

Sampling low-dimensional Markovian dynamics for pre-asymptotically recovering reduced models from data with operator inference

Benjamin Peherstorfer
Shane McQuarrie

<https://arxiv.org/pdf/1908.11233.pdf>
<https://tinyurl.com/sv64umt>

CSE 392: SCIENTIF COMP MACH/DEEP LRN
10 December 2019



Setting

Consider the model reduction problem of reducing a large discrete dynamical system (the *full-order model* or FOM)

$$\mathbf{x}_{j+1} = \mathbf{f}(\mathbf{x}_j, \mathbf{u}_j)$$

to a smaller system (the *reduced-order model* or ROM)

$$\hat{\mathbf{x}}_{j+1} = \hat{\mathbf{f}}(\hat{\mathbf{x}}_j, \mathbf{u}_j)$$

where

$$\begin{array}{lll} \mathbf{u}_j \in \mathbb{R}^m & \mathbf{x}_j \in \mathbb{R}^n, & \mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n, \\ & \hat{\mathbf{x}}_j \in \mathbb{R}^r, & \hat{\mathbf{f}} : \mathbb{R}^r \times \mathbb{R}^m \rightarrow \mathbb{R}^r. \end{array}$$

with $r \ll n$ and $\hat{\mathbf{x}}_j$ related to \mathbf{x}_j . Usually $\mathbf{x}_j \approx V \hat{\mathbf{x}}_j$ where $V \in \mathbb{R}^{n \times r}$ is a POD basis matrix derived from FOM snapshot data.

Problem

Given a basis $V \in \mathbb{R}^{n \times r}$, the standard *intrusive* approach is to define

$$\tilde{\mathbf{f}}(\tilde{\mathbf{x}}, \mathbf{u}) := V^T \mathbf{f}(V\tilde{\mathbf{x}}, \mathbf{u}), \quad \tilde{\mathbf{x}}_0 := V^T \mathbf{x}_0.$$

For example,

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = A\mathbf{x} + B\mathbf{u}, \quad A \in \mathbb{R}^{n \times n}, \quad B \in \mathbb{R}^{n \times m},$$

$$\implies \tilde{\mathbf{f}}(\tilde{\mathbf{x}}, \mathbf{u}) = \underbrace{V^T A V}_{\tilde{A}} \tilde{\mathbf{x}} + \underbrace{V^T B}_{\tilde{B}} \mathbf{u}, \quad \tilde{A} \in \mathbb{R}^{r \times r}, \quad \tilde{B} \in \mathbb{R}^{r \times m},$$

The computational advantage is that $\tilde{A} = V^T A V$ and $\tilde{B} = V^T B$ can be precomputed. **But what if the operators A or B are unknown, or they are hard to access or construct explicitly?**

Approach: Operator Inference

Instead of computing $\tilde{A} \in \mathbb{R}^{r \times r}$ and $\tilde{B} \in \mathbb{R}^{r \times m}$ intrusively, assume that the ROM operator $\hat{\mathbf{f}}$ has the form

$$\hat{\mathbf{f}}(\hat{\mathbf{x}}) = \hat{A}\hat{\mathbf{x}} + \hat{B}\mathbf{u},$$

then solve a least-squares problem to get the best $\hat{A} \in \mathbb{R}^{r \times r}$ and $\hat{B} \in \mathbb{R}^{r \times m}$ according to the snapshot data [PW16]:

$$\arg \min_{\hat{A}, \hat{B}} \sum_{j=0}^{k-1} \left\| \hat{A}\hat{\mathbf{x}}_j + \hat{B}\mathbf{u}_j - \hat{\mathbf{x}}_{j+1} \right\|_2^2.$$

This is a relatively small ordinary linear least squares problem that decouples nicely, and is computationally inexpensive to solve. The problem can be generalized to more complicated forms of \mathbf{f} , for example,

$$\arg \min_{\hat{\mathbf{c}}, \hat{A}, \hat{H}, \hat{B}} \sum_{j=0}^{k-1} \left\| \hat{\mathbf{c}} + \hat{A}\hat{\mathbf{x}}_j + \hat{H}(\mathbf{x}_j \otimes \mathbf{x}_j) + \hat{B}\mathbf{u}_j - \hat{\mathbf{x}}_{j+1} \right\|_2^2.$$

Algorithm 1 Operator inference for reducing discrete systems

```
1: procedure OPINF( $\mathbf{f}$ ,  $\mathbf{x}_0$ ,  $\{\mathbf{u}_j\}_{j=0}^k$ ,  $r$ )
2:   for  $j = 0, 1, \dots, k - 1$  do
3:      $\mathbf{x}_{j+1} \leftarrow \mathbf{f}(\mathbf{x}_j, \mathbf{u}_j)$  ▷ Generate state snapshots.
4:   end for
5:    $V \leftarrow \text{POD\_basis} \left( \text{data} = \{\mathbf{x}_j\}_{j=0}^k, \text{rank} = r \right)$ 
6:   

---


7:   for  $j = 0, 1, \dots, k$  do:
8:      $\hat{\mathbf{x}}_j \leftarrow V^\top \mathbf{x}_j$  ▷ Project the trajectory.
9:   end for
10:  

---


11:   $\hat{A}, \hat{B} \leftarrow \arg \min \sum_{j=0}^{k-1} \left\| \hat{A} \hat{\mathbf{x}}_j + \hat{B} \mathbf{u}_j - \hat{\mathbf{x}}_{j+1} \right\|_2^2$  ▷ Infer operators.
12:  return  $\hat{A}, \hat{B}$ 
13: end procedure
```

Another Problem: Non-Markovian Dynamics

The intrusive operators \tilde{A} and \tilde{B} often inherit desirable properties from A and B (e.g., $\lambda_{\max}(A) < 0$ often implies $\lambda_{\max}(\tilde{A}) < 0$). However, the inferred operators \hat{A} and \hat{B} are **not** guaranteed to inherit such properties [Peh19] since $\tilde{A} \neq \hat{A}$ in general.

New Approach: Data “Re-projection”

After generating state snapshots $\{\mathbf{x}_j\}_{j=0}^k \subset \mathbb{R}^n$ and computing the basis $V \in \mathbb{R}^{n \times r}$, compute the projected snapshot data that would be produced by the corresponding intrusive model. Using this re-projected data with operator inference yields the intrusive model, that is, $\hat{A} = \tilde{A}$ and $\hat{B} = \tilde{B}$.

Algorithm 2 Operator inference with re-projected trajectories

```
1: procedure OPINFRP( $\mathbf{f}$ ,  $\mathbf{x}_0$ ,  $\{\mathbf{u}_j\}_{j=0}^k$ ,  $r$ )
2:   for  $j = 0, 1, \dots, k - 1$  do
3:      $\mathbf{x}_{j+1} \leftarrow \mathbf{f}(\mathbf{x}_j, \mathbf{u}_j)$  ▷ Generate state snapshots.
4:   end for
5:    $V \leftarrow \text{POD\_basis} \left( \text{data} = \{\mathbf{x}_j\}_{j=0}^k, \text{rank} = r \right)$ 
6:   

---


7:    $\hat{\mathbf{x}}_0 \leftarrow V^T \mathbf{x}_0$ 
8:   for  $j = 0, 1, \dots, k - 1$  do:
9:      $\hat{\mathbf{x}}_{j+1} \leftarrow V^T \mathbf{f}(V \hat{\mathbf{x}}_j, \mathbf{u}_j)$  ▷ Re-project the trajectory.
10:  end for ▷ (was  $\hat{\mathbf{x}}_j = V^T \mathbf{x}_j$  before.)
11:  

---


12:   $\hat{A}, \hat{B} \leftarrow \arg \min \sum_{j=0}^{k-1} \left\| \hat{A} \hat{\mathbf{x}}_j + \hat{B} \mathbf{u}_j - \hat{\mathbf{x}}_{j+1} \right\|_2^2$  ▷ Infer operators.
13:  return  $\hat{A}, \hat{B}$ 
14: end procedure
```

Results

Theorem (Re-projection)

Re-projection generates the trajectories obtained with the intrusive reduced model with initial condition $\tilde{\mathbf{x}}_0 = V^T \mathbf{x}_0$.

Corollary (Operator inference)

For a polynomial system of degree ℓ , if $k \geq m + \sum_{i=1}^{\ell} \binom{n}{i}$ and the data matrix \hat{X} is full rank, then the operator inference least squares problem with re-projected trajectories has a unique solution: the operators of the intrusive model.

Complexity of Algorithm 2: if $\mathbf{f} \in \mathcal{O}(F(n))$,

$$\underbrace{\mathcal{O}(kF(n))}_{\text{snapshot generation}} + \underbrace{\mathcal{O}(kn^2)}_{\text{basis computation}} + \underbrace{\mathcal{O}(\mathbf{k}(F(n) + nr))}_{\text{re-projection sampling}} + \underbrace{\mathcal{O}(kr^3)}_{\text{operator inference}}$$

Methodology

For a few different PDE-driven problems,

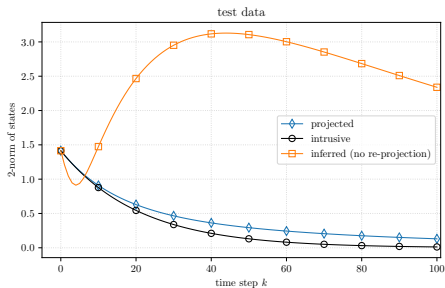
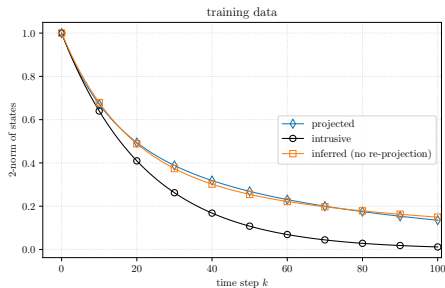
- Construct the FOM.
- Simulate the FOM to generate data and compute a basis.
- Compute the intrusive ROM operators (\tilde{A}).
- Use operator inference to compute the ROM operators.
- Re-project the trajectories, then use operator inference to compute the ROM operators (\hat{A}).
- Compare intrusive and inferred operators (check $\tilde{A} = \hat{A}$).

We examine

- A motivational toy problem
- A simple homogeneous heat equation
- The viscous Burgers' equation with inputs for boundary conditions

Results: Toy Problem

Let $\mathbf{f}(\mathbf{x}) = A\mathbf{x}$ where $\lambda(A) \subset [.9, .99]$.



Results: Heat Equation

Consider the one-dimensional heat equation

$$\begin{aligned}\frac{\partial x}{\partial t} - \mu \frac{\partial^2 x}{\partial \omega^2} &= 0 & \forall (\omega, t) \in (0, L) \times (0, T] \\ x(0, t) = x(L, t) &= 0 & \forall t \in [0, T], \\ x(\omega, 0) &= g(\omega) & \forall \omega \in [0, L].\end{aligned}$$

Discretized with simple finite differences, we obtain an ODE of the form

$$\frac{d\mathbf{x}}{dt} = A\mathbf{x}(t).$$

Two experiments:

- Discrete: Implicit Euler takes the form $\mathbf{x}_{j+1} = (I - \delta t A)^{-1} \mathbf{x}_j$. We successfully learned $V^\top (I - \delta t A)^{-1} V$ with re-projection.
- Continuous: with re-projection, $\hat{A} \rightarrow V^\top A V$ as $\delta t \rightarrow 0$ in the scheme used to compute the snapshots.

Results: Burgers' Equation

Consider the viscous Burgers' equation,

$$\begin{aligned}\frac{\partial x}{\partial t} + x \frac{\partial x}{\partial \omega} - \mu \frac{\partial^2 x}{\partial \omega^2} &= 0 & \forall (\omega, t) \in (-1, 1) \times (0, T] \\ x(-1, t) = u(t), \quad x(1, t) &= -u(t) & \forall t \in [0, T], \\ x(\omega, 0) &= 0 & \forall \omega \in [0, L].\end{aligned}$$

Discretized in space with finite differences and in time with Explicit Euler, this becomes a discrete system of the form

$$\mathbf{x}_{j+1} = A\mathbf{x}_j + H(\mathbf{x}_j \otimes \mathbf{x}_j) + Bu_j.$$

We confirmed that we can compute the intrusive model,

$$\tilde{\mathbf{x}}_{j+1} = V^T A V \tilde{\mathbf{x}}_j + V^T H(V \otimes V)(\tilde{\mathbf{x}}_j \otimes \tilde{\mathbf{x}}_j) + V^T B u_j,$$

via operator inference with re-projection.

References



Benjamin Peherstorfer.

Sampling low-dimensional markovian dynamics for pre-asymptotically recovering reduced models from data with operator inference.

arXiv preprint arXiv:1908.11233, 2019.



B. Peherstorfer and K. Willcox.

Data-driven operator inference for nonintrusive projection-based model reduction.

Computer Methods in Applied Mechanics and Engineering,
306:196–215, 2016.

Appendix

Why Re-projection Works (by example)

Consider the FOM $\mathbf{x}_{j+1} = \mathbf{f}(\mathbf{x}_j) := A\mathbf{x}_j$. Given a basis matrix V , the intrusive reduced model is $\tilde{\mathbf{x}}_{j+1} = \tilde{A}\tilde{\mathbf{x}}_j$ where $\tilde{A} = V^T A V$. In operator inference, we set $\hat{\mathbf{x}}_j := V^T \mathbf{x}_j$, then minimize

$$\sum_{j=0}^{k-1} \left\| \hat{A} \hat{\mathbf{x}}_j - \hat{\mathbf{x}}_{j+1} \right\|_2^2 = \sum_{j=0}^{k-1} \left\| \hat{A} V^T \mathbf{x}_j - V^T A \mathbf{x}_j \right\|_2^2,$$

which gives an answer $\hat{A} \approx V^T A V = \tilde{A}$ since $\mathbf{x}_j \approx V \hat{\mathbf{x}}_j$. On the other hand, re-projection sets $\hat{\mathbf{x}}_{j+1} = V^T \mathbf{f}(V \mathbf{x}_j) = V^T A V \mathbf{x}_j = \tilde{A} \mathbf{x}_j$, so the sum to minimize becomes

$$\sum_{j=0}^{k-1} \left\| \hat{A} \hat{\mathbf{x}}_j - \tilde{A} \hat{\mathbf{x}}_j \right\|_2^2 = \sum_{j=0}^{k-1} \left\| (\hat{A} - \tilde{A}) \hat{\mathbf{x}}_j \right\|_2^2,$$

so of course $\hat{A} = \tilde{A}$ as long as the system is overdetermined and well-conditioned.

Extension 1: Continuous Systems (ODEs)

The model reduction is most often posed in the continuous setting, i.e., the full-order model is a differential equation:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)).$$

The goal of re-projection is to learn the intrusive reduced model,

$$\frac{d\tilde{\mathbf{x}}}{dt} = \tilde{\mathbf{f}}(t, \tilde{\mathbf{x}}(t), \mathbf{u}(t)) := V^T \mathbf{f}(t, V\hat{\mathbf{x}}(t), \mathbf{u}(t)),$$

from data with operator inference.

In this setting, operator inference requires velocity snapshots $\{\hat{\mathbf{x}}_j\}_{j=1}^k$ that correspond to the state snapshots (these replace $\hat{\mathbf{x}}_{j+1}$ in the least squares problem). Vanilla operator inference sets $\hat{\mathbf{x}}_j = V^T \mathbf{f}(\mathbf{x}_j)$, while re-projection sets $\hat{\mathbf{x}}_j = V^T \mathbf{f}(V\hat{\mathbf{x}}_j) = V^T \mathbf{f}(VV^T \mathbf{x}_j)$. Only the velocity snapshots are modified; the states remain unchanged. However, the intrusive operator is only recovered in the limit as $\delta t \rightarrow 0$ in the scheme that generates the snapshots.

Algorithm 3 Operator inference with re-projected trajectories (continuous)

```
1: procedure OPINFRP2( $\mathbf{f}$ ,  $\{t_j\}_{j=0}^{k-1}$ ,  $\mathbf{x}_0$ ,  $\mathbf{u}$ ,  $r$ )
2:   for  $j = 0, 1, \dots, k-1$  do
3:      $\mathbf{x}_j \leftarrow \text{solve } \frac{d\mathbf{x}}{dt}(t_j) = \mathbf{f}(t_j, \mathbf{x}(t_j), \mathbf{u}(t_j)), \quad \mathbf{x}(0) = \mathbf{x}_0.$ 
4:   end for
5:    $V \leftarrow \text{POD\_basis} \left( \text{data} = \{\mathbf{x}_j\}_{j=0}^{k-1}, \text{rank} = r \right)$ 
6:   

---


7:   for  $j = 0, 1, \dots, k-1$  do:
8:      $\hat{\mathbf{x}}_j \leftarrow V^\top \mathbf{x}_j$ 
9:      $\dot{\hat{\mathbf{x}}}_j \leftarrow V^\top \mathbf{f}(V \hat{\mathbf{x}}_j, \mathbf{u}_j)$  ▷ “Re-project” the velocity.
10:  end for
11:  

---


12:   $\hat{A}, \hat{B} \leftarrow \arg \min \sum_{j=0}^{k-1} \left\| \hat{A} \hat{\mathbf{x}}_j + \hat{B} \mathbf{u}_j - \dot{\hat{\mathbf{x}}}_j \right\|_2^2$  ▷ Infer operators.
13:  return  $\hat{A}, \hat{B}$ 
14: end procedure
```

Extension 2: Parametric Systems

Consider the (discrete) full-order model

$$\mathbf{x}_{j+1} = \mathbf{f}(\mathbf{x}_j, \mathbf{u}_j; \boldsymbol{\mu}),$$

where $\boldsymbol{\mu} \in \mathbb{R}^p$ is a parameter. One way to deal with $\boldsymbol{\mu}$ is via interpolation:

- 1 Select parameter samples $\{\boldsymbol{\mu}_i\}_{i=1}^s \subset \mathbb{R}^p$ to train on.
- 2 Solve the FOM for each $\boldsymbol{\mu}_i$.
- 3 Compute a (global) POD basis from the resulting snapshots.
- 4 Use re-projected operator inference to compute a ROM for each $\boldsymbol{\mu}_i$, using only the data corresponding to that $\boldsymbol{\mu}_i$.
- 5 For a new parameter $\bar{\boldsymbol{\mu}}$, interpolate between the ROM operators for each $\boldsymbol{\mu}_i$ to get a ROM corresponding to $\bar{\boldsymbol{\mu}}$.

This methodology can be used in the continuous setting as well, but it only works well if the parameter dimension p is small—otherwise, the interpolation becomes unwieldy.