Optimizing oblique projections for nonlinear systems

Sam Otto, Alberto Padovan, and Clancy Rowley

5 May 2021
Outline

Goals and motivation

Optimizing oblique projections

Results for toy model

Application to high-dimensional nonlinear fluid flow

Conclusions and future work
Main goals

▶ We seek reduced-order models for nonlinear systems far from an equilibrium
  ▶ We assume we know a model for the full nonlinear system
  ▶ However, it is too high-dimensional, and we want a simpler approximation of it
▶ We aim to develop models via oblique projections of the governing equations
▶ We will determine these projections based on observations from data
▶ Disclaimer: this is not an “on-the-fly” approach
Our approach

- We need to model **dynamically significant** features of nonlinear systems and not just energetically significant ones (e.g., principal components).

- Linear model reduction approaches (e.g., balanced truncation) accomplish this by exploiting **linear** structure to find oblique projections.

- We don’t have the same nice structure for **nonlinear systems**, but we often have data from simulations or experiments.

- We will try to **optimize** an oblique projection for the nonlinear system so that the resulting reduced-order model predictions closely match a collection of trajectories from the full-order model.
Oblique projections
Oblique projections

- To define an oblique projection, we need to specify two subspaces, $V$ and $W$.

- In order for the projection $P_{V,W}$ to be well defined, these need to satisfy some conditions:
  - In particular, no nonzero element of $V$ is orthogonal to $W$. 
Projection-based reduced-order model

Consider a nonlinear model

\[
\frac{dx}{dt} = f(x, u), \quad x(0) = x_0
\]

\[
y = g(x).
\]

Applying the oblique projection \( P_V, W \) yields a reduced-order model

\[
\frac{d\hat{x}}{dt} = P_V, W f(\hat{x}, u), \quad \hat{x}(0) = P_V, W x_0
\]

\[
\hat{y} = g(\hat{x}).
\]

that evolves in the subspace \( V \).

We wish to choose the subspaces \( V, W \) so that \( \hat{y} \) approximates \( y \) well, for some set of inputs \( u \).
Existing methods for linear systems

▶ Some good approaches are known for linear systems

\[
\dot{x} = Ax + Bu \\
y = Cx
\]

▶ For instance, balanced truncation gives a bound on \( \|y - \hat{y}\|_2 / \|u\|_2 \), that is provably close to the optimal achievable by any linear system.

▶ \( H_2 \) optimal model reduction gives subspaces that minimize the 2-norm of the linear operator that maps \( u \mapsto (y - \hat{y}) \).
Nonlinear systems

- For a **nonlinear system**, if the states remain near an equilibrium point (e.g., because of control), it is reasonable to linearize about this equilibrium, and use the resulting subspaces for projection of the full nonlinear system.

- If the system does not stay near an equilibrium point, then the typical approach is to find **principal components** of a given dataset:
  - Subspace $V$ spanned by the first several principal components
  - Subspace $W = V$ (orthogonal projection)

- This approach often does not work well:
  - States with small norm can play an important role in the dynamics.
Outline

Goals and motivation

Optimizing oblique projections

Results for toy model

Application to high-dimensional nonlinear fluid flow

Conclusions and future work
The optimization problem

- Consider observations $\{y_l\}_{l=0}^{L-1}$ along a trajectory from the full model and observations $\{\hat{y}_l\}_{l=0}^{L-1}$ from the reduced-order model.

- Let $L_y : \mathbb{R}^{\text{dim}y} \rightarrow [0, \infty)$ be a smooth loss function for the predicted system outputs. We'd like to minimize

$$J(V, W) = \frac{1}{L} \sum_{l=0}^{L-1} L_y(\hat{y}_l - y_l).$$

- But we need to ensure that the subspace pairs $(V, W)$ satisfy the non-orthogonality condition, so we introduce a regularization $\rho(V, W)$ that enforces this constraint, and define the objective

$$J(V, W) = \frac{1}{L} \sum_{l=0}^{L-1} L_y(\hat{y}_l - y_l) + \gamma \rho(V, W).$$
Optimizing over subspaces

- The parameters in our optimization are the subspaces $V$, $W$.
- Can define an $r$-dimensional subspace $V$ of $\mathbb{R}^n$ as the span of $r$ vectors in $\mathbb{R}^n$; or equivalently by the range of an $n \times r$ matrix $\Phi$.
- However, many different matrices $\Phi$ span the same subspace (change of basis)
- The set of all $r$-dimensional subspaces in $\mathbb{R}^n$ can be endowed with the structure of a smooth manifold called the Grassmann manifold $G(n, r)$
- We optimize over two copies of this manifold, with the additional non-orthogonality constraint between $V$ and $W$. 
Optimization algorithm

- We optimize the subspaces $(V, W)$ defining the reduced-order model using a geometric conjugate gradient algorithm.\(^1\)\(^2\)
- The gradient is computed using an adjoint sensitivity method.
- This entails solving a linear ODE with the same dimension as the reduced-order model backwards in time.
- The cost is dominated by the need to assemble the reduced-order model and to evaluate the full model dynamics at quadrature points along the predicted trajectory.

---

\(^1\)Absil et al., “Optimization Algorithms on Matrix Manifolds”, 2008
Outline

Goals and motivation

Optimizing oblique projections

Results for toy model

Application to high-dimensional nonlinear fluid flow

Conclusions and future work
Nonlinear toy model

\[
\begin{align*}
\dot{x}_1 &= -x_1 + 15x_1x_3 + u \\
\dot{x}_2 &= -2x_2 + 15x_2x_3 + u \\
\dot{x}_3 &= -5x_3 + u \\
y &= x_1 + x_2 + x_3,
\end{align*}
\]

- The state $x_3$ is dynamically important, but remains small compared with $x_1$ and $x_2$ due to its fast decay rate.
- The linearized dynamics do not capture the system’s behavior away from the origin, which exhibits transient growth.
- We seek 2-dimensional reduced-order models and compare to POD/Galerkin and oblique projection using subspaces identified by balanced truncation.
Training trajectories

We find the subspaces $V, W$ using two nonlinear impulse-response trajectories with magnitudes $u_0 = 0.5, 1.0$. 
We tested our optimized Petrov-Galerkin model on 50 nonlinear impulse response trajectories with magnitudes drawn uniformly at random from the interval $[0, 1]$. 
Performance using a different input signal

The model also has excellent performance with other input signals like $u(t) = \sin t$. 
Outline

Goals and motivation

Optimizing oblique projections

Results for toy model

Application to high-dimensional nonlinear fluid flow

Conclusions and future work
We consider a spatially discretized incompressible mixing layer fluid flow with non-periodic dynamics.

Flow moves left to right and is faster on top than on the bottom. This leads to vortex formation through a Kelvin-Helmholtz instability.

The full-order model has $1.2 \times 10^5$ states and we seek 50-dimensional reduced-order models.
Short horizon performance

![Graphs showing trajectory errors for training and testing with POD and optimized models.](image)

- We train on 20 trajectories with 15 snapshots in each.
- We compare with a POD/Galerkin model. The Petrov-Galerkin model based on balanced truncation about an equilibrium rapidly diverged.
- \(E_{\text{train/test}}\) is the average energy of training/testing trajectories.
Testing data predictions

Vorticity at $t = 19.6$

Vorticity at $t = 39.2$
Long horizon performance

Training Trajectory Error

Vorticity at $t = 140$

$\frac{(\hat{q} - q)^T (\hat{q} - q)}{E_{test}}$

$\frac{(\hat{q} - q)^T (\hat{q} - q)}{E_{test}}$

$\frac{(\hat{q} - q)^T (\hat{q} - q)}{E_{test}}$
Outline

Goals and motivation

Optimizing oblique projections

Results for toy model

Application to high-dimensional nonlinear fluid flow

Conclusions and future work
Conclusions

- We determine (locally) optimal subspaces for constrained projection of known high-dimensional dynamics.
- We optimize the subspaces using a geometric conjugate-gradient algorithm on two copies of the Grassmann manifold, with a non-orthogonality constraint.
- On a toy nonlinear model, the method performs significantly better than projection onto subspaces determined from a linearization about an equilibrium.
- On a high-dimensional fluid example, the method performs significantly better than standard approaches (POD/principal components).
Future direction: learn nonlinear projection operators

- Build and train an autoencoder where the decoder is always a left inverse of the encoder

- Such an autoencoder defines a smooth nonlinear projection operator that can be used for model reduction