomial systems, we present extensions of
  - Balanced truncation
  - Interpolation-based methods

- Relies on the concept of reachability and observability.

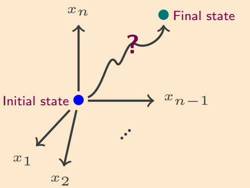
- Relies on the concept of reachability and observability.

- **Reachability:** Ability to reach a state from zero.



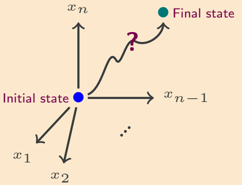
- Relies on the concept of reachability and observability.

- **Reachability:** Ability to reach a state from zero.
- **Observability:** Ability to construct the state from the observation.



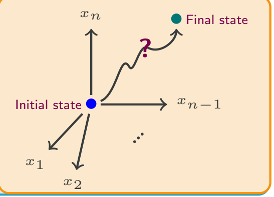
- Relies on the concept of reachability and observability.

- **Reachability:** Ability to reach a state from zero.
- **Observability:** Ability to construct the state from the observation.
- Quantified using reachability and observability **energy functionals**.



- Relies on the concept of reachability and observability.

- **Reachability:** Ability to reach a state from zero.
- **Observability:** Ability to construct the state from the observation.
- Quantified using reachability and observability **energy functionals**.



Idea of balanced truncation

- Construct state transformation, allowing to find states which are hard to reach, as well as hard to observe.
- Truncating such states yields a reduced-order system.

Balanced Truncation

–Balanced truncation for linear systems–

For linear systems

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \; x(0) = 0. \end{cases}$$

- Map between $u(t) \mapsto x(t)$:

$$x(t) = \int_0^t e^{A\sigma} B \, u(t - \sigma) d\sigma.$$

Reachability Gramian:

$$P := \int_0^{+\infty} e^{At} B (e^{At} B)^T dt.$$

- Map between $x(t) \mapsto y(t)$:

$$y(t) = Ce^{At} x_0.$$

Observability Gramian:

$$Q := \int_0^{+\infty} (Ce^{At})^T Ce^{At} dt,$$

Gramians

The controllability and observability Gramians satisfy

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

Energy functionals for linear systems

For linear systems, the **energy functionals** are given by

$$L_r(x_0) = x_0^T P^{-1} x_0, \quad L_o(x_0) = x_0^T Q x_0.$$

Energy functionals for linear systems

For linear systems, the **energy functionals** are given by

$$L_r(x_0) = x_0^T P^{-1} x_0, \;\; L_o(x_0) = x_0^T Q x_0.$$

- The Hankel singular values are given by $\{\sigma_i | i = \{1, \ldots, n\}\} = \sqrt{\Lambda\,(PQ)}$.

Energy functionals for linear systems

For linear systems, the **energy functionals** are given by

$$L_r(x_0) = x_0^T P^{-1} x_0, \;\; L_o(x_0) = x_0^T Q x_0.$$

- The Hankel singular values are given by $\{\sigma_i | i = \{1, \ldots, n\}\} = \sqrt{\Lambda\,(PQ)}$.
- Find state transformation such that $P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$.

Energy functionals for linear systems

For linear systems, the **energy functionals** are given by

$$L_r(x_0) = x_0^T P^{-1} x_0, \;\; L_o(x_0) = x_0^T Q x_0.$$

- The Hankel singular values are given by $\{\sigma_i | i = \{1, \ldots, n\}\} = \sqrt{\Lambda\left(PQ\right)}$.
- Find state transformation such that $P = Q = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$.
- Truncation of states related to small singular values

$$\|y(t) - \hat{y}(t)\|_{L_2} \leq 2 \left(\sum_{j=k+1}^{n} \sigma_j \right) \|u\|_{L_2}.$$

- Generally, exact energy functionals are given by the solutions of **nonlinear Hamilton-Jacobi equations** and **nonlinear Lyapunov**-type equations. [Scherpen '93]

- Generally, exact energy functionals are given by the solutions of **nonlinear Hamilton-Jacobi equations** and **nonlinear Lyapunov**-type equations. [Scherpen '93]
- However,
  - **infeasible** to solve for high-dimensional nonlinear systems,

Balanced Truncation

–Main Challenge and a Possible Way–

- Generally, exact energy functionals are given by the solutions of **nonlinear Hamilton-Jacobi equations** and **nonlinear Lyapunov**-type equations. [Scherpen '93]
- However,
  - **infeasible** to solve for high-dimensional nonlinear systems,
  - **hard to use** in the model reduction framework.

- Generally, exact energy functionals are given by the solutions of **nonlinear Hamilton-Jacobi equations** and **nonlinear Lyapunov**-type equations. [Scherpen '93]
- However,
  - **infeasible** to solve for high-dimensional nonlinear systems,
  - **hard to use** in the model reduction framework.
- We aim at determining the **algebraic Gramians** for polynomial systems, which
  - provide **bounds for energy functionals** of PC systems, and

Balanced Truncation

–Main Challenge and a Possible Way–

- Generally, exact energy functionals are given by the solutions of **nonlinear Hamilton-Jacobi equations** and **nonlinear Lyapunov**-type equations. [Scherpen '93]
- However,
  - **infeasible** to solve for high-dimensional nonlinear systems,
  - **hard to use** in the model reduction framework.

- We aim at determining the **algebraic Gramians** for polynomial systems, which
  - provide **bounds for energy functionals** of PC systems, and
  - allow us to **find the states** that are hard to reach and observe in an efficient way.

- Extending the Volterra series concept QB systems, we propose **the controllability Gramian**. [B./Goyal '17]
- Second step, we define an adjoint system of the polynomial system. [Fujimoto et al. '02]
- Based on it, we define the **observability Gramian**.

Theorem [B./Goyal/Pontes '19]

The **reachability Gramian** $(\mathbf{P})$ of a polynomial system solves the **polynomial Lyapunov** equation

$$AP + PA^T + BB^T + \sum_{\xi=2}^{d} H_\xi P^{\textcircled{\xi}} H_\xi^T + \sum_{\eta=1}^{d} N_\eta P^{\textcircled{\eta}} (N_\eta)^T = 0.$$

The **observability Gramian** $(\mathbf{Q})$ of a polynomial system solves the **polynomial Lyapunov** equation

$$A^T Q + QA + C^T C + \sum_{\xi=1}^{d-1} H_{\xi+1}^{(2)} \left(P^{\textcircled{\xi}} \otimes Q\right) \left(H_{\xi+1}^{(2)}\right)^T + \sum_{\eta=0}^{d-1} N_{\eta+1} \left(P^{\textcircled{\eta}} \otimes Q\right) \left(N_{\eta+1}\right)^T = 0.$$

Balanced Truncation

–Gramians and Energy Functionals–

- We show **bounds** for the **energy functionals** (at least in the neighborhood of the origin), similar to the bilinear and quadratic-bilinear case, as:

$$\boxed{L_r(x_0) \geq \frac{1}{2}x_0^T P^{-1} x_0, \qquad L_o(x_0) \leq \frac{1}{2}x_0^T Q x_0.}$$

Balanced Truncation

–Gramians and Energy Functionals–

- We show **bounds** for the **energy functionals** (at least in the neighborhood of the origin), similar to the bilinear and quadratic-bilinear case, as:

$$\boxed{L_r(x_0) \geq \frac{1}{2} x_0^T P^{-1} x_0, \qquad L_o(x_0) \leq \frac{1}{2} x_0^T Q x_0.}$$

Another interpretation of Gramians in terms of energy functionals [B./Goyal/Pontes '19]

1. Assuming zero initial condition, $\underline{x(t,0,u) \in \text{Im}P}$, $\forall\ t \geq 0$ and all input functions.

   $\Rightarrow$ If the final state $\notin \text{Im}P$, it is unreachable.

- We show **bounds** for the **energy functionals** (at least in the neighborhood of the origin), similar to the bilinear and quadratic-bilinear case, as:

$$\boxed{L_r(x_0) \geq \frac{1}{2}x_0^T P^{-1} x_0, \qquad L_o(x_0) \leq \frac{1}{2}x_0^T Q x_0.}$$

Another interpretation of Gramians in terms of energy functionals [B./Goyal/Pontes '19]

1. Assuming zero initial condition, $\underline{x(t,0,u) \in \mathrm{Im}P}$, $\forall\ t \geq 0$ and all input functions.
   $\Rightarrow$ If the final state $\notin \mathrm{Im}P$, it is unreachable.
2. If $P > 0$ and the initial state $\in \mathrm{Ker}Q$, then it is unobservable.

- Polynomial Lyapunov equations are very expensive to solve.

- Polynomial Lyapunov equations are very expensive to solve.
- We propose truncated Gramians that only involve a finite number of kernels.

Definition [B./Goyal/Pontes '19]

The truncated **reachability Gramian** $(\mathbf{P}_{\mathcal{T}})$ of a polynomial system solves the **linear Lyapunov** equation

$$AP_{\mathcal{T}} + P_{\mathcal{T}}A^T + BB^T + \sum_{\xi=2}^{d} H_\xi P_l^{\textcircled{\xi}} H_\xi^T + \sum_{\eta=1}^{d} N_\eta P_l^{\textcircled{\eta}} (N_\eta)^T = 0.$$

The truncated **observability Gramian** $(\mathbf{Q}_{\mathcal{T}})$ of a polynomial system solves the **linear Lyapunov** equation

$$A^T Q_{\mathcal{T}} + Q_{\mathcal{T}} A + C^T C + \sum_{\xi=1}^{d-1} H_{\xi+1}^{(2)} \left(P_l^{\textcircled{\xi}} \otimes Q_l\right) \left(H_{\xi+1}^{(2)}\right)^T + \sum_{\eta=0}^{d-1} N_{\eta+1}^{(2)} \left(P_l^{\textcircled{\eta}} \otimes Q_l\right) \left(N_{\eta+1}^{(2)}\right)^T = 0,$$

where $AP_l + P_l A^T + BB^T = 0$ and $A^T Q_l + Q_l A + C^T C = 0$.

- **Advantage:** Only need to solve four (linear) Lyapunov equations.

- Similar to linear and bilinear cases, **balancing** allows us to find hard to control and observe states, see, e.g., [Antoulas '05, B./Damm '08].

- Similar to linear and bilinear cases, **balancing** allows us to find hard to control and observe states, see, e.g., [Antoulas '05, B./Damm '08].

Algorithm: balanced truncation for polynomial systems

Provide system matrices $A, H_\xi, N_\eta^k, B, C,$ and order of the reduced system r (optional).

- Similar to linear and bilinear cases, **balancing** allows us to find hard to control and observe states, see, e.g., [Antoulas '05, B./Damm '08].

Algorithm: balanced truncation for polynomial systems

Provide system matrices A, H_ξ, N_η^k, B, C, and order of the reduced system r (optional).

Step 1: Compute system Gramians:

$$AP + PA^T + BB^T + \sum_{\xi=2}^{d} H_\xi P^{\text{\textcircled{ξ}}} H_\xi^T + \sum_{\eta=1}^{d} \sum_{k=1}^{m} N_\eta^k P^{\text{\textcircled{η}}} \left(N_\eta^k\right)^T = 0.$$

Low-rank factors: $P \approx SS^T$ and $Q \approx RR^T$.

Balanced Truncation
–Balancing Algorithm–

- Similar to linear and bilinear cases, **balancing** allows us to find hard to control and observe states, see, e.g., [ANTOULAS '05, B./DAMM '08].

Algorithm: balanced truncation for polynomial systems

Provide system matrices A, H_ξ, N_η^k, B, C, and order of the reduced system r (optional).

Step 1: Compute truncated system Gramians:

$$AP_\mathcal{T} + P_\mathcal{T}A^T + BB^T + \sum_{\xi=2}^{d} H_\xi P_l^{\textcircled{\xi}} H_\xi^T + \sum_{\eta=1}^{d}\sum_{k=1}^{m} N_\eta^k P_l^{\textcircled{\eta}} \left(N_\eta^k\right)^T = 0.$$

Low-rank factors: $P_\mathcal{T} \approx SS^T$ and $Q_\mathcal{T} \approx RR^T$.

$$AP_l + P_lA^T + BB^T = 0,$$
$$A^TQ_l + Q_lA + C^TC = 0.$$

$P_l \approx$, $Q_l \approx$

- Similar to linear and bilinear cases, **balancing** allows us to find hard to control and observe states, see, e.g., [Antoulas '05, B./Damm '08].

Algorithm: balanced truncation for polynomial systems

Provide system matrices A, H_ξ, N_η^k, B, C, and order of the reduced system r (optional).

Step 1: Compute truncated system Gramians:

Step 2: Determine projection matrices:

- Similar to linear and bilinear cases, **balancing** allows us to find hard to control and observe states, see, e.g., [Antoulas '05, B./Damm '08].

Algorithm: balanced truncation for polynomial systems

Provide system matrices A, H_ξ, N_η^k, B, C, and order of the reduced system r (optional).

Step 1: Compute truncated system Gramians:

Step 2: Determine projection matrices:

$$S^T R = U \Sigma V^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^T, \quad \Sigma_1 \in \mathbb{R}^{r \times r},$$

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

- Similar to linear and bilinear cases, **balancing** allows us to find hard to control and observe states, see, e.g., [Antoulas '05, B./Damm '08].

Algorithm: balanced truncation for polynomial systems

Provide system matrices A, H_ξ, N_η^k, B, C, and order of the reduced system r (optional).

Step 1: Compute truncated system Gramians:

Step 2: Determine projection matrices:

Step 3: Compute the reduced-order system:

$$\hat{A} = \mathbf{W}^T A \mathbf{V}, \qquad \hat{H}_\xi = \mathbf{W}^T H_\xi \mathbf{V}^{\textcircled{\xi}}, \qquad \hat{N}_\eta^k = \mathbf{W}^T N_\eta^k \mathbf{V}^{\textcircled{\eta}},$$

$$\hat{B} = \mathbf{W}^T B, \qquad \hat{C} = C\mathbf{V}.$$

Governing equations

$$
\begin{aligned}
\epsilon v_t(x,t) &= \epsilon^2 v_{xx}(x,t) + v(1-v)(v-0.1) - w(x,t) + q,\\
w_t(x,t) &= hv(x,t) - \gamma w(x,t) + q,\\
v(x,0) &= 0, \qquad w(x,0) = 0, \qquad x \in [0,L],\\
v_x(0,t) &= u(t), \qquad v_x(L,t) = 0, \qquad t \geq 0,
\end{aligned}
$$

where $\epsilon = 0.015,\ h = 0.5,\ \gamma = 2,\ q = 0.05,\ L = 0.3.$

- After discretization, we obtain a polynomial system (PC) with cubic nonlinearity of order $n_{pc} = 600$. [B./Breiten '15]

Balanced Truncation
–FitzHugh-Nagumo Model–

Governing equations

$$
\begin{aligned}
\epsilon v_t(x,t) &= \epsilon^2 v_{xx}(x,t) + v(1-v)(v-0.1) - w(x,t) + q,\\
w_t(x,t) &= hv(x,t) - \gamma w(x,t) + q,\\
v(x,0) &= 0, \qquad w(x,0) = 0, \qquad x \in [0,L],\\
v_x(0,t) &= u(t), \qquad v_x(L,t) = 0, \qquad t \geq 0,
\end{aligned}
$$

where $\epsilon = 0.015,\ h = 0.5,\ \gamma = 2,\ q = 0.05,\ L = 0.3.$

- After discretization, we obtain a polynomial system (PC) with cubic nonlinearity of order $n_{pc} = 600$. [B./Breiten '15]
- The transformed quadratic-bilinear (QB) system is of order $n_{qb} = 900$.

Governing equations

$$\begin{aligned}
\epsilon v_t(x,t) &= \epsilon^2 v_{xx}(x,t) + v(1-v)(v-0.1) - w(x,t) + q,\\
w_t(x,t) &= hv(x,t) - \gamma w(x,t) + q,\\
v(x,0) &= 0, \qquad w(x,0) = 0, \qquad x \in [0,L],\\
v_x(0,t) &= u(t), \qquad v_x(L,t) = 0, \qquad t \geq 0,
\end{aligned}$$

where $\epsilon = 0.015,\ h = 0.5,\ \gamma = 2,\ q = 0.05,\ L = 0.3.$

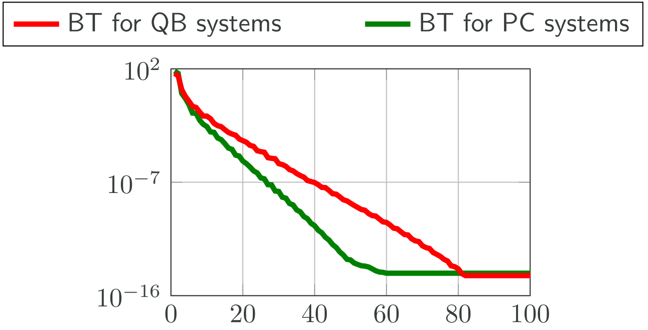
- After discretization, we obtain a polynomial system (PC) with cubic nonlinearity of order $n_{pc} = 600$. [B./Breiten '15]
- The transformed quadratic-bilinear (QB) system is of order $n_{qb} = 900$.
- The outputs of interest $v(0,t), w(0,t)$ are the responses at the left boundary at $x = 0$.

Governing equations

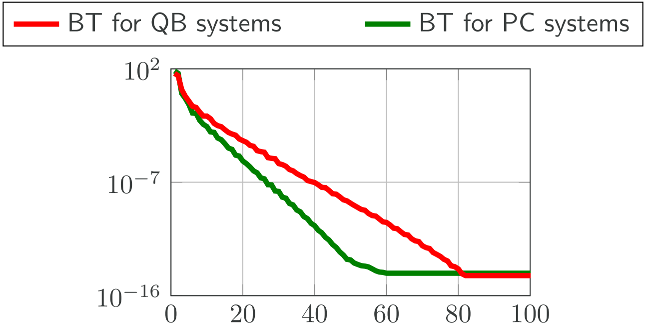
$$
\begin{aligned}
\epsilon v_t(x,t) &= \epsilon^2 v_{xx}(x,t) + v(1-v)(v-0.1) - w(x,t) + q,\\
w_t(x,t) &= hv(x,t) - \gamma w(x,t) + q,\\
v(x,0) &= 0, \qquad w(x,0) = 0, \qquad x \in [0,L],\\
v_x(0,t) &= u(t), \qquad v_x(L,t) = 0, \qquad t \geq 0,
\end{aligned}
$$

where $\epsilon = 0.015,\ h = 0.5,\ \gamma = 2,\ q = 0.05,\ L = 0.3.$

- After discretization, we obtain a polynomial system (PC) with cubic nonlinearity of order $n_{pc} = 600$. [B./Breiten '15]
- The transformed quadratic-bilinear (QB) system is of order $n_{qb} = 900$.
- The outputs of interest $v(0,t), w(0,t)$ are the responses at the left boundary at $x = 0$.
- We compare balanced truncation for PC and QB.



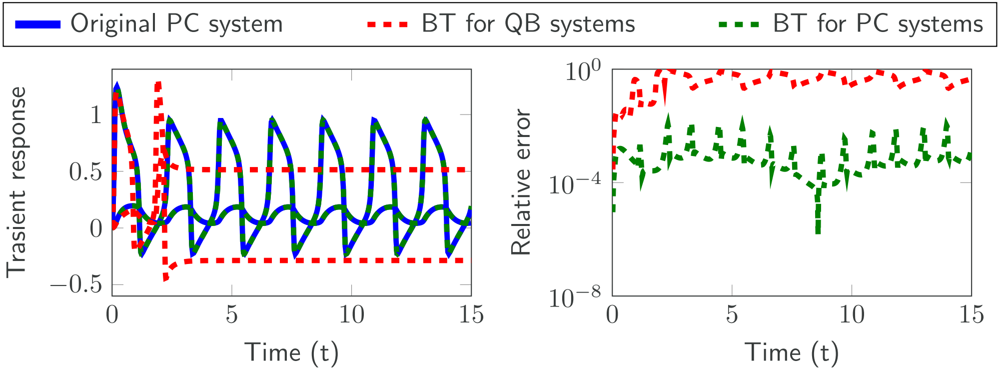
BT for QB systems
BT for PC systems
10^{2}
10^{-7}
10^{-16}
0
20
40
60
80
100



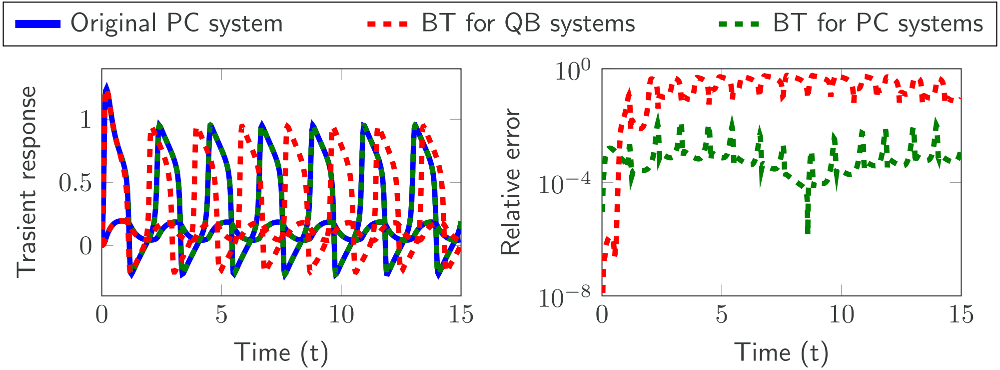
- Decay of singular values for PC systems is faster $\Rightarrow$ smaller reduced order model!

Balanced Truncation

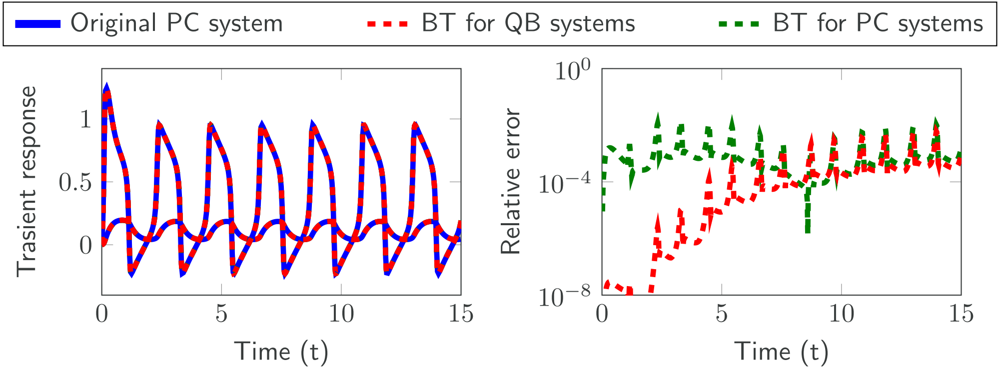
–Time-domain simulations–



- Original PC system of order 600. Original QB of order 900.
- Reduced PC system of order 10. Reduced QB system of order 10.



- Original PC system of order 600. Original QB of order 900.
- Reduced PC system of order 10. Reduced QB system of order 30.



- Original PC system of order 600. Original QB of order 900.
- Reduced PC system of order 10. Reduced QB system of order 43.

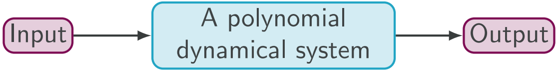
Interpolation-Based Method

–Input-output mapping–

Input $\longrightarrow$ A polynomial dynamical system $\longrightarrow$ Output

Interpolation-Based Method

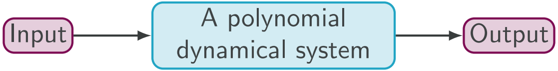
–Input-output mapping–



- As for linear systems, we can define the input-output mapping by generalized transfer functions.

Interpolation-Based Method

–Input-output mapping–



- As for linear systems, we can define the input-output mapping by generalized transfer functions.
- Instead of having a single transfer function, we have a sequence of transfer functions.

Interpolation-Based Method

–Input-output mapping–

$$
\begin{aligned}
E\dot{x}(t) &= Ax(t) + \sum_{\xi=2}^{d} H_\xi x^{\textcircled{\xi}}(t) + \sum_{\eta=2}^{d} N_\xi \left(u(t) \otimes x^{\textcircled{\eta}}(t)(t)\right) + Bu(t),\\
y(t) &= Cx(t), \quad x(0).
\end{aligned}
$$

- As for linear systems, we can define the input-output mapping by generalized transfer functions.
- Instead of having a single transfer function, we have a sequence of transfer functions.
- **Generalized transfer functions** [B./Goyal '19]

$$
\begin{aligned}
\mathbf{F}_L(s_1) &:= C\Phi(s_1)B,\\
\mathbf{F}_H^{(\xi)}(s_1,\ldots,s_{\xi+1}) &:= C\Phi(s_{\xi+1})H_\xi\left(\Phi(s_\xi)B \otimes \cdots \otimes \Phi(s_1B)\right),\\
\mathbf{F}_N^{(\eta)}(s_1,\ldots,s_{\eta+1}) &:= C\Phi(s_{\eta+1})N_\eta\left(I_m \otimes \Phi(s_\eta)B \otimes \cdots \otimes \Phi(s_1)B\right),
\end{aligned}
$$

where $\Phi(s) := (sE - A)^{-1}$.

Interpolating Reduced System

Goal [B./Goyal '19]

Construct projection matrices $\mathbf{V}$ and $\mathbf{W}$ such that

$$(\mathsf{GTF})^{\sigma}_{\mathsf{original}} = (\mathsf{GTF})^{\sigma}_{\mathsf{reduced}},$$

GTF = Generalized transfer functions

and reduced matrices are constructed via Petro-Galerkin projection:

$$\begin{aligned}
\hat{E} &= \mathbf{W}^T A\mathbf{V}, & \hat{A} &= \mathbf{W}^T A\mathbf{V}, & \hat{H}_\xi &= \mathbf{W}^T H_\xi \mathbf{V}^{\circledS}, & \xi &\in \{2,\ldots,d\},\\
\hat{B} &= \mathbf{W}^T B, & \hat{C} &= C\mathbf{V}, & \hat{N}_\eta &= \mathbf{W}^T H_\eta \mathbf{V}^{\textcircled{\eta}}, & \eta &\in \{1,\ldots,d\}.
\end{aligned}$$

Interpolating Reduced System

Goal [B./Goyal '19]

Construct projection matrices $\mathbf{V}$ and $\mathbf{W}$ such that

$$(\mathsf{GTF})^{\sigma}_{\mathsf{original}} = (\mathsf{GTF})^{\sigma}_{\mathsf{reduced}},$$

GTF = Generalized transfer functions

and reduced matrices are constructed via Petro-Galerkin projection:

$$\begin{aligned} \hat{E} &= \mathbf{W}^T A\mathbf{V}, \quad \hat{A} = \mathbf{W}^T A\mathbf{V}, \quad \hat{H}_\xi = \mathbf{W}^T H_\xi \mathbf{V}^{\textcircled{\xi}}, \quad \xi \in \{2,\ldots,d\}, \\ \hat{B} &= \mathbf{W}^T B, \quad \hat{C} = C\mathbf{V}, \quad \hat{N}_\eta = \mathbf{W}^T H_\eta \mathbf{V}^{\textcircled{\eta}}, \quad \eta \in \{1,\ldots,d\}. \end{aligned}$$

- Extended ideas from linear systems to polynomial systems.

Interpolating Reduced System

Goal [B./Goyal '19]

Construct projection matrices $\mathbf{V}$ and $\mathbf{W}$ such that

$$(\mathsf{GTF})^{\sigma}_{\mathsf{original}} = (\mathsf{GTF})^{\sigma}_{\mathsf{reduced}},$$

GTF = Generalized transfer functions

and reduced matrices are constructed via Petro-Galerkin projection:

$$\begin{aligned} \hat{E} &= \mathbf{W}^T A \mathbf{V}, & \hat{A} &= \mathbf{W}^T A \mathbf{V}, & \hat{H}_\xi &= \mathbf{W}^T H_\xi \mathbf{V}^{\textcircled{\xi}}, & \xi &\in \{2, \ldots, d\}, \\ \hat{B} &= \mathbf{W}^T B, & \hat{C} &= C\mathbf{V}, & \hat{N}_\eta &= \mathbf{W}^T H_\eta \mathbf{V}^{\textcircled{\eta}}, & \eta &\in \{1, \ldots, d\}. \end{aligned}$$

- Extended ideas from linear systems to polynomial systems.
- Interpolating points play an important role.

Goal [B./Goyal '19]

Construct projection matrices $\mathbf{V}$ and $\mathbf{W}$ such that

$$(\mathsf{GTF})^{\sigma}_{\text{original}} = (\mathsf{GTF})^{\sigma}_{\text{reduced}},$$

GTF = Generalized transfer functions

and reduced matrices are constructed via Petro-Galerkin projection:

$$\begin{aligned}
\hat{E} &= \mathbf{W}^T A\mathbf{V}, & \hat{A} &= \mathbf{W}^T A\mathbf{V}, & \hat{H}_\xi &= \mathbf{W}^T H_\xi \mathbf{V}^{\textcircled{\xi}}, & \xi &\in \{2,\ldots,d\},\\
\hat{B} &= \mathbf{W}^T B, & \hat{C} &= C\mathbf{V}, & \hat{N}_\eta &= \mathbf{W}^T H_\eta \mathbf{V}^{\textcircled{\eta}}, & \eta &\in \{1,\ldots,d\}.
\end{aligned}$$

- Extended ideas from linear systems to polynomial systems.
- Interpolating points play an important role.
- To make the **process fully automatic**, we propose a Loewner-type approach to construct good reduced-order systems.

Algorithm to Construct ROMs

Algorithm: Loewner-inspired method for determining reduced-order systems

1. Take $\sigma_i, \mu_i,\ i = 1, \dots, \mathcal{N}$.

Algorithm to Construct ROMs

Algorithm: Loewner-inspired method for determining reduced-order systems

1. Take $\sigma_i, \mu_i,\ i = 1, \ldots, \mathcal{N}$.
2. Compute $\mathcal{R} := \begin{cases} \operatorname{range}\left(\Phi(\sigma_1)B, \ldots, \Phi(\sigma_\mathcal{N})B\right) \\ \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)N_\eta\left(I_m \otimes \Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i\right)\right). \\ \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)H_\xi\left(\Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i\right)\right) \end{cases}$

Algorithm to Construct ROMs

Algorithm: Loewner-inspired method for determining reduced-order systems

1. Take $\sigma_i, \mu_i,\ i = 1, \ldots, \mathcal{N}$.
2. Compute $\mathcal{R} := \begin{cases} \operatorname{range}\left(\Phi(\sigma_1)B, \ldots, \Phi(\sigma_{\mathcal{N}})B\right) \\ \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)N_\eta \left(I_m \otimes \Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i\right)\right). \\ \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)H_\xi \left(\Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i\right)\right) \end{cases}$
3. Compute $\mathcal{O} = \begin{cases} \operatorname{range}\left(\Phi(\mu_1)^T C^T, \ldots, \Phi(\mu_{\mathcal{N}})^T C^T\right), \\ \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)\left(N_\eta\right)_{(2)} \left(I_m \otimes \Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\sigma_i)^T C^T\right)\right). \\ \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)\left(H_\xi\right)_{(2)} \left(\Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\mu_i)^T C^T\right)\right) \end{cases}$

Algorithm to Construct ROMs

Algorithm: Loewner-inspired method for determining reduced-order systems

1. Take σ_i, μ_i, $i = 1, \ldots, \mathcal{N}$.
2. Compute $\mathcal{R} := \begin{cases} \text{range}\left(\Phi(\sigma_1)B, \ldots, \Phi(\sigma_{\mathcal{N}})B\right) \\ \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{\mathcal{N}} \text{range}\left(\Phi(\sigma_i)N_\eta \left(I_m \otimes \Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i\right)\right). \\ \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{\mathcal{N}} \text{range}\left(\Phi(\sigma_i)H_\xi \left(\Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i\right)\right) \end{cases}$
3. Compute $\mathcal{O} = \begin{cases} \text{range}\left(\Phi(\mu_1)^T C^T, \ldots, \Phi(\mu_{\mathcal{N}})^T C^T\right), \\ \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{\mathcal{N}} \text{range}\left(\Phi(\sigma_i)\left(N_\eta\right)_{(2)} \left(I_m \otimes \Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\sigma_i)^T C^T\right)\right). \\ \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{\mathcal{N}} \text{range}\left(\Phi(\sigma_i)\left(H_\xi\right)_{(2)} \left(\Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\mu_i)^T C^T\right)\right) \end{cases}$
4. Determine Loewner and shifted-Loewner matrices: $\mathbb{L} = -\mathcal{O}^T E\mathcal{R}$, $\quad \mathbb{L}_s = -\mathcal{O}^T A\mathcal{R}$.

Algorithm to Construct ROMs

Algorithm: Loewner-inspired method for determining reduced-order systems

1. Take $\sigma_i, \mu_i,\ i = 1, \ldots, \mathcal{N}$.
2. Compute $\mathcal{R} := \begin{cases} \text{range}\left(\Phi(\sigma_1)B, \ldots, \Phi(\sigma_\mathcal{N})B\right) \\ \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{\mathcal{N}} \text{range}\left(\Phi(\sigma_i)N_\eta\left(I_m \otimes \Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i\right)\right). \\ \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{\mathcal{N}} \text{range}\left(\Phi(\sigma_i)H_\xi\left(\Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i\right)\right) \end{cases}$
3. Compute $\mathcal{O} = \begin{cases} \text{range}\left(\Phi(\mu_1)^T C^T, \ldots, \Phi(\mu_\mathcal{N})^T C^T\right), \\ \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{\mathcal{N}} \text{range}\left(\Phi(\sigma_i)\left(N_\eta\right)_{(2)}\left(I_m \otimes \Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\sigma_i)^T C^T\right)\right). \\ \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{\mathcal{N}} \text{range}\left(\Phi(\sigma_i)\left(H_\xi\right)_{(2)}\left(\Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\mu_i)^T C^T\right)\right) \end{cases}$
4. Determine Loewner and shifted-Loewner matrices: $\mathbb{L} = -\mathcal{O}^T E \mathcal{R}, \quad \mathbb{L}_s = -\mathcal{O}^T A \mathcal{R}$.
5. Compute singular value decomposition:

$$[\mathbf{Y}_1, \Sigma_1, \mathbf{X}_1] = \text{svd}\left([\mathbb{L}, \mathbb{L}_s]\right), \quad [\mathbf{Y}_2, \Sigma_2, \mathbf{X}_2] = \text{svd}\left(\begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix}\right).$$

Algorithm to Construct ROMs

Algorithm: Loewner-inspired method for determining reduced-order systems

1. Take $\sigma_i, \mu_i,\ i = 1, \ldots, \mathcal{N}$.
2. Compute $\mathcal{R} := \begin{cases} \operatorname{range}\left(\Phi(\sigma_1)B, \ldots, \Phi(\sigma_{\mathcal{N}})B\right) \\ \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)N_\eta \left(I_m \otimes \Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i\right)\right). \\ \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)H_\xi \left(\Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i\right)\right) \end{cases}$
3. Compute $\mathcal{O} = \begin{cases} \operatorname{range}\left(\Phi(\mu_1)^T C^T, \ldots, \Phi(\mu_{\mathcal{N}})^T C^T\right), \\ \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)\left(N_\eta\right)_{(2)} \left(I_m \otimes \Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\sigma_i)^T C^T\right)\right). \\ \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)\left(H_\xi\right)_{(2)} \left(\Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\mu_i)^T C^T\right)\right) \end{cases}$
4. Determine Loewner and shifted-Loewner matrices: $\mathbb{L} = -\mathcal{O}^T E \mathcal{R}, \quad \mathbb{L}_s = -\mathcal{O}^T A \mathcal{R}$.
5. Compute singular value decomposition:
$$[\mathbf{Y}_1, \Sigma_1, \mathbf{X}_1] = \operatorname{svd}\left([\mathbb{L}, \mathbb{L}_s]\right), \quad [\mathbf{Y}_2, \Sigma_2, \mathbf{X}_2] = \operatorname{svd}\left(\begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix}\right).$$
6. Determine projection matrices: $\mathbf{V} := \mathcal{R} X_2(:, 1:r), \quad \mathbf{W} := \mathcal{O} Y_2(:, 1:r)$.

Algorithm to Construct ROMs

Algorithm: Loewner-inspired method for determining reduced-order systems

Construct reduced-order system

$$\hat{E} = W^T AV, \ \ \hat{A} = W^T AV, \ \ \hat{H}_\xi = W^T H_\xi V^{\textcircled{\scriptsize ξ}},$$
$$\hat{B} = W^T B, \ \ \hat{C} = CV, \ \ \hat{N}_\eta = W^T N_\eta V^{\textcircled{\scriptsize η}}.$$

1. Take σ_i, μ_i, $i = 1, \ldots, \mathcal{N}$.
2. Compute $\mathcal{R} := \left\{ \begin{array}{l} \operatorname{range}\left(\Phi(\sigma_1)B, \ldots, \Phi(\sigma_\mathcal{N})B\right) \\ \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)N_\eta\left(I_m \otimes \Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i\right)\right). \\ \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)H_\xi\left(\Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i\right)\right) \end{array} \right.$
3. Compute $\mathcal{O} = \left\{ \begin{array}{l} \operatorname{range}\left(\Phi(\mu_1)^T C^T, \ldots, \Phi(\mu_\mathcal{N})^T C^T\right), \\ \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)\left(N_\eta\right)_{(2)}\left(I_m \otimes \Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\sigma_i)^T C^T\right)\right). \\ \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{\mathcal{N}} \operatorname{range}\left(\Phi(\sigma_i)\left(H_\xi\right)_{(2)}\left(\Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\mu_i)^T C^T\right)\right) \end{array} \right.$
4. Determine Loewner and shifted-Loewner matrices: $\mathbb{L} = -\mathcal{O}^T E\mathcal{R}$, $\quad \mathbb{L}_s = -\mathcal{O}^T A\mathcal{R}$.
5. Compute singular value decomposition:
$$[\mathbf{Y}_1, \Sigma_1, \mathbf{X}_1] = \operatorname{svd}\left([\mathbb{L}, \mathbb{L}_s]\right), \quad [\mathbf{Y}_2, \Sigma_2, \mathbf{X}_2] = \operatorname{svd}\left(\begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix}\right).$$
6. Determine projection matrices: $\mathbf{V} := \mathcal{R}X_2(:, 1:r)$, $\quad \mathbf{W} := \mathcal{O}Y_2(:, 1:r)$.

Numerical Example

Fitz-Hugh Nagumo model: Governing coupled equation

$$\begin{aligned}\epsilon v_t &= \epsilon^2 v_{xx} + v(v-0.1)(1-v) - w + q,\\ w_t &= hv - \gamma w + q\end{aligned}$$

with boundary condition

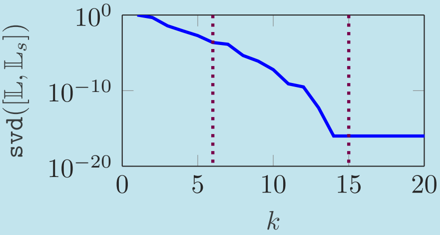
$$v(x,0)=0, \qquad w(x,0)=0, \qquad x\in(0,L), \qquad v_x(0,t)=i_0(t), \qquad v_x(1,t)=0, \qquad t\geq 0.$$

- To employ the interpolation-based algorithm, we choose random 100 interpolation points in a logarithmic way between $\left[10^{-2},10^{2}\right]$ and set $\sigma_i=\mu_i$, $i\in\{1,\ldots,100\}$.

Fitz-Hugh Nagumo model: Governing coupled equation

$$\begin{aligned} \epsilon v_t &= \epsilon^2 v_{xx} + v(v-0.1)(1-v) - w + q, \\ w_t &= hv - \gamma w + q \end{aligned}$$

Decay of singular values of $[\mathbb{L}, \mathbb{L}_s]$

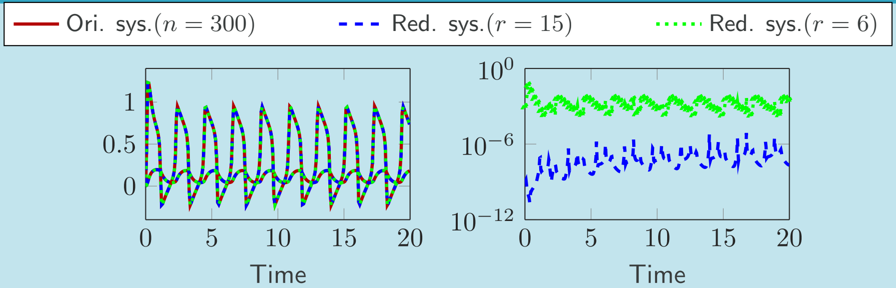


Numerical Example

Fitz-Hugh Nagumo model: Governing coupled equation

$$\begin{aligned} \epsilon v_t &= \epsilon^2 v_{xx} + v(v-0.1)(1-v) - w + q, \\ w_t &= hv - \gamma w + q \end{aligned}$$

Construction of reduced systems



Outlook

Contributions

- Extended two important system-theoretic MOR techniques, namely **balanced truncation** and **interpolation** of the transfer function.

Outlook

Contributions

- Extended two important system-theoretic MOR techniques, namely **balanced truncation** and **interpolation** of the transfer function.
- **Computational aspects** in a large-scale setting (low-rank factors, randomized SVDs, application of CUR).

Outlook

Contributions

- Extended two important system-theoretic MOR techniques, namely **balanced truncation** and **interpolation** of the transfer function.
- **Computational aspects** in a large-scale setting (low-rank factors, randomized SVDs, application of CUR).
- Proposed algorithms are **fully automatic** $\rightsquigarrow$ no expert knowledge required.

Outlook

Contributions

- Extended two important system-theoretic MOR techniques, namely **balanced truncation** and **interpolation** of the transfer function.
- **Computational aspects** in a large-scale setting (low-rank factors, randomized SVDs, application of CUR).
- Proposed algorithms are **fully automatic** $\rightsquigarrow$ no expert knowledge required.
- Extended results to parametric systems (not in the talk). [B./Goyal '19]

Outlook

Contributions

- Extended two important system-theoretic MOR techniques, namely **balanced truncation** and **interpolation** of the transfer function.
- **Computational aspects** in a large-scale setting (low-rank factors, randomized SVDs, application of CUR).
- Proposed algorithms are **fully automatic** $\rightsquigarrow$ no expert knowledge required.
- Extended results to parametric systems (not in the talk). [B./Goyal '19]

On-going work

- Possible extension to structured polynomial systems, e.g., second-order polynomial systems.

Outlook

Contributions

- Extended two important system-theoretic MOR techniques, namely **balanced truncation** and **interpolation** of the transfer function.
- **Computational aspects** in a large-scale setting (low-rank factors, randomized SVDs, application of CUR).
- Proposed algorithms are **fully automatic** $\rightsquigarrow$ no expert knowledge required.
- Extended results to parametric systems (not in the talk). [B./Goyal '19]

On-going work

- Possible extension to structured polynomial systems, e.g., second-order polynomial systems.
- Non-intrusive reduced-order modeling for nonlinear systems.

Outlook

Contributions

- Extended two important system-theoretic MOR techniques, namely **balanced truncation** and **interpolation** of the transfer function.
- **Computational aspects** in a large-scale setting (low-rank factors, randomized SVDs, application of CUR).
- Proposed algorithms are **fully automatic** $\rightsquigarrow$ no expert knowledge required.
- Extended results to parametric systems (not in the talk). [B./Goyal '19]

Thank you for your attention!!

On-going work

- Possible extension to structured polynomial systems, e.g., second-order polynomial systems.
- Non-intrusive reduced-order modeling for nonlinear systems.