System-Theoretic Model Reduction for Nonlinear (Parametric) Systems

Peter Benner

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

Minisymposium: Recent Advancements in Model Reduction for Stochastic and Nonlinear Systems

ICIAM, Valencia, Spain
June 19, 2019
1. Introduction

2. Nonlinear Systems

3. Balanced Truncation

4. Interpolation-Based Method

5. Numerical Example

6. Outlook
Introduction

–Reduced-order modeling motivation–

Physical systems

\[ \begin{align*}
\dot{v} + (v \cdot \nabla)v - \frac{1}{Re} \Delta v + \nabla p &= 0, \\
\nabla \cdot v &= 0.
\end{align*} \]

Partial differential equations

Circuits laws, mechanics, etc.

\[ \begin{align*}
E \dot{x}(t) &= f(x(t), u(t)), \\
y(t) &= g(x(t), u(t)).
\end{align*} \]

Differential (algebraic) equations

Using expert knowledge, experimental data

Discretize simulations, control, optimization, UQ, etc.

Large-scale system simulations, control, optimization, etc.

Goal

Reduced-order model

MOR
Introduction
–Reduced-order modeling motivation–

Physical systems
e.g., Navier-Stokes equations
\[ \dot{v} + (v \cdot \nabla)v - \frac{1}{\text{Re}} \Delta v + \nabla p = 0, \]
\[ \nabla \cdot v = 0. \]

Partial differential equations

Circuits laws, mechanics, etc.
Introduction

–Reduced-order modeling motivation–

Physical systems

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

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

Using expert knowledge, experimental data

Circuits laws, mechanics, etc.
Introduction
–Reduced-order modeling motivation–

Physical systems

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

Differential (algebraic) equations

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

Using expert knowledge, experimental data

Circuits laws, mechanics, etc.

Discretize

Partial differential equations

e.g., Navier-Stokes equations
Introduction

–Reduced-order modeling motivation–

Physical systems

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

Differential (algebraic) equations

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

Partial differential equations

Circuits laws, mechanics, etc.

Using expert knowledge, experimental data

Discretize

simulations, control, optimization, UQ, etc.

©Peter Benner, benner@mpi-magdeburg.mpg.de

System-Theoretic MOR for Nonlinear (Parametric) Systems
Introduction

–Reduced-order modeling motivation–

Physical systems

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

Partial differential equations

Circuits laws, mechanics, etc.

Using expert knowledge, experimental data

Differential (algebraic) equations

\[ Ex(t) = f(x(t), u(t)), \]
\[ y(t) = g(x(t), u(t)). \]

Large-scale system

Discretize

simulations, control, optimization, UQ, etc.

© Peter Benner, benner@mpi-magdeburg.mpg.de

System-Theoretic MOR for Nonlinear (Parametric) Systems

3/24
Introduction

–Reduced-order modeling motivation–

Physical systems

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

Using expert knowledge, experimental data

Differential (algebraic) equations

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

Partial differential equations

Circuits laws, mechanics, etc.

Large-scale system

Discretize

Simulations, control, optimization, UQ, etc.

©Peter Benner, benner@mpi-magdeburg.mpg.de

System-Theoretic MOR for Nonlinear (Parametric) Systems
Introduction

–Reduced-order modeling motivation–

Physical systems

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

Partial differential equations

Circuits laws, mechanics, etc.

Using expert knowledge, experimental data

Differential (algebraic) equations

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

Discretize

Large-scale system

Goal

Reduce numerical complexity

Simulations, control, optimization, UQ, etc.
Introduction
–Reduced-order modeling motivation–

Physical systems

\begin{align*}
\dot{v} + (v \cdot \nabla)v - \frac{1}{Re} \Delta v + \nabla p &= 0, \\
\nabla \cdot v &= 0.
\end{align*}

Differential (algebraic) equations

\begin{align*}
E \dot{x}(t) &= f(x(t), u(t)), \\
y(t) &= g(x(t), u(t)).
\end{align*}

Circuits laws, mechanics, etc.

Discretize

Using expert knowledge, experimental data

Large-scale system

Differential (algebraic) equations

Discretize

Partial differential equations

E.g., Navier-Stokes equations

Goal

Reduce numerical complexity

MOR

A solution

Reduced-order model

Simulations, control, optimization, UQ, etc.
Introduction
–Reduced-order modeling motivation–

**Using expert knowledge, experimental data**

**Physical systems**

**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

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

**Discretize**

**Large-scale system**

**Goal**

Reduce numerical complexity

**MOR**

Reduced-order model

Using expert knowledge, experimental data

Circuits laws, mechanics, etc.

Differential (algebraic) equations

Partial differential equations

Large-scale system

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

\[ \dot{x}(t) = Ax(t) + Bu(t) \]

\[ y(t) = Cx(t), \quad x(0) = 0. \]
Nonlinear Systems

Nonlinear systems

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

\[
\begin{align*}
    E\dot{x}(t) &= Ax(t) + Bu(t) + \sum_{\xi=2}^{d} H_{\xi} x^{(\xi)}(t) + \sum_{\eta=2}^{d} N_{\eta} (u(t) \otimes x^{(\eta)}(t)), \\
y(t) &= Cx(t), \quad x(0) = 0.
\end{align*}
\]
Nonlinear systems

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

\[
\begin{align*}
E \dot{x}(t) &= Ax(t) + Bu(t) + \sum_{\xi=2}^{d} H_\xi x^{\otimes}(t) + \sum_{\eta=2}^{d} N_\eta (u(t) \otimes x^{\otimes}(t)), \\
y(t) &= Cx(t), \quad x(0) = 0.
\end{align*}
\]

- \(d\) is the **degree of the polynomial term** in the system,
- (generalized) states \(x(t) \in \mathbb{R}^n\), \(x^{\otimes} := x(t) \otimes \cdots \otimes x(t)\),
- inputs (controls) \(u(t) \in \mathbb{R}^m\),
- outputs (measurements, quantities of interest) \(y(t) \in \mathbb{R}^q\).
In this talk, we consider nonlinear systems with polynomial terms:

\[ E \dot{x}(t) = Ax(t) + Bu(t) + \sum_{\xi=2}^{d} H_\xi x^{\otimes}(t) + \sum_{\eta=2}^{d} N_\eta (u(t) \otimes x^{\otimes}(t)), \]

\[ y(t) = C x(t), \quad x(0) = 0. \]

- \( d \) is the **degree of the polynomial term** in the system,
- (generalized) states \( x(t) \in \mathbb{R}^n, x^{\otimes} := x(t) \otimes \cdots \otimes x(t) \),
- 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

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

\[ E\dot{x}(t) = Ax(t) + Bu(t) + \sum_{\xi=2}^{d} H_\xi x^{\otimes}\xi(t) + \sum_{\eta=2}^{d} N_\eta (u(t) \otimes x^{\otimes}\eta(t)), \]

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

- \( d \) is the **degree of the polynomial term** in the system,
- (generalized) states \( x(t) \in \mathbb{R}^n, x^{\otimes} := x(t) \otimes \cdots \otimes x(t), \)
- 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 \( \leadsto \) **no approximation**. [McCormick '76, Gu '09]
Applications

Nonlinear systems

†† Courtesy of [HAWICK/PLAYNE ’10]
A nonlinear system

\[ \dot{x}_1(t) = -x_1(t) + x_3^3(t) + e^{-x_2(t)}, \]  \hspace{1em} (1a)

\[ \dot{x}_2(t) = -x_1(t) + u(t). \]  \hspace{1em} (1b)
A nonlinear system

\[
\begin{align*}
\dot{x}_1(t) &= -x_1(t) + x_2^3(t) + e^{-x_2(t)}, \\
\dot{x}_2(t) &= -x_1(t) + u(t).
\end{align*}
\]  

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

\[
\begin{align*}
\dot{x}_1(t) &= -x_1(t) + x_2^3(t) + e^{-x_2(t)}, \\
\dot{x}_2(t) &= -x_1(t) + u(t).
\end{align*}
\]

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

Then, we differentiate \( z_1(t) \):

\[
\dot{z}_1(t) = -e^{-x_2(t)} \cdot \dot{x}_2(t) = -z_1(t) (-x_1(t) + u(t)).
\]
A nonlinear system

\[
\begin{align*}
\dot{x}_1(t) &= -x_1(t) + x_2^3(t) + e^{-x_2(t)}, \\
\dot{x}_2(t) &= -x_1(t) + u(t).
\end{align*}
\]  

(1a)  

(1b)

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

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

An equivalent polynomial system

\[
\dot{x}_1(t) = -x_1(t) + x_2^3(t) + z_1(t),
\]
### A nonlinear system

\[
\begin{align*}
\dot{x}_1(t) &= -x_1(t) + x_2^3(t) + e^{-x_2(t)}, \\
\dot{x}_2(t) &= -x_1(t) + u(t).
\end{align*}
\]  

(1a)  

(1b)

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

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

### An equivalent polynomial system

\[
\begin{align*}
\dot{x}_1(t) &= -x_1(t) + x_2^3(t) + z_1(t), \\
\dot{x}_2(t) &= -x_1(t) + u(t),
\end{align*}
\]

© Peter Benner, benner@mpi-magdeburg.mpg.de
A nonlinear system

\[
\begin{align*}
\dot{x}_1(t) &= -x_1(t) + x_2^3(t) + e^{-x_2(t)}, \\
\dot{x}_2(t) &= -x_1(t) + u(t).
\end{align*}
\] (1a) (1b)

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

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

An equivalent polynomial system

\[
\begin{align*}
\dot{x}_1(t) &= -x_1(t) + x_2^3(t) + z_1(t), \\
\dot{x}_2(t) &= -x_1(t) + u(t), \\
\dot{z}_1(t) &= x_1(t)z_1(t) - z_1(t)u(t).
\end{align*}
\]
Construction of ROMs

Full-order system

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

\[
y(t) = Cx(t), \quad x(0) = 0,
\]
Construction of ROMs

**Full-order system**

\[
\begin{align*}
E\dot{x}(t) &= Ax(t) + \sum_{\xi=2}^{d} H_\xi x^{\otimes}(t)(t) + \sum_{\eta=1}^{d} N_\eta (u(t) \otimes x^{\otimes}(t)) + Bu(t), \\
y(t) &= Cx(t), \quad x(0) = 0,
\end{align*}
\]

**Petrov-Galerkin projection**

**Reduced-order system**

\[
\begin{align*}
\hat{E}\hat{x}(t) &= \hat{A}\hat{x}(t) + \sum_{\xi=2}^{d} \hat{H}_\xi \hat{x}^{\otimes}(t)(t) + \sum_{\eta=1}^{d} \hat{N}_\eta (u(t) \otimes \hat{x}^{\otimes}(t)) + \hat{B}u(t), \\
\hat{y}(t) &= \hat{C}\hat{x}(t), \quad \hat{x}(0) = 0,
\end{align*}
\]

\[
\begin{align*}
\hat{E} &= W^T EV, \quad \hat{A} = W^T AV, \quad \hat{H}_\xi = W^T H_\xi V^{\otimes}, \quad \xi \in \{2, \ldots, d\}, \\
\hat{B} &= W^T B, \quad \hat{C} = CV, \quad \hat{N}_\eta = W^T N_\eta V^{\otimes}, \quad \eta \in \{1, \ldots, d\}.
\end{align*}
\]
Existing Approaches

Snapshot-based methods

- Proper orthogonal decomposition,
- Reduced basis methods,
- Non-intrusive reduced-order modeling.

e.g., [Volkwein ’08]
e.g., [Quarteroni et al. ’16]
[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,
- Reduced basis methods,
- Non-intrusive reduced-order modeling.

*E.g., [Volkwein '08]*
*E.g., [Quarteroni et al. '16]*
*[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
  - Interpolation-based methods

*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,
- Reduced basis methods,
- Non-intrusive reduced-order modeling.

**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 \textit{quadratic-bilinear} systems)
  - Balanced truncation
  - Interpolation-based methods

**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 ≥ 3** polynomial systems, we present extensions of
  - Balanced truncation
  - Interpolation-based methods
Balanced Truncation
–Main Concept–

- 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.

\[
\begin{align*}
\mathbf{x}_n & \quad \text{Initial state} \\
\mathbf{x}_{n-1} & \quad \text{Final state} \\
\mathbf{x}_{n-2} & \\
\vdots & \\
\mathbf{x}_1 & \\
\mathbf{x}_0 & \quad \text{Initial state}
\end{align*}
\]

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.

Idea of balanced truncation

© Peter Benner, benner@mpi-magdeburg.mpg.de
Balanced Truncation
–Main Concept–

- Relies on the concept of reachability and observability.

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

![Diagram showing reachability and observability in a state space.](image)
Balanced Truncation

Main Concept

- 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.

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

－Main Concept－

- 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**.

![Diagram of state transformation](image)
Balanced Truncation

- Main Concept -

- 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

\[ \Sigma : \begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t), \\
y(t) &= Cx(t), \quad x(0) = 0.
\end{align*} \]

- 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:
\[ Q := \int_0^{+\infty} \left( Ce^{At} \right)^T C e^{At} dt. \]

Gramians

The controllability and observability Gramians satisfy

\[ AP + PA^T + BB^T = 0, \quad A^T Q + QA + C^T C = 0. \]
Energy functionals for linear systems

For linear systems, the energy functionals are given by

\[ L_c(x_0) = x_0^T P^{-1} x_0, \quad L_o(x_0) = x_0^T Q x_0. \]
Energy functionals for linear systems

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

\[ L_c(x_0) = x_0^T P^{-1} x_0, \quad L_o(x_0) = x_0^T Q x_0. \]

- 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

\[ L_c(x_0) = x_0^T P^{-1} x_0, \quad L_o(x_0) = x_0^T Q x_0. \]

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

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

\[ L_c(x_0) = x_0^T P^{-1} x_0, \quad L_o(x_0) = x_0^T Q x_0. \]

- The Hankel singular values are given by \( \{ \sigma_i | i = \{1, \ldots, n\} \} = \sqrt{\Lambda(PQ)}. \)
- Find state transformation such that \( P = Q = \text{diag}(\sigma_1, \ldots, \sigma_n). \)
- 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.
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,
Generally, exact energy functionals are given by the solutions of **nonlinear Hamilton-Jacobi equations** and **nonlinear Lyapunov**-type equations.

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. 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 polynomial control (PC) systems, and
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 polynomial control (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** ($P$) of a polynomial system solves the **polynomial Lyapunov** equation

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

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

$$A^T Q + QA + C^T C + \sum_{\xi=1}^{d-1} H^{(2)}_{\xi+1} (P \otimes Q) \left( H^{(2)}_{\xi+1}\right)^T + \sum_{\eta=0}^{d-1} N_{\eta+1} (P \otimes Q) (N_{\eta+1})^T = 0.$$
We show **bounds** for the **energy functionals** (at least in the neighborhood of the origin), similar to the bilinear and quadratic-bilinear case, as:

\[
L_c(x_0) \geq \frac{1}{2} x_0^T P^{-1} x_0, \quad L_o(x_0) \leq \frac{1}{2} x_0^T Q x_0.
\]
We show **bounds** for the **energy functionals** (at least in the neighborhood of the origin), similar to the bilinear and quadratic-bilinear case, as:

\[
L_c(x_0) \geq \frac{1}{2} x_0^T P^{-1} x_0, \quad L_o(x_0) \leq \frac{1}{2} x_0^T Q x_0.
\]

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

1. Assuming zero initial condition, \(x(t, 0, u) \in \text{Im} P, \forall \ t \geq 0\) and all input functions.
   \[\Rightarrow \text{If the final state } \not\in \text{Im} P, \text{ it is unreachable.}\]
We show **bounds** for the **energy functionals** (at least in the neighborhood of the origin), similar to the bilinear and quadratic-bilinear case, as:

\[
L_c(x_0) \geq \frac{1}{2} x_0^T P^{-1} x_0, \quad L_o(x_0) \leq \frac{1}{2} x_0^T Q x_0.
\]

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

1. Assuming zero initial condition, \(x(t,0,u) \in \text{Im}P, \forall \ t \geq 0\) and all input functions.
   \(\Rightarrow\) If the final state \(\not\in \text{Im}P\), it is unreachable.

2. If \(P > 0\) and the initial state \(\in \text{Ker}Q\), then it is unobservable.
Polynomial Lyapunov equations are very expensive to solve.

Definition

The truncated reachability Gramian \((P_T)\) of a polynomial system solves the linear Lyapunov equation

\[
AP_T + P_TA_T + BB_T + d\sum_{\xi=2}^{\infty} H(\xi) P = 0.
\]

The truncated observability Gramian \((Q_T)\) of a polynomial system solves the linear Lyapunov equation

\[
A_T Q_T + Q_TA_T + C_T C + d_{-1}\sum_{\xi=1}^{\infty} H(2\xi) + 1 (P\xi\odot l)(H(2\xi) + 1)^T + d_{-1}\sum_{\eta=0}^{\infty} N(2\xi + 1) (P\eta\odot l)(N(2\xi + 1))^T = 0,
\]

where

\[
AP_l + P_l A_T = 0 \quad \text{and} \quad A_T Q_l + Q_l A + C_T C = 0.
\]

Advantage:

Only need to solve four (linear) Lyapunov equations.
Polynomial Lyapunov equations are very expensive to solve.

We propose truncated Gramians that only involve a finite number of kernels.

**Definition**

The truncated **reachability Gramian** $(P_T)$ of a polynomial system solves the **linear Lyapunov** equation

$$AP_T + P_T A^T + BB^T + \sum_{\xi=2}^{d} H^{(2)}_{\xi} P_l^{\otimes} H^{(2)}_{\xi} + \sum_{\eta=1}^{d} N_{\eta} P_l^{\odot} (N_{\eta})^T = 0.$$  

The truncated **observability Gramian** $(Q_T)$ of a polynomial system solves the **linear Lyapunov** equation

$$A^T Q_T + Q_T A + C^T C + \sum_{\xi=1}^{d-1} H^{(2)}_{\xi+1} (P_l^{\otimes} \otimes Q_l) \left(H^{(2)}_{\xi+1}\right)^T + \sum_{\eta=0}^{d-1} N_{\eta+1}^{(2)} (P_l^{\odot} \otimes Q_l) \left(N_{\eta+1}^{(2)}\right)^T = 0,$$

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].

Algorithm: balanced truncation for polynomial systems

Provide system matrices \( A, H \), \( \xi, N_k \), \( \eta \), \( B, C \), and order of the reduced system \( r \) (optional).

Step 1: Compute system Gramians:

\[
AP^T + P^T A^T + BB^T + \sum_{\xi=2}^{\infty} H\xi P\xi \odot l H^T \xi + \sum_{k=1}^{m} \sum_{\eta=1}^{\infty} N_k \eta P\eta \odot l (N_k \eta) T = 0.
\]

Low-rank factors:

\[
P^T \approx SS^T \text{ and } Q^T \approx RR^T.
\]

\[
A P_l + P_l A^T + B B^T = 0,
\]

\[
A T Q_l + Q_l A + C T C = 0.
\]

\[
P_l \approx , \quad Q_l \approx 
\]

Step 2: Determine projection matrices:

Step 3: Compute the reduced-order system:

\[
\hat{A} = W^T A V,
\]

\[
\hat{H}_\xi = W^T H_\xi V \xi \odot l ,
\]

\[
\hat{N}_k \eta = W^T N_k \eta V \eta \odot l ,
\]

\[
\hat{B} = W^T B,
\]

\[
\hat{C} = C V.
\]
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 $A, H_\xi, N^k_\eta, B, C$, and order of the reduced system $r$ (optional).
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 \( A, H_\xi, N^k_\eta, B, C \), and order of the reduced system \( r \) (optional).

**Step 1: Compute system Gramians:**

\[
AP + P A^T + B B^T + \sum_{\xi=2}^{d} H_\xi P^{\otimes} H_\xi^T + \sum_{\eta=1}^{d} \sum_{k=1}^{m} N^k_\eta P^{\otimes} (N^k_\eta)^T = 0.
\]

**Low-rank factors:** \( P \approx S S^T \) and \( Q \approx R R^T \).
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 $A, H_\xi, N^k_\eta, B, C$, and order of the reduced system $r$ (optional).

**Step 1: Compute truncated system Gramians:**

$$AP_T + P_T A^T + BB^T + \sum_{\xi=2}^d H_\xi P_l \otimes H_\xi^T + \sum_{\eta=1}^d \sum_{k=1}^m N^k_\eta P_l \otimes \left(N^k_\eta\right)^T = 0.$$  

Low-rank factors: $P_T \approx SS^T$ and $Q_T \approx RR^T$.

$$AP_l + P_l A^T + BB^T = 0,$$

$$A^T Q_l + Q_l A + C^T C = 0.$$  

$$P_l \approx I, \quad Q_l \approx I.$$
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 $A, H_\xi, N^k_\eta, B, C$, and order of the reduced system $r$ (optional).

**Step 1: Compute truncated system Gramians:**

**Step 2: Determine projection matrices:**
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 $A, H_\xi, N^k_\eta, B, C$, and order of the reduced system $r$ (optional).

**Step 1: Compute truncated system Gramians:**

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

$$V = SU_1\Sigma_1^{-\frac{1}{2}}, \quad W = RV_1\Sigma_1^{-\frac{1}{2}}.$$
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 $A, H_\xi, N^k_\eta, B, C$, 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:

\[
\begin{align*}
\hat{A} &= W^T A V, \\
\hat{H}_\xi &= W^T H_\xi V^\xi, \\
\hat{N}^k_\eta &= W^T N^k_\eta V^@, \\
\hat{B} &= W^T B, \\
\hat{C} &= C V.
\end{align*}
\]
Balanced Truncation
–FitzHugh-Nagumo Model–

Governed equations

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

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

After discretization, we obtain a polynomial control (PC) system with cubic nonlinearity of order \( n_{pc} = 600. \)

[B./Breiten ’15]
Balanced Truncation
–FitzHugh-Nagumo Model–

Governing equations

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

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

- After discretization, we obtain a polynomial control (PC) system with cubic nonlinearity of order \( n_{pc} = 600 \).

  - The transformed quadratic-bilinear (QB) system is of order \( n_{qb} = 900 \).
Balanced Truncation
–FitzHugh-Nagumo Model–

**Governing equations**

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

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

- After discretization, we obtain a polynomial control (PC) system 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 \(x = 0\).
Balanced Truncation
–FitzHugh-Nagumo Model–

\[ \varepsilon v_t(x, t) = \varepsilon^2 v_{xx}(x, t) + v(1 - v)(v - 0.1) - w(x, t) + q, \]
\[ w_t(x, t) = h v(x, t) - \gamma w(x, t) + q, \]
\[ v(x, 0) = 0, \quad w(x, 0) = 0, \quad x \in [0, L], \]
\[ v_x(0, t) = u(t), \quad v_x(L, t) = 0, \quad t \geq 0, \]

where \( \varepsilon = 0.015, \ h = 0.5, \ \gamma = 2, \ q = 0.05, \ L = 0.3. \)

- After discretization, we obtain a polynomial control (PC) system with cubic nonlinearity of order \( n_{pc} = 600. \)

- 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.
Balanced Truncation
–Singular values decay–

BT for QB systems

BT for PC systems

Decay of singular values for PC systems is faster \( \Rightarrow \) smaller reduced order model!
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.
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 30.
- Original PC system of order 600. Original QB of order 900.
- Reduced PC system of order 10. Reduced QB system of order 43.
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

\[ B./Goyal '19 \]

\[
F_L(s_1) := C \Phi(s_1) B,
\]

\[
F(\xi) := H(s_1, \ldots, s_\xi+1) := C \Phi(s_\xi+1) H_\xi(\Phi(s_\xi) B \otimes \cdots \otimes \Phi(s_1) B),
\]

\[
F(\eta) := N(s_1, \ldots, s_\eta+1) := C \Phi(s_\eta+1) N_\eta(I_m \otimes \Phi(s_\eta) B \otimes \cdots \otimes \Phi(s_1) B),
\]

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

\[
\dot{E}x(t) = Ax(t) + d \sum_{\xi=2} H_\xi x_\xi \circ(t) + d \sum_{\eta=2} N_\eta(u(t) \otimes x_\eta \circ(t))(t) + Bu(t),
\]

\[ y(t) = Cx(t), \quad x(0). \]
As for linear systems, we can define the input-output mapping by generalized transfer functions.
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.
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**

\[
F_L(s) := C \Phi(s) B, \\
F_H^{(\xi)}(s_1, \ldots, s_{\xi+1}) := C \Phi(s_{\xi+1}) H_\xi (\Phi(s_{\xi}) B \otimes \cdots \otimes \Phi(s_1 B)), \\
F_N^{(\eta)}(s_1, \ldots, s_{\eta+1}) := C \Phi(s_{\eta+1}) N_\eta (I_m \otimes \Phi(s_{\eta}) B \otimes \cdots \otimes \Phi(s_1 B)),
\]

where \( \Phi(s) := (sE - A)^{-1} \).
Construct projection matrices $V$ and $W$ such that

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

and reduced matrices are constructed via Petro-Galerkin projection:

$\hat{E} = W^T AV,$  $\hat{A} = W^T AV,$  $\hat{H}_\xi = W^T H_\xi V^\xi$,  $\xi \in \{2, \ldots, d\},$

$\hat{B} = W^T B,$  $\hat{C} = CV,$  $\hat{N}_\eta = W^T H_\eta V^\eta$,  $\eta \in \{1, \ldots, d\}.$
Construct projection matrices \( \mathbf{V} \) and \( \mathbf{W} \) such that

\[
(GTF)^\sigma_{\text{original}} = (GTF)^\sigma_{\text{reduced}},
\]

and reduced matrices are constructed via Petro-Galerkin projection:

\[
\begin{align*}
\hat{\mathbf{E}} &= \mathbf{W}^T \mathbf{A} \mathbf{V}, & \hat{\mathbf{A}} &= \mathbf{W}^T \mathbf{A} \mathbf{V}, & \hat{\mathbf{H}}_{\xi} &= \mathbf{W}^T \mathbf{H}_\xi \mathbf{V}^\otimes, & \xi \in \{2, \ldots, d\}, \\
\hat{\mathbf{B}} &= \mathbf{W}^T \mathbf{B}, & \hat{\mathbf{C}} &= \mathbf{C} \mathbf{V}, & \hat{\mathbf{N}}_{\eta} &= \mathbf{W}^T \mathbf{H}_\eta \mathbf{V}^\otimes, & \eta \in \{1, \ldots, d\}.
\end{align*}
\]

Extended ideas from linear systems to polynomial systems.
Goal

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

$$(\text{GTF})^\sigma_{\text{original}} = (\text{GTF})^\sigma_{\text{reduced}},$$

and reduced matrices are constructed via Petro-Galerkin projection:

$$\hat{E} = \mathbf{W}^T \mathbf{A} \mathbf{V}, \quad \hat{A} = \mathbf{W}^T \mathbf{A} \mathbf{V}, \quad \hat{H}_\xi = \mathbf{W}^T H_\xi \mathbf{V}^\otimes, \quad \xi \in \{2, \ldots, d\},$$

$$\hat{B} = \mathbf{W}^T \mathbf{B}, \quad \hat{C} = \mathbf{C} \mathbf{V}, \quad \hat{N}_\eta = \mathbf{W}^T H_\eta \mathbf{V}^\otimes, \quad \eta \in \{1, \ldots, d\}.$$

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

Goal

Construct projection matrices $V$ and $W$ such that

\[(GTF)^\sigma_{\text{original}} = (GTF)^\sigma_{\text{reduced}},\]

and reduced matrices are constructed via Petro-Galerkin projection:

\[
\begin{align*}
\hat{E} &= W^TAV, \\
\hat{A} &= W^TAV, \\
\hat{H}_\xi &= W^T H_\xi V^{\circ}, & \xi \in \{2, \ldots, d\}, \\
\hat{B} &= W^TB, \\
\hat{C} &= CV, \\
\hat{N}_\eta &= W^T H_\eta V^{\circ}, & \eta \in \{1, \ldots, d\}.
\end{align*}
\]

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

1. Take $\sigma_i, \mu_i, i = 1, \ldots, N$. 

<table>
<thead>
<tr>
<th>Line</th>
<th>Math</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>$R = \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{N} \text{range} (\Phi(\sigma_i) B \cdots \Phi(\sigma_i) B \cdots \Phi(\sigma_i) B) \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{N} \text{range} (\Phi(\sigma_i) H^{(2)}(\Phi(\sigma_i) B \cdots \Phi(\sigma_i) B \cdots \Phi(\sigma_i) B))$.</td>
</tr>
<tr>
<td>3.</td>
<td>$O = \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{N} \text{range} (\Phi(\mu_i)^T C^T, \ldots, \Phi(\mu_i)^T C^T)$, $\bigcup_{\xi=2}^{d} \bigcup_{i=1}^{N} \text{range} (\Phi(\sigma_i) H^{(2)}(\Phi(\sigma_i) B \cdots \Phi(\sigma_i) B \cdots \Phi(\sigma_i) B))$.</td>
</tr>
<tr>
<td>5.</td>
<td>Compute singular value decomposition: $[Y_1, \Sigma_1, X_1] = \text{svd}([L, L_s])$, $[Y_2, \Sigma_2, X_2] = \text{svd}([L, L_s])$.</td>
</tr>
<tr>
<td>6.</td>
<td>Determine projection matrices: $V = RX_2(:,1:r)$, $W = OY_2(:,1:r)$. $\hat{E} = W^T AV$, $\hat{A} = W^T AV$, $\hat{H}^{(\xi)} = W^T H^{(\xi)} V^{(\xi)} \circ$, $\hat{B} = W^T B$, $\hat{C} = CV$, $\hat{N}^{(\eta)} = W^T N^{(\eta)} V^{(\eta)} \circ$.</td>
</tr>
</tbody>
</table>
Algorithm: Loewner-inspired method for determining reduced-order systems

1. Take $\sigma_i, \mu_i, i = 1, \ldots, N$.

2. Compute $\mathcal{R} := \begin{cases} \text{range } (\Phi(\sigma_1)B, \ldots, \Phi(\sigma_N)B) \\ \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{N} \text{range } (\Phi(\sigma_i)N_{\eta} (I_m \otimes \Phi(\sigma_i)B_{i} \otimes \cdots \otimes \Phi(\sigma_i)B_{i})) \\ \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{N} \text{range } (\Phi(\sigma_i)H_{\xi} (\Phi(\sigma_i)B_{i} \otimes \cdots \otimes \Phi(\sigma_i)B_{i})) \end{cases}$

3. Compute $\mathcal{O} := \begin{cases} \text{range } (\Phi(\mu_1)^T C_{1}^T, \ldots, \Phi(\mu_N)^T C_{1}^T) \\ \bigcup_{\eta=1}^{d} \bigcup_{i=1}^{N} \text{range } (\Phi(\sigma_i)(N_{\eta} (I_m \otimes \Phi(\sigma_i)B_{i} \otimes \cdots \otimes \Phi(\sigma_i)B_{i}))) \\ \bigcup_{\xi=2}^{d} \bigcup_{i=1}^{N} \text{range } (\Phi(\sigma_i)(H_{\xi} (\Phi(\sigma_i)B_{i} \otimes \cdots \otimes \Phi(\sigma_i)B_{i}))) \end{cases}$

4. Determine Loewner and shifted-Loewner matrices:
   
   \[ L = -\mathcal{O}^T E \mathcal{R}, \quad L_s = -\mathcal{O}^T A \mathcal{R}. \]

5. Compute singular value decomposition:
   
   \[ [Y_1, \Sigma_1, X_1] = \text{svd}(L, L_s), \quad [Y_2, \Sigma_2, X_2] = \text{svd}(L, L_s). \]

6. Determine projection matrices:
   
   \[ V := \mathcal{R} X_2(:, 1:r), \quad W := \mathcal{O} Y_2(:, 1:r). \]

   \[ \hat{E} = W^T A V, \quad \hat{A} = W^T C V, \quad \hat{H}_{\xi} = W^T H_{\xi} V_{\xi} \circ, \quad \hat{B} = W^T B, \quad \hat{C} = CV, \quad \hat{N}_{\eta} = W^T N_{\eta} V_{\eta} \circ. \]
Algorithm: Loewner-inspired method for determining reduced-order systems

1. Take $\sigma_i, \mu_i$, $i = 1, \ldots, N$.

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

3. Compute $\mathcal{O} := \left\{ \begin{array}{l}
\text{range } (\Phi(\mu_1)^T C^T, \ldots, \Phi(\mu_N)^T C^T), \\
\bigcup_{\eta=1}^{d} \bigcup_{i=1}^{N} \text{range } \left( \Phi(\sigma_i) (N_\eta)_2 \left( I_m \otimes \Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\sigma_i)^T C^T \right) \right) \\
\bigcup_{\xi=2}^{d} \bigcup_{i=1}^{N} \text{range } \left( \Phi(\sigma_i) (H_\xi)_2 \left( \Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\mu_i)^T C^T \right) \right)
\end{array} \right.$

4. Determine Loewner and shifted-Loewner matrices:

   $L = -\mathcal{O}^T \mathcal{R}$,
   $L_s = -\mathcal{O}^T \mathcal{A} \mathcal{R}$.

5. Compute singular value decomposition:

   $[\mathcal{Y}_1, \Sigma_1, \mathcal{X}_1] = \text{svd}( [L, L_s] )$,
   $[\mathcal{Y}_2, \Sigma_2, \mathcal{X}_2] = \text{svd}( [L_L, L_s] )$.

6. Determine projection matrices:

   $\mathcal{V} := \mathcal{R} \mathcal{X}_2(:,1:r)$,
   $\mathcal{W} := \mathcal{O} \mathcal{Y}_2(:,1:r)$.

   $\hat{E} = \mathcal{W}^T \mathcal{A} \mathcal{V}$,
   $\hat{A} = \mathcal{W}^T \mathcal{A} \mathcal{V}$,
   $\hat{H}_\xi = \mathcal{W}^T \mathcal{H}_\xi \mathcal{V} \circ \mathcal{H}_\xi$,
   $\hat{B} = \mathcal{W}^T \mathcal{B}$,
   $\hat{C} = \mathcal{C} \mathcal{V}$,
   $\hat{N}_\eta = \mathcal{W}^T \mathcal{N}_\eta \mathcal{V} \circ \mathcal{N}_\eta$.
Algorithm to Construct ROMs

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

1. Take $\sigma_i, \mu_i$, $i = 1, \ldots, N$.

2. Compute $\mathcal{R} := \left\{ \begin{array}{l}
\text{range } (\Phi(\sigma_1)B, \ldots, \Phi(\sigma_N)B) \\
\bigcup_{\eta=1}^{d} \bigcup_{i=1}^{N} \text{range } (\Phi(\sigma_i)N_\eta (I_m \otimes \Phi(\sigma_i)B_b \otimes \cdots \otimes \Phi(\sigma_i)B_b)) \\
\bigcup_{\xi=2}^{d} \bigcup_{i=1}^{N} \text{range } (\Phi(\sigma_i)H_\xi (\Phi(\sigma_i)B_b \otimes \cdots \otimes \Phi(\sigma_i)B_b))
\end{array} \right\}$.

3. Compute $\mathcal{O} = \left\{ \begin{array}{l}
\bigcup_{\eta=1}^{d} \bigcup_{i=1}^{N} \text{range } \left( \Phi(\sigma_i) (N_\eta)_{(2)} (I_m \otimes \Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\sigma_i)^T C^T) \right) \\
\bigcup_{\xi=2}^{d} \bigcup_{i=1}^{N} \text{range } \left( \Phi(\sigma_i) (H_\xi)_{(2)} (\Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\mu_i)^T C^T) \right)
\end{array} \right\}$.

4. Determine Loewner and shifted-Loewner matrices: $L = -\mathcal{O}^T E \mathcal{R}$, $L_s = -\mathcal{O}^T A \mathcal{R}$. 
Algorithm: Loewner-inspired method for determining reduced-order systems

1. Take $\sigma_i, \mu_i, i = 1, \ldots, N$.

2. Compute $\mathcal{R} := \left\{ \right.$

   \begin{align*}
   &\text{range } (\Phi(\sigma_1)B, \ldots, \Phi(\sigma_N)B) \\
   &\cup_{\eta=1}^d \cup_{i=1}^N \text{range } (\Phi(\sigma_i)N_\eta (I_m \otimes \Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i)) \\
   &\cup_{\xi=2}^d \cup_{i=1}^N \text{range } (\Phi(\sigma_i)H_\xi (\Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i))
   \end{align*}

3. Compute $\mathcal{O} := \left\{ \right.$

   \begin{align*}
   &\text{range } (\Phi(\mu_1)^T C^T, \ldots, \Phi(\mu_N)^T C^T) \\
   &\cup_{\eta=1}^d \cup_{i=1}^N \text{range } (\Phi(\sigma_i)(N_\eta)(2) (I_m \otimes \Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\sigma_i)^T C^T)) \\
   &\cup_{\xi=2}^d \cup_{i=1}^N \text{range } (\Phi(\sigma_i)(H_\xi)(2) (\Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\mu_i)^T C^T))
   \end{align*}

4. Determine Loewner and shifted-Loewner matrices: $\mathbb{L} = -\mathcal{O}^T \mathcal{E} \mathcal{R}, \quad \mathbb{L}_s = -\mathcal{O}^T \mathcal{A} \mathcal{R}$.

5. Compute singular value decomposition:

   \begin{align*}
   \begin{bmatrix} Y_1, \Sigma_1, X_1 \end{bmatrix} &= \text{svd } (\begin{bmatrix} \mathbb{L}, \mathbb{L}_s \end{bmatrix}) , \quad \begin{bmatrix} Y_2, \Sigma_2, X_2 \end{bmatrix} = \text{svd } \left( \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} \right).
   \end{align*}
Algorithm to Construct ROMs

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

1. Take $\sigma_i, \mu_i, i = 1, \ldots, N$.

2. Compute $\mathcal{R} := \left\{ \begin{array}{l} \text{range} \left( \Phi(\sigma_1)B, \ldots, \Phi(\sigma_N)B \right) \\ \cup_{\eta=1}^{d} \cup_{i=1}^{N} \text{range} \left( \Phi(\sigma_i)N_\eta (I_m \otimes \Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i) \right) \\ \cup_{\xi=2}^{d} \cup_{i=1}^{N} \text{range} \left( \Phi(\sigma_i)H_\xi (\Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i) \right) \end{array} \right.$

3. Compute $\mathcal{O} = \left\{ \begin{array}{l} \text{range} \left( \Phi(\mu_1)^T C^T, \ldots, \Phi(\mu_N)^T C^T \right), \\ \cup_{\eta=1}^{d} \cup_{i=1}^{N} \text{range} \left( \Phi(\sigma_i)(N_\eta)_{(2)} (I_m \otimes \Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\sigma_i)^T C^T) \right) \\ \cup_{\xi=2}^{d} \cup_{i=1}^{N} \text{range} \left( \Phi(\sigma_i)(H_\xi)_{(2)} (\Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\mu_i)^T C^T) \right) \end{array} \right.$

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:

\[
\begin{bmatrix} Y_1, \Sigma_1, X_1 \end{bmatrix} = \text{svd} \left( \begin{bmatrix} \mathbb{L}, \mathbb{L}_s \end{bmatrix} \right), \quad \begin{bmatrix} Y_2, \Sigma_2, X_2 \end{bmatrix} = \text{svd} \left( \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} \right).
\]

6. Determine projection matrices: $\mathbf{V} := \mathcal{R}X_2(:, 1 : r), \quad \mathbf{W} := \mathcal{O}Y_2(:, 1 : r)$. 

©Peter Benner, benner@mpi-magdeburg.mpg.de 

System-Theoretic MOR for Nonlinear (Parametric) Systems
Algorithm: Loewner-inspired method for determining reduced-order systems

1. Take $\sigma_i, \mu_i, i = 1, \ldots, N$.

2. Compute $R := \left\{ \right.$
   \begin{align*}
   &\text{range } (\Phi(\sigma_1)B, \ldots, \Phi(\sigma_N)B) \\
   &\bigcup_{\eta=1}^d \bigcup_{i=1}^N \text{range } (\Phi(\sigma_i)N_\eta (I_m \otimes \Phi(\sigma_i)B_{\eta}) \otimes \cdots \otimes \Phi(\sigma_i)B_{\eta})) \\
   &\bigcup_{\xi=2}^d \bigcup_{i=1}^N \text{range } (\Phi(\sigma_i)H_\xi(\Phi(\sigma_i)B_{\eta} \otimes \cdots \otimes \Phi(\sigma_i)B_{\eta}))
   \left. \right\}$

3. Compute $O := \left\{ \right.$
   \begin{align*}
   &\text{range } (\Phi(\mu_1)^T C^T, \ldots, \Phi(\mu_N)^T C^T), \\
   &\bigcup_{\eta=1}^d \bigcup_{i=1}^N \text{range } \left( \Phi(\sigma_i)(N_\eta)_{(2)}(I_m \otimes \Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\sigma_i)^T C^T) \right) \\
   &\bigcup_{\xi=2}^d \bigcup_{i=1}^N \text{range } \left( \Phi(\sigma_i)(H_\xi)_{(2)}(\Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\mu_i)^T C^T) \right)
   \left. \right\}$

4. Determine Loewner and shifted-Loewner matrices:
   \[ L = -O^T ER, \quad L_s = -O^T A R. \]

5. Compute singular value decomposition:
   \[ [Y_1, \Sigma_1, X_1] = \text{svd } \left[ [L, L_s] \right], \quad [Y_2, \Sigma_2, X_2] = \text{svd } \left[ \frac{L}{L_s} \right]. \]

6. Determine projection matrices:
   \[ V := RX_2(:, 1 : r), \quad W := OY_2(:, 1 : r). \]
Fitz-Hugh Nagumo model: Governing coupled equation

\[ \epsilon v_t = \epsilon^2 v_{xx} + v(v - 0.1)(1 - v) - w + q, \]
\[ w_t = hv - \gamma w + q \]

with boundary condition

\[ v(x, 0) = 0, \quad w(x, 0) = 0, \quad x \in (0, L), \quad v_x(0, t) = i_0(t), \quad v_x(1, t) = 0, \quad t \geq 0. \]

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

Fitz-Hugh Nagumo model: Governing coupled equation

\[ \epsilon v_t = \epsilon^2 v_{xx} + v(v - 0.1)(1 - v) - w + q, \]
\[ w_t = hv - \gamma w + q \]

Decay of singular values of \([L, L_s]\)
Fitz-Hugh Nagumo model: Governing coupled equation

\[ \epsilon v_t = \epsilon^2 v_{xx} + v(v - 0.1)(1 - v) - w + q, \]
\[ w_t = hv - \gamma w + q \]

Construction of reduced systems

- Ori. sys. \((n = 300)\)
- Red. sys. \((r = 15)\)
- Red. sys. \((r = 6)\)
### 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).

Proposed algorithms are fully automatic → no expert knowledge required.

Extended results to parametric systems (not in the talk).

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

Non-intrusive reduced-order modeling for nonlinear systems.

Thank you for your attention!!
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 \(\Rightarrow\) 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** ⇝ no expert knowledge required.

- Extended results to parametric systems (not in the talk).  

[B./Goyal ’19]
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** ⇔ no expert knowledge required.
- Extended results to parametric systems (not in the talk).

On-going work

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

[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** → 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 ⇝ no expert knowledge required.
- Extended results to parametric systems (not in the talk).

On-going work

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

Thank you for your attention!!