Model Order Reduction of the Navier-Stokes Equations at High Reynolds Number

Maciej Balajewicz\textsuperscript{1}  Earl Dowell\textsuperscript{2}  Bernd Noack\textsuperscript{3}

\textsuperscript{1}Aeronautics and Astronautics  Stanford University

\textsuperscript{2}Mechanical Engineering  Duke University

\textsuperscript{3}Département Fluides, Thermique, Combustion  Institut PPRIME,  Université de Poitiers

December 10, 2012
Background and motivation

Common elements in most Model Order Reduction (MOR) techniques:

1. Spectral discretization

\[ u(x, t) \approx \sum_{i=1}^{n} a_i(t)u_i(x) \]  

2. Projection

\[ \langle v_i, R \left( \sum_{i=1}^{n} a_i u_i \right) \rangle_{\Omega} = 0 \]
Common elements in most Model Order Reduction (MOR) techniques:

1. Spectral discretization

\[ u(x, t) \approx \sum_{i=1}^{n} a_i(t)u_i(x) \]  

2. Projection

\[ \langle v_i, R \left( \sum_{i=1}^{n} a_i u_i \right) \rangle_{\Omega} = 0 \]
Common elements in most Model Order Reduction (MOR) techniques:

1. Spectral discretization

\[ u(x, t) \approx \sum_{i=1}^{n} a_i(t)u_i(x) \]  

2. Projection

\[ \langle v_i, R \left( \sum_{i=1}^{n} a_i u_i \right) \rangle_\Omega = 0 \]
Background and motivation

One popular approach: first $n$ left-singular vectors of $M$, i.e.

$$u_i = U(:, i)$$

where $M = U \Sigma V^T$ and

$$M = \begin{bmatrix}
  u(x_1, t_1) & u(x_1, t_2) & \cdots & u(x_1, t_{N_t}) \\
  u(x_2, t_1) & u(x_2, t_2) & \cdots & u(x_2, t_{N_t}) \\
  \vdots & \vdots & \ddots & \vdots \\
  u(x_{N_x}, t_1) & u(x_{N_x}, t_2) & \cdots & u(x_{N_x}, t_{N_t})
\end{bmatrix}$$

These basis functions are optimal in the sense that no other basis functions capture a greater proportion of kinetic energy of the flow.
Background and motivation

► One popular approach: first $n$ left-singular vectors of $M$, i.e.
$u_i = U(:, i)$ where $M = U \Sigma V^T$ and

$$M = \begin{bmatrix}
    u(x_1, t_1) & u(x_1, t_2) & \cdots & u(x_1, t_{N_t}) \\
    u(x_2, t_1) & u(x_2, t_2) & \cdots & u(x_2, t_{N_t}) \\
    \vdots & \vdots & \ddots & \vdots \\
    u(x_{N_x}, t_1) & u(x_{N_x}, t_2) & \cdots & u(x_{N_x}, t_{N_t})
\end{bmatrix} \tag{3}$$

► These basis functions are optimal in the sense that no other basis functions capture a greater proportion of kinetic energy of the flow.
Turbulence is a multi-scale phenomenon: large scale flow features are broken down into smaller and smaller scales until the scales are fine enough that viscous forces can dissipate their energy.

Application of any POD-based MOR strategy to a turbulent flow is problematic because POD, by construction, is biased toward the large, energy containing scales of the turbulent flow.

Reduced Order Models (ROMs) generated using only the first most energetic POD basis functions are, therefore, not endowed with the natural energy dissipation of the smaller, lower energy turbulent scales.
Turbulence is a multi-scale phenomenon: large scale flow features are broken down into smaller and smaller scales until the scales are fine enough that viscous forces can dissipate their energy.

Application of any POD-based MOR strategy to a turbulent flow is problematic because POD, by construction, is biased toward the large, energy containing scales of the turbulent flow.

Reduced Order Models (ROMs) generated using only the first most energetic POD basis functions are, therefore, not endowed with the natural energy dissipation of the smaller, lower energy turbulent scales.
Background and motivation

- Turbulence is a multi-scale phenomenon: large scale flow features are broken down into smaller and smaller scales until the scales are fine enough that viscous forces can dissipate their energy.
- Application of any POD-based MOR strategy to a turbulent flow is problematic because POD, by construction, is biased toward the large, energy containing scales of the turbulent flow.
- Reduced Order Models (ROMs) generated using only the first most energetic POD basis functions are, therefore, not endowed with the natural energy dissipation of the smaller, lower energy turbulent scales.
Background and motivation

Classical benchmark: Incompressible flow inside a square, two-dimensional lid-driven cavity at $Re_u = 3 \times 10^4$
Background and motivation

POD basis functions of the lid-driven cavity

(a) $u_1$  (b) $u_2$  (c) $u_{10}$

(d) $u_{20}$  (e) $u_{50}$  (f) $u_{200}$
Percent of turbulent kinetic energy, \( e(t) \equiv \frac{1}{2} \int_{\Omega} |u(x, t)|^2 \, dx \) captured by the first \( n \) basis functions, \( u_i \) of the lid-driven cavity:

<table>
<thead>
<tr>
<th>( n )</th>
<th>( % )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16.06</td>
</tr>
<tr>
<td>2</td>
<td>29.21</td>
</tr>
<tr>
<td>3</td>
<td>37.45</td>
</tr>
<tr>
<td>4</td>
<td>44.88</td>
</tr>
<tr>
<td>5</td>
<td>50.37</td>
</tr>
<tr>
<td>10</td>
<td>67.16</td>
</tr>
<tr>
<td>20</td>
<td>82.40</td>
</tr>
<tr>
<td>50</td>
<td>93.21</td>
</tr>
<tr>
<td>200</td>
<td>99.31</td>
</tr>
</tbody>
</table>
Background and motivation

Standard POD-Galerkin ROMs of the lid-driven cavity

- ROM performance quantified using the turbulent kinetic energy, \( e(t) \equiv \frac{1}{2} \int_{\Omega} |u(x, t)|^2 \, dx \)
Background and motivation

Standard POD-Galerkin ROMs of the lid-driven cavity

- ROM performance quantified using the turbulent kinetic energy, $e(t) \equiv \frac{1}{2} \int_{\Omega} |u(x, t)|^2 \, dx$

\[ e(t) \begin{cases} \text{DNS} \\ \text{$n = 5$ POD-Galerkin ROM} \end{cases} \]
Background and motivation

Standard POD-Galerkin ROMs of the lid-driven cavity

- ROM performance quantified using the turbulent kinetic energy, $e(t) \equiv \frac{1}{2} \int_{\Omega} |\mathbf{u}(\mathbf{x}, t)|^2 d\mathbf{x}$

![Graph showing $e(t)$ over time for different ROMs](image)
Standard POD-Galerkin ROMs of the lid-driven cavity

- ROM performance quantified using the turbulent kinetic energy, $e(t) \equiv 1/2 \int_{\Omega} |\mathbf{u}(x, t)|^2 \, dx$
Background and motivation

Standard POD-Galerkin ROMs of the lid-driven cavity

- ROM performance quantified using the turbulent kinetic energy, $e(t) \equiv \frac{1}{2} \int_{\Omega} |u(x, t)|^2 \, dx$

![Graph showing the performance of different ROMs over time, with DNS and several POD-Galerkin ROMs. The graph plots $e(t)$ on a logarithmic scale against time $t$. The DNS line is grey, and there are lines for $n = 5$, $n = 20$, $n = 50$, and $n = 200$ POD-Galerkin ROMs. The ROMs with higher $n$ values show better performance compared to DNS.]
Standard POD-Galerkin ROMs of the lid-driven cavity

▶ ROM performance quantified using the turbulent kinetic energy, \( e(t) \equiv \frac{1}{2} \int_{\Omega} |u(x, t)|^2 \, dx \)

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{plot.png}
\caption{Comparison of ROM performance for different POD-Galerkin ROMs with DNS.}
\end{figure}
Basic idea: Instead of using *just* the most energetic POD basis functions, include a few lower energy POD basis functions so that dissipative scales are resolved.

- For example, consider we are interested in forming a $n = 3$ ROM.
- Standard approach: use the first 3 most energetic POD basis functions, i.e. $u_1$, $u_2$, and $u_3$
- Proposed Approach: $u_1$, $u_2$, and $u_5$
Proposed new approach

Basic idea: Instead of using just the most energetic POD basis functions, include a few lower energy POD basis functions so that dissipative scales are resolved.

- For example, consider we are interested in forming a $n = 3$ ROM.
- Standard approach: use the first 3 most energetic POD basis functions, i.e. $u_1$, $u_2$, and $u_3$
- Proposed Approach: $u_1$, $u_2$, and $u_5$
Proposed new approach

Basic idea: Instead of using *just* the most energetic POD basis functions, include a few lower energy POD basis functions so that dissipative scales are resolved.

- For example, consider we are interested in forming a $n = 3$ ROM.
- Standard approach: use the first 3 most energetic POD basis functions, i.e. $u_1$, $u_2$, and $u_3$
- Proposed Approach: $u_1$, $u_2$, and $u_5$
Proposed new approach

Basic idea: Instead of using *just* the most energetic POD basis functions, include a few lower energy POD basis functions so that dissipative scales are resolved.

- For example, consider we are interested in forming a $n = 3$ ROM.
- Standard approach: use the first 3 most energetic POD basis functions, i.e. $u_1$, $u_2$, and $u_3$
- Proposed Approach: $u_1$, $u_2$, and $u_5$
Proposed new approach

Generalization: We search for the $n$ new basis functions in the range of $N$ standard POD basis functions (where $N > n$).

- If we label the $n$ new basis functions as $\tilde{u}_i$;
- We can write

$$\tilde{U} = UX \tag{4}$$

where $X \in \mathbb{R}^{N \times n}$ is an orthonormal ($X^T X = I_{n \times n}$) transformation matrix and the basis functions are vectorized and assembled as follows

$$U = \begin{bmatrix} \text{vec}(u_1) & \text{vec}(u_2) & \cdots & \text{vec}(u_N) \end{bmatrix} \tag{5a}$$

$$\tilde{U} = \begin{bmatrix} \text{vec}(\tilde{u}_1) & \text{vec}(\tilde{u}_2) & \cdots & \text{vec}(\tilde{u}_n) \end{bmatrix} \tag{5b}$$
Proposed new approach

Generalization: We search for the \( n \) new basis functions in the range of \( N \) standard POD basis functions (where \( N > n \)).

- If we label the \( n \) new basis functions as \( \tilde{u}_i \);
- We can write

\[
\tilde{U} = U X
\]

where \( X \in \mathbb{R}^{N \times n} \) is an orthonormal \((X^T X = I_{n \times n})\) transformation matrix and the basis functions are vectorized and assembled as follows

\[
U = \begin{bmatrix}
\text{vec}(u_1) & \text{vec}(u_2) & \cdots & \text{vec}(u_N)
\end{bmatrix}
\]

\[
\tilde{U} = \begin{bmatrix}
\text{vec}(\tilde{u}_1) & \text{vec}(\tilde{u}_2) & \cdots & \text{vec}(\tilde{u}_n)
\end{bmatrix}
\]
Proposed new approach

Generalization: We search for the \( n \) new basis functions in the range of \( N \) standard POD basis functions (where \( N > n \)).

- If we label the \( n \) new basis functions as \( \tilde{u}_i \);
- We can write

\[
\tilde{U} = UX
\]  

(4)

where \( X \in \mathbb{R}^{N \times n} \) is an orthonormal (\( X^T X = I_{n \times n} \)) transformation matrix and the basis functions are vectorized and assembled as follows

\[
U = \begin{bmatrix}
\text{vec} \left( u_1 \right) & \text{vec} \left( u_2 \right) & \cdots & \text{vec} \left( u_N \right)
\end{bmatrix}
\]  

(5a)

\[
\tilde{U} = \begin{bmatrix}
\text{vec} \left( \tilde{u}_1 \right) & \text{vec} \left( \tilde{u}_2 \right) & \cdots & \text{vec} \left( \tilde{u}_n \right)
\end{bmatrix}
\]  

(5b)
Proposed new approach

We have developed an algorithm for X that uses the turbulent kinetic energy (tke) equation as a side constraint to the standard POD/SVD snapshot approach. The algorithm is computationally efficient and thus can be implemented using standard MATLAB constrained optimization algorithms. Currently, our algorithm is limited to steady Dirichlet, ambient flow or free-stream conditions at infinity, or periodic boundary conditions.

Lid driven cavity, revisited

\[ e(t) \]

\[ n = 5 \]

\[ n = 10 \]

\[ n = 20 \]

Time evolution of the error \( e(t) \) for different values of \( n \).
Lid driven cavity, revisited

\[ e(t) \]

\[ \log_{10}(e(t)) \]

\[ n = 5 \]

\[ n = 10 \]

\[ n = 20 \]
Conclusions

- Our approach models small scales of the flow *directly* using basis functions that are different from the standard POD basis functions.
- For many boundary conditions, a computationally efficient algorithm that can be implemented in MATLAB using *fmincon* is available.
- Details, results from a second benchmark, and MATLAB code are available online: arXiv:1211.1720 [physics.flu-dyn]
Conclusions

- Our approach models small scales of the flow directly using basis functions that are different from the standard POD basis functions.
- For many boundary conditions, a computationally efficient algorithm that can be implemented in MATLAB using fmincon is available.
- Details, results from a second benchmark, and MATLAB code are available online: arXiv:1211.1720 [physics.flu-dyn]
Conclusions

- Our approach models small scales of the flow directly using basis functions that are different from the standard POD basis functions.

- For many boundary conditions, a computationally efficient algorithm that can be implemented in MATLAB using fmincon is available.

- Details, results from a second benchmark, and MATLAB code are available online: arXiv:1211.1720 [physics.flu-dyn]
Conclusions

- Our approach models small scales of the flow *directly* using basis functions that are different from the standard POD basis functions.
- For many boundary conditions, a computationally efficient algorithm that can be implemented in MATLAB using `fmincon` is available.
- Details, results from a second benchmark, and MATLAB code are available online: arXiv:1211.1720 [physics.flu-dyn]