Proper Orthogonal Decomposition (POD) Closure Models for Turbulent Flows

Irina K. Tezaur
Extreme Scale Data Science & Analytics Department, Sandia National Laboratories, Livermore CA

Data Science Reading Group Seminar

Thursday, May 18, 2017
Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison

Zhu Wang\textsuperscript{a}, Imran Akhtar\textsuperscript{b}, Jeff Borggaard\textsuperscript{a}, Traian Iliescu\textsuperscript{a,*}

\textsuperscript{a}Department of Mathematics, Virginia Tech, Blacksburg, VA 24061-0123, USA
\textsuperscript{b}Department of Mechanical Engineering, NUST College of Electrical & Mechanical Engineering, National University of Sciences & Technology, Islamabad, Pakistan
Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison

Zhu Wang\textsuperscript{a}, Imran Akhtar\textsuperscript{b}, Jeff Borggaard\textsuperscript{a}, Traian Iliescu\textsuperscript{a,}\textsuperscript{*}

\textsuperscript{a}Department of Mathematics, Virginia Tech, Blacksburg, VA 24061-0123, USA
\textsuperscript{b}Department of Mechanical Engineering, NUST College of Electrical & Mechanical Engineering, National University of Sciences & Technology, Islamabad, Pakistan

• **My interest in this work:**
• **My interest in this work:**
  • I am interested in **stable POD-Galerkin** ROMs for fluid problems (next slides).
• **My interest in this work:**
  • I am interested in **stable POD-Galerkin** ROMs for fluid problems (next slides).
  • Authors of this paper have **similar requirements** for ROMs as me: use ROMs for long-time integration, “extreme model reduction”, QoI = statistics of flow, etc.
My interest in this work:

- I am interested in **stable POD-Galerkin** ROMs for fluid problems (next slides).
- Authors of this paper have **similar requirements** for ROMs as me: use ROMs for long-time integration, “extreme model reduction”, QoI = statistics of flow, etc.
- Methods in this paper are alternatives to my work on **basis rotation**\(^1\).

---

• **My interest in this work:**
  • I am interested in *stable POD-Galerkin* ROMs for fluid problems (next slides).
  • Authors of this paper have *similar requirements* for ROMs as me: use ROMs for long-time integration, “extreme model reduction”, QoI = statistics of flow, etc.
  • Methods in this paper are alternatives to my work on *basis rotation*¹.
  • I am working with T. Iliescu to try to understand how to extend methods such as those in the paper to *compressible flow* problems and to make them more rigorous.

Outline

1. Motivation/Background
2. Section 2: POD/Galerkin ROMs for Incompressible Flows
3. Background on Turbulence Modeling/Large Eddy Simulation
4. Section 3: POD Closure Models
   • Mixing Length (ML)
   • Smagorinsky (S)
   • Variational Multi-Scale (VMS)
   • Dynamic Subgrid (DS)
5. Section 4.1: Computational Efficiency
6. Section 4.2: Numerical Results for 3D Flow Around Cylinder
7. Section 5 and Beyond: Future/Follow-Up Work
   [8. Basis Rotation (My Work – an Alternative Approach)]
Outline

1. Motivation/Background
2. Section 2: POD/Galerkin ROMs for Incompressible Flows
3. Background on Turbulence Modeling/Large Eddy Simulation
4. Section 3: POD Closure Models
   • Mixing Length (ML)
   • Smagorinsky (S)
   • Variational Multi-Scale (VMS)
   • Dynamic Subgrid (DS)
5. Section 4.1: Computational Efficiency
6. Section 4.2: Numerical Results for 3D Flow Around Cylinder
7. Section 5 and Beyond: Future/Follow-Up Work
   [8. Basis Rotation (My Work – an Alternative Approach)]
Motivation

- We are interested in the *compressible captive-carry problem*. 
Motivation

• We are interested in the **compressible captive-carry problem**.

• Of **primary interest** are **long-time predictive simulations**: ROM run at same parameters as FOM but much longer in time.
Motivation

- We are interested in the **compressible captive-carry problem**.

- Of primary interest are **long-time predictive simulations**: ROM run at same parameters as FOM but much longer in time.
  
  → **QoIs**: statistics of flow, e.g., pressure Power Spectral Densities (PSDs) [right].
Motivation

• We are interested in the **compressible captive-carry problem**.

• Of **primary interest** are **long-time predictive simulations**: ROM run at same parameters as FOM but much longer in time.
  
  → **QoIs**: statistics of flow, e.g., pressure **Power Spectral Densities (PSDs)** [right].

• **Secondary interest**: ROMs robust w.r.t. **parameter changes** (e.g., Reynolds, Mach number) for enabling **uncertainty quantification**.
Proper Orthogonal Decomposition (POD)/Galerkin method to model reduction

High fidelity CFD simulations:
- Snapshot 1
- Snapshot 2
- \vdots
- Snapshot K

Fluid modal decomposition (POD):
\[ u(x, t) \approx \sum_{k=1}^{M} a_{M,k}(t) \phi_k(x) \]

Galerkin projection of fluid PDEs:
\[ (\phi_k, \dot{u} + \nabla \cdot F(u)) = 0 \]

- **Snapshot matrix:** \( X = (x^1, \ldots, x^K) \in \mathbb{R}^{NxK} \)
- **SVD:** \( X = U\Sigma V^T \)
- **Truncation:** \( \Phi_M = (\phi_1, \ldots, \phi_M) = U(:, 1:M) \)

Basis energy:
\[ \sigma_i^2 = \sum_{i=1}^{M} \sigma_i^2 \]

“Small” ROM ODE system:
\[ \dot{a}_{M,k} = f(a_{M,1}, \ldots, a_{M,M}) \]

FOM = full order model
\( N = \# \text{ of dofs in FOM} \)
\( K = \# \text{ of snapshots} \)
\( M = \# \text{ of dofs in ROM} \)
\( M \ll N, M \ll K \)
Extreme Model Reduction

• Most realistic applications (e.g., high Re compressible cavity): basis that captures >99% snapshot energy is required to accurately reproduce snapshots.
  → leads to $M > O(1000)$ except for toy problems and/or low-fidelity models.

• Higher order modes are in general unreliable for prediction, so including them in the basis is unlikely to improve the predictive capabilities of a ROM.

Figure (right) shows projection error for POD basis constructed using 800 snapshots for cavity problem. Dashed line = end of snapshot collection period.

We are looking for an approach that enables extreme model reduction: ROM basis size is $O(10)$ or $O(100)$. 
Projection-based MOR necessitates *truncation*.
Mode truncation instability

Projection-based MOR necessitates *truncation*.

- POD is, by definition and design, biased towards the *large, energy producing* scales of the flow (i.e., modes with large POD eigenvalues).
Mode truncation instability

Projection-based MOR necessitates truncation.

- POD is, by definition and design, biased towards the large, energy producing scales of the flow (i.e., modes with large POD eigenvalues).
- Truncated/unresolved modes are negligible from a data compression point of view (i.e., small POD eigenvalues) but are crucial for the dynamical equations.
Mode truncation instability

Projection-based MOR necessitates *truncation*.

- POD is, by definition and design, biased towards the *large, energy producing* scales of the flow (i.e., modes with large POD eigenvalues).
- Truncated/unresolved modes are negligible from a *data compression* point of view (i.e., small POD eigenvalues) but are crucial for the *dynamical equations*.
- For fluid flow applications, higher-order modes are associated with energy *dissipation*.
Mode truncation instability

Projection-based MOR necessitates **truncation**.

- POD is, by definition and design, biased towards the **large, energy producing** scales of the flow (i.e., modes with large POD eigenvalues).
- Truncated/unresolved modes are negligible from a **data compression** point of view (i.e., small POD eigenvalues) but are crucial for the **dynamical equations**.
- For fluid flow applications, higher-order modes are associated with energy **dissipation**

→ low-dimensional ROMs can be **inaccurate** and **unstable**.
Mode truncation instability

Projection-based MOR necessitates **truncation**.

- POD is, by definition and design, biased towards the *large, energy producing* scales of the flow (i.e., modes with large POD eigenvalues).
- Truncated/unresolved modes are negligible from a *data compression* point of view (i.e., small POD eigenvalues) but are crucial for the *dynamical equations*.
- For fluid flow applications, higher-order modes are associated with energy *dissipation*

→ low-dimensional ROMs can be **inaccurate** and **unstable**.

For a low-dimensional ROM to be stable and accurate, the **truncated/unresolved subspace** must be accounted for.
Mode truncation instability

- POD is, by definition and design, biased towards the **large, energy producing** scales of the flow (i.e., modes with large POD eigenvalues).
- Truncated/unresolved modes are negligible from a **data compression** point of view (i.e., small POD eigenvalues) but are crucial for the **dynamical equations**.
- For fluid flow applications, higher-order modes are associated with energy **dissipation**

\[ \Rightarrow \text{low-dimensional ROMs can be } \textbf{inaccurate and unstable.} \]

For a low-dimensional ROM to be stable and accurate, the **truncated/unresolved subspace** must be accounted for.

**Turbulence Modeling** (this paper)

**Subspace Rotation** (our approach)
Outline

1. Motivation/Background

2. **Section 2: POD/Galerkin ROMs for Incompressible Flows**

3. Background on Turbulence Modeling/Large Eddy Simulation

4. Section 3: POD Closure Models
   - Mixing Length (ML)
   - Smagorinsky (S)
   - Variational Multi-Scale (VMS)
   - Dynamic Subgrid (DS)

5. Section 4.1: Computational Efficiency

6. Section 4.2: Numerical Results for 3D Flow Around Cylinder

7. Section 5 and Beyond: Future/Follow-Up Work

8. Basis Rotation (My Work – an Alternative Approach)
Section 2: POD-Galerkin-ROM (POD-G-ROM) for Incompressible Flow

- Governing equations of *incompressible flow*:

\[
\begin{align*}
\mathbf{u}_t - Re^{-1} \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p &= 0 \\
\nabla \cdot \mathbf{u} &= 0
\end{align*}
\]
Section 2: POD-Galerkin-ROM (POD-G-ROM) for Incompressible Flow

• Governing equations of *incompressible flow*:

\[
\begin{align*}
\begin{cases}
    u_t - Re^{-1} \Delta u + (u \cdot \nabla)u + \nabla p = 0 \\
    \nabla \cdot u = 0
\end{cases}
\end{align*}
\]  

(1)

• POD approximation of velocity solution\(^2\) \(u\):

\[
 u(x, t) \approx u_r(x, t) = U(x) + \sum_{j=1}^{r} a_j(t) \varphi_j(x),
\]

where \(U(x) = \) base flow, \(\varphi_j(x) = \) POD modes.

\(^2\) Pressure ROM can be obtained by solving pressure-Poisson equation. Pressure term drops out from (1) following projection due to BCs. See [36,56].
Section 2: POD-Galerkin-ROM (POD-G-ROM) for Incompressible Flow

- Governing equations of *incompressible flow*:

\[
\begin{align*}
\frac{\partial \mathbf{u}}{\partial t} - Re^{-1} \nabla \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p &= 0, \\
\nabla \cdot \mathbf{u} &= 0,
\end{align*}
\]  

(1)

- POD approximation of velocity solution\(^2\) \(\mathbf{u}\):

\[
\mathbf{u}(x, t) \approx \mathbf{u}_r(x, t) \equiv \mathbf{U}(x) + \sum_{j=1}^{r} a_j(t) \phi_j(x),
\]

(6)

where \(\mathbf{U}(x) = \text{base flow}, \phi_j(x) = \text{POD modes.}\)

- Projecting (1) onto reduced basis \(\phi_j(x)\), the following **POD-G(alerkin)-ROM** is obtained:

\[
\left( \frac{\partial \mathbf{u}_r}{\partial t}, \phi \right) + ((\mathbf{u}_r \cdot \nabla) \mathbf{u}_r, \phi) + \left( \frac{2}{Re} \mathbb{D}(\mathbf{u}_r), \nabla \phi \right) = 0 \quad \forall \phi \in \mathbf{X}^r,
\]

(7)

where \(\mathbf{X}^r = \text{reduced subspace}, \mathbb{D}(\mathbf{u}_r) = \frac{1}{2} \nabla \mathbf{u}_r + \frac{1}{2} (\nabla \mathbf{u}_r)^T = \text{deformation tensor of} \ \mathbf{u}_r.\)

\(^2\) Pressure ROM can be obtained by solving pressure-Poisson equation. Pressure term drops out from (1) following projection due to BCs. See [36,56].
Section 2: POD-G-ROM for Incompressible Flow

- **POD-G-ROM** algebraic system:

\[
\dot{a}_k(t) = b_k + \sum_{m=1}^{r} A_{km} a_m(t) + \sum_{m=1}^{r} \sum_{n=1}^{r} B_{kmn} a_n(t) a_m(t),
\]

where:

\[
b_k = - (\varphi_k, U \cdot \nabla U) - \frac{2}{Re} \left( \nabla \varphi_k, \frac{\nabla U + \nabla U^T}{2} \right),
\]

\[
A_{km} = -(\varphi_k, U \cdot \nabla \varphi_m) - (\varphi_k, \varphi_m \cdot \nabla U) - \frac{2}{Re} \left( \nabla \varphi_k, \frac{\nabla \varphi_m + \nabla \varphi_m^T}{2} \right),
\]

\[
B_{kmn} = -(\varphi_k, \varphi_m \cdot \nabla \varphi_n).
\]
Section 2: POD-G-ROM for Incompressible Flow

• **POD-G-ROM** algebraic system:

\[
\dot{a}_k(t) = b_k + \sum_{m=1}^{r} A_{km} a_m(t) + \sum_{m=1}^{r} \sum_{n=1}^{r} B_{kmn} a_n(t) a_m(t),
\]

where:

\[
b_k = - (\varphi_k, U \cdot \nabla U) - \frac{2}{Re} \left( \nabla \varphi_k, \frac{\nabla U + \nabla U^T}{2} \right),
\]

\[
A_{km} = - (\varphi_k, U \cdot \nabla \varphi_m) - (\varphi_k, \varphi_m \cdot \nabla U) - \frac{2}{Re} \left( \nabla \varphi_k, \frac{\nabla \varphi_m + \nabla \varphi_m^T}{2} \right),
\]

\[
B_{kmn} = - (\varphi_k, \varphi_m \cdot \nabla \varphi_n).
\]

• **POD-G-ROM** in matrix form:

\[
\dot{a} = b + Aa + a^T B a
\]
Section 2: POD-G-ROM for Incompressible Flow

- **POD-G-ROM** algebraic system:

\[
\dot{a}_k(t) = b_k + \sum_{m=1}^{r} A_{km} a_m(t) + \sum_{m=1}^{r} \sum_{n=1}^{r} B_{kmn} a_n(t) a_m(t),
\]

where:

\[
b_k = -\left( \varphi_k, U \cdot \nabla U \right) - \frac{2}{Re} \left( \nabla \varphi_k, \frac{\nabla U + \nabla U^T}{2} \right),
\]

\[
A_{km} = -\left( \varphi_k, U \cdot \nabla \varphi_m \right) - \left( \varphi_k, \varphi_m \cdot \nabla U \right) - \frac{2}{Re} \left( \nabla \varphi_k, \frac{\nabla \varphi_m + \nabla \varphi_m^T}{2} \right),
\]

\[
B_{kmn} = -\left( \varphi_k, \varphi_m \cdot \nabla \varphi_n \right).
\]

- **POD-G-ROM** in matrix form:

\[
\dot{a} = b + Aa + a^T B a
\]

- **POD-G-ROM (8)** has been successfully used for *laminar flows*. 


Section 2: POD-G-ROM for Incompressible Flow

- **POD-G-ROM** algebraic system:

\[
\dot{a}_k(t) = b_k + \sum_{m=1}^{r} A_{km} a_m(t) + \sum_{m=1}^{r} \sum_{n=1}^{r} B_{kmn} a_n(t) a_m(t),
\]

where:

\[
b_k = -\left( \varphi_k, \mathbf{U} \cdot \nabla \mathbf{U} \right) - \frac{2}{Re} \left( \nabla \varphi_k, \frac{\nabla \mathbf{U} + \nabla \mathbf{U}^T}{2} \right),
\]

\[
A_{km} = -\left( \varphi_k, \mathbf{U} \cdot \nabla \varphi_m \right) - \left( \varphi_k, \varphi_m \cdot \nabla \mathbf{U} \right) - \frac{2}{Re} \left( \nabla \varphi_k, \frac{\nabla \varphi_m + \nabla \varphi_m^T}{2} \right),
\]

\[
B_{kmn} = -\left( \varphi_k, \varphi_m \cdot \nabla \varphi_n \right).
\]

- **POD-G-ROM** in matrix form:

\[
\dot{a} = b + Aa + a^T B a
\]

- POD-G-ROM (8) has been successfully used for **laminar flows**.

- For **structurally-dominated turbulent flows**, POD-G-ROM **fails**: effect of discarded modes \( \varphi_{r+1}, \ldots, \varphi_N \) need to be included in some way.
Section 2: POD-G-ROM for Incompressible Flow

- **POD-G-ROM** algebraic system:

\[
\dot{a}_k(t) = b_k + \sum_{m=1}^{r} A_{km}a_m(t) + \sum_{m=1}^{r} \sum_{n=1}^{r} B_{kmn}a_n(t)a_m(t),
\]

where:

\[
b_k = - (\varphi_k, U \cdot \nabla U) - \frac{2}{Re} \left( \nabla \varphi_k, \frac{\nabla U + \nabla U^T}{2} \right),
\]

\[
A_{km} = - (\varphi_k, U \cdot \nabla \varphi_m) - (\varphi_k, \varphi_m \cdot \nabla U) - \frac{2}{Re} \left( \nabla \varphi_k, \frac{\nabla \varphi_m + \nabla \varphi_m^T}{2} \right),
\]

\[
B_{kmn} = - (\varphi_k, \varphi_m \cdot \nabla \varphi_n).
\]

- **POD-G-ROM** in matrix form:

\[
\dot{a} = b + Aa + a^T B a
\]

- POD-G-ROM (8) has been successfully used for *laminar flows*.

- For *structurally-dominated turbulent flows*, POD-G-ROM fails: effect of discarded modes \( \varphi_{r+1}, ..., \varphi_N \) need to be included in some way.

“POD closure problem”
Outline

1. Motivation/Background
2. Section 2: POD/Galerkin ROMs for Incompressible Flows
3. Background on Turbulence Modeling/Large Eddy Simulation
4. Section 3: POD Closure Models
   • Mixing Length (ML)
   • Smagorinsky (S)
   • Variational Multi-Scale (VMS)
   • Dynamic Subgrid (DS)
5. Section 4.1: Computational Efficiency
6. Section 4.2: Numerical Results for 3D Flow Around Cylinder
7. Section 5 and Beyond: Future/Follow-Up Work
8. Basis Rotation (My Work – an Alternative Approach)
Turbulence

- For sufficiently high Reynolds number, flow becomes turbulent.
Turbulence

- For sufficiently high Reynolds number, flow becomes *turbulent*.

- Turbulence is *nonlinear, chaotic, 3D* phenomenon.
Turbulence

- For sufficiently high Reynolds number, flow becomes turbulent.

- Turbulence is nonlinear, chaotic, 3D phenomenon.

- **Kolmogorov hypothesis / energy cascade:**
  - Kinetic energy enters the turbulence through the production mechanism at largest scales of motion.
  - Energy is transferred (by inviscid processes) to smaller and smaller scales.
  - At smallest scales, energy is dissipated by viscous action.
Turbulence Modeling

- **Direct Numerical Simulation (DNS):** solves full Navier-Stokes (NS) equations (1) → requires fine meshes in boundary layer to resolve fine scales.
  - Too computationally expensive to be feasible for realistic complex flows.

- **Large Eddy Simulation (LES):** reduces computational cost of DNS by ignoring smallest length scales (most computationally expensive to resolve).
  - LES equations obtained by low-pass-filtering full NS equations.

- **Reynolds Averaged Navier-Stokes (RANS):** time-averaged versions of NS equations → turbulence is modeled, not resolved.
Large Eddy Simulation (LES)

- **Four conceptual steps of LES (Pope, Chapter 13):**
  
  (i) *Filtering operation* to decompose velocity into filtered (or resolved) component $\bar{u}(x, t)$ and residual (or subgrid-scale) component $u'(x, t)$.
  
  (ii) *Equations for evolution* of the filtered velocity are derived from the NS equations.
  
  (iii) *Closure* is obtained by modeling the residual-stress tensor (most simply with eddy-viscosity model).
  
  (iv) Model filtered equations are solved *numerically* for $\bar{u}(x, t)$, which provides approximation of large-scale motions in one realization of turbulent flow.

Left: energy cascade / “Kolmogorov spectrum” (energy transfer from large to small scales); LES filter filters out small scales.
LES Filtering

- General filtering operation defined by:

\[ \bar{u}(x, t) = \int G(r, x)u(x - r, t)dr \]

where \( G \) is a specified rapidly-decaying “filter function”, which has an associated “cut-off” length and time scale. Scales smaller than these cut-offs are eliminated using filter.
LES Filtering

- General filtering operation defined by:

\[
\bar{u}(x, t) = \int G(r, x)u(x - r, t)dr
\]

where \( G \) is a specified rapidly-decaying “filter function”, which has an associated “cut-off” length and time scale. Scales smaller than these cut-offs are eliminated using filter.

- Given a filter, any field can be split up into filtered \( \bar{u} \) and sub-filtered \( u' \) scale:

\[
u(x, t) = \bar{u}(x, t) + u'(x, t)
\]
LES Filtered Governing Equations

• Applying filtering operation to (1) gives the following equations for the filtered variables:

\[ \nabla \cdot \bar{u} = 0 \]

\[ \bar{u}_t - \frac{1}{Re} \Delta \bar{u} + \nabla \cdot (\bar{u}\bar{u}) + \nabla \cdot \tau + \nabla \bar{p} = 0 \]

where \( \tau \) is the \textit{subfilter-scale stress tensor}:

\[ \tau = \bar{u}\bar{u} - \bar{u}\bar{u} \]
LES Filtered Governing Equations

- Applying filtering operation to (1) gives the following equations for the filtered variables:

\[
\nabla \cdot \bar{u} = 0
\]
\[
\bar{u}_t - \frac{1}{Re} \Delta \bar{u} + \nabla \cdot (\bar{u}\bar{u}) + \nabla \cdot \tau + \nabla \bar{p} = 0
\]

where \( \tau \) is the subfilter-scale stress tensor:

\[
\tau = \bar{u}\bar{u} - \bar{u}\bar{u}
\]

- \( \bar{u}\bar{u} \) is most difficult term to model, as it requires knowledge of unfiltered velocity field, which is unknown.
LES Filtered Governing Equations

• Applying filtering operation to (1) gives the following equations for the **filtered variables**:

\[ \nabla \cdot \bar{u} = 0 \]
\[ \bar{u}_t - \frac{1}{Re} \Delta \bar{u} + \nabla \cdot (\bar{u}\bar{u}) + \nabla \cdot \tau + \nabla \bar{p} = 0 \]

where \( \tau \) is the **subfilter-scale stress tensor**:

\[ \tau = \bar{u}\bar{u} - \bar{u}\bar{u} \]

• \( \bar{u}\bar{u} \) is most difficult term to model, as it requires knowledge of unfiltered velocity field, which is unknown.

• Common approaches to model \( \bar{u}\bar{u} \): **eddy-viscosity (EV) models**

\[ \tau_{ij} - \frac{1}{3} \tau_{kk} \delta_{ij} = \nu_T \left( \frac{\partial \bar{u}_j}{\partial x_i} + \frac{\partial \bar{u}_i}{\partial x_j} \right) \]

where \( \nu_T \) is the eddy-viscosity.

• Expression for \( \nu_T \): “**eddy-viscosity ansatz**”

• **Examples**: mixing-length, Smagorinsky, etc. – parameters based on Kolmogorov spectrum.
Outline

1. Motivation/Background
2. Section 2: POD/Galerkin ROMs for Incompressible Flows
3. Background on Turbulence Modeling/Large Eddy Simulation
4. Section 3: POD Closure Models
   • Mixing Length (ML)
   • Smagorinsky (S)
   • Variational Multi-Scale (VMS)
   • Dynamic Subgrid (DS)
5. Section 4.1: Computational Efficiency
6. Section 4.2: Numerical Results for 3D Flow Around Cylinder
7. Section 5 and Beyond: Future/Follow-Up Work
8. Basis Rotation (My Work – an Alternative Approach)
Section 3.3: POD Closure Models

- POD-G-ROM (8) has been successfully used for *laminar flows*.
- For *structurally-dominated turbulent flows*, POD-G-ROM *fails*: effect of discarded modes $\phi_{r+1}, \ldots, \phi_N$ need to be included in some way.

“POD closure problem”
Section 3.3: POD Closure Models

- POD-G-ROM (8) has been successfully used for *laminar flows*.
- For *structurally-dominated turbulent flows*, POD-G-ROM *fails*: effect of discarded modes $\phi_{r+1}, \ldots, \phi_N$ need to be included in some way.

---

“POD closure problem”

- Natural way to tackle POD closure problem: “*eddy-viscosity*” (EV) turbulence modeling.
  - EV model states that role of discarded modes is to extract energy from system.
  - Concept of energy cascade has been confirmed numerically in POD setting.
Section 3.3: POD Closure Models

- POD-G-ROM (8) has been successfully used for *laminar flows*.
- For *structurally-dominated turbulent flows*, POD-G-ROM *fails*: effect of discarded modes $\varphi_{r+1}, \ldots, \varphi_N$ need to be included in some way.

**“POD closure problem”**

- Natural way to tackle POD closure problem: “*eddy-viscosity*” (EV) turbulence modeling.
  - EV model states that role of discarded modes is to extract energy from system.
  - Concept of energy cascade has been confirmed numerically in POD setting.
- **EV-POD-ROM** formulation:
  \[
  \dot{a} = (b + \tilde{b}(a)) + (A + \tilde{A}(a))a + a^T Ba \tag{24}
  \]
  where $\tilde{b}(a)$ and $\tilde{A}(a)$ correspond to numerical discretization of EV closure model.
Section 3.3: POD Closure Models

- POD-G-ROM (8) has been successfully used for *laminar flows*.
- For *structurally-dominated turbulent flows*, POD-G-ROM fails: effect of discarded modes $\phi_{r+1}, \ldots, \phi_N$ need to be included in some way.

  “POD closure problem”

- Natural way to tackle POD closure problem: “*eddy-viscosity*” (EV) turbulence modeling.
  - EV model states that role of discarded modes is to extract energy from system.
  - Concept of energy cascade has been confirmed numerically in POD setting.
- EV-POD-ROM formulation:

  $\dot{a} = (b + \tilde{b}(a)) + (A + \tilde{A}(a))a + a^T B a$  \hspace{1cm} (24)

  where $\tilde{b}(a)$ and $\tilde{A}(a)$ correspond to numerical discretization of EV closure model.

(24) is equivalent to adding $\nu_T \left( \frac{\partial \bar{u}_j}{\partial x_i} + \frac{\partial \bar{u}_i}{\partial x_j} \right)$ to equations (1) and projecting.
Section 3.3: POD Closure Models

- POD-G-ROM (8) has been successfully used for laminar flows.
- For structurally-dominated turbulent flows, POD-G-ROM fails: effect of discarded modes \( \varphi_{r+1}, \ldots, \varphi_N \) need to be included in some way.
  
  "POD closure problem"

- Natural way to tackle POD closure problem: “eddy-viscosity” (EV) turbulence modeling.
  - EV model states that role of discarded modes is to extract energy from system.
  - Concept of energy cascade has been confirmed numerically in POD setting.

- EV-POD-ROM formulation:

\[
\dot{a} = (b + \tilde{b}(a)) + (A + \tilde{A}(a))a + a^T Ba
\]  

(24)

where \( \tilde{b}(a) \) and \( \tilde{A}(a) \) correspond to numerical discretization of EV closure model.

(24) is equivalent to adding \( \nu_T \left( \frac{\partial \tilde{u}_j}{\partial x_i} + \frac{\partial \tilde{u}_i}{\partial x_j} \right) \) to equations (1) and projecting.

- Four EV-POD-ROMs of the form (24) proposed/evaluated:
  
  1. **ML-POD-ROM** (ML = mixing length).
  2. **S-POD-ROM** (S = Smagorinsky).
  3. **VMS-POD-ROM** (VMS = variational multi-scale).
  4. **DS-POD-ROM** (DS = dynamic subgrid).
• **POD/Galerkin projection filter** (Section 3.1):
  - In POD, there is no explicit spatial filter used ⇒ to develop LES-type POD closure models, a POD filter needs to be introduced.
  - **Natural filter is Galerkin projection**: for all $u \in \mathcal{H}$, the Galerkin projection $\bar{u} \in X^r$ is the solution to the following equation:
    
    $$
    (u - \bar{u}, \varphi) = 0 \quad \forall \varphi \in X^r.
    $$

*By doing POD/Galerkin projection to build the ROM, one is applying a filter.*

In the context of LES, filtered equations require introduction of closure model to model effect of neglected POD modes. This is where idea of adding EV models to ROM equations comes from.
POD Filter and Lengthscale (Section 3.1-3.2)

• **POD/Galerkin projection filter** (Section 3.1):
  
  • In POD, there is no explicit spatial filter used ⇒ to develop LES-type POD closure models, a POD filter needs to be introduced.
  
  • **Natural filter is Galerkin projection**: for all \( u \in \mathcal{H} \), the Galerkin projection \( \bar{u} \in X^r \) is the solution to the following equation:

  \[
  (u - \bar{u}, \varphi) = 0 \quad \forall \varphi \in X^r. \tag{14}
  \]

  By doing POD/Galerkin projection to build the ROM, one is applying a filter.

  In the context of LES, filtered equations require introduction of closure model to model effect of neglected POD modes. This is where idea of adding EV models to ROM equations comes from.

• **POD lengthscale** (Section 3.2): implicitly defined by neglected modes \( \{\varphi_j\}_{j=r+1}^N \)

  \[
  \delta := \left( \frac{1}{L_1 L_2} \int_0^{L_1} \int_0^{L_2} \frac{\langle u_{i>}, u_{i>} \rangle}{\langle u_{i>j}, u_{i>j} \rangle} dx_1 dx_2 \right)^{1/2}. \tag{23}
  \]

  where \( u_{i>} = \sum_{j=r+1}^N a_j^i \varphi_j, \langle \cdot \rangle = \text{spatial average in homogeneous direction}, L_1, L_2 \) are streamwise and spanwise dimensions of computational domain.
Section 3.3.1: ML*-POD-ROM

**Mixing length model**: \( \nu_T = \nu_{ML} = \alpha U_{ML} L_{ML} \).

- \( U_{ML} \) = characteristic velocity scale (estimated using dimensional analysis; Sec. 3.2).
- \( L_{ML} \) = characteristic length scale (estimated using dimensional analysis; Sec. 3.2).
- \( \alpha = O(1) \) non-dimensional parameter that characterized energy being dissipated.

**ML-POD-ROM** is of form (24) with:

\[
\tilde{b}_k(a) = -\nu_{ML} \left( \nabla \phi_k, \frac{\nabla U + \nabla U^T}{2} \right), \quad (27)
\]

\[
\tilde{A}_{km}(a) = -\nu_{ML} \left( \nabla \phi_k, \frac{\nabla \phi_m + \nabla \phi_m^T}{2} \right). \quad (28)
\]

**Remarks**:  
- Different values of \( \alpha \) may result in different dynamics (\( \alpha \) varies in real turbulent flow).  
- \( \nu_{ML} \) typically computed once at beginning of simulation.  
- Improvements to ML-POD-ROM where \( \nu_{ML} \) are mode dependent have been proposed in [30, 32, 58].
Section 3.3.2: S*-POD-ROM

- **Smagorinsky model:** $\nu_T = \nu_S = (C_S \delta)^2 \|\mathbb{D}(u_r)\|_F$.
  - $C_S =$ Smagorinsky constant.
  - $\delta =$ length scale (estimated using dimensional analysis; Sec. 3.2).
  - $\|\mathbb{D}(u_r)\|_F =$ Frobenius norm of deformation tensor.
  - $\nu_S =$ “EV ansatz”.

- **S-POD-ROM** is of form (24) with:

\[
\begin{align*}
\tilde{b}_k(a) &= -2(C_S \delta)^2 \left( \nabla \varphi_k, \|\mathbb{D}(u_r)\| \frac{\nabla U + \nabla U^T}{2} \right), \\
\tilde{A}_{km}(a) &= -2(C_S \delta)^2 \left( \nabla \varphi_k, \|\mathbb{D}(u_r)\| \frac{\nabla \varphi_m + \nabla \varphi_m^T}{2} \right).
\end{align*}
\] (30) (31)

- **Remarks:**
  - Main advantage over ML-POD-ROM: EV coefficient recomputed at every time-step.
  - EV terms are nonlinear and need to be handled efficiently – discussed in Section 4.

* S = Smagorinsky
Section 3.3.3: VMS*-POD-ROM

- **VMS LES**: based on principle of locality of energy transfer (energy is transferred mainly between neighboring scales) – shown to be valid in POD context [43].

- Decompose space of POD modes into 2 spaces, one of “large” and one of “small” scale modes: $X^r = X^r_L \oplus X^r_S$ where:

  $$X^r_L := \text{span}\{\varphi_1, \varphi_2, \ldots, \varphi_{r_L}\} \quad X^r_S := \text{span}\{\varphi_{r_L+1}, \varphi_{r_L+2}, \ldots, \varphi_r\}.$$

- Decompose ROM solution into “large resolved” and “small resolved” scales: $u^r = u^L_r + u^S_r$ where:

  $$u^L_r = U + \sum_{j=1}^{r_L} a_j \varphi_j, \quad u^S_r = \sum_{j=r_L+1}^{r} a_j \varphi_j.$$

- **VMS-POD-ROM** has the form:

  $$\begin{bmatrix} \dot{a}^L \\ \dot{a}^S \end{bmatrix} = \begin{bmatrix} b^L \\ b^S \end{bmatrix} + A^r \begin{bmatrix} a^L \\ a^S \end{bmatrix} + \begin{bmatrix} A^L \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ A^S + \tilde{A}^S(a^S) \end{bmatrix} \begin{bmatrix} a^L \\ a^S \end{bmatrix} + \begin{bmatrix} a^L \end{bmatrix}^T B \begin{bmatrix} a^L \\ a^S \end{bmatrix}.$$  

  (38)

* VMS = Variational Multi-Scale
Section 3.3.3: VMS-POD-ROM

- **VMS-POD-ROM** has the form:

\[
\begin{bmatrix}
\dot{a}_L^L \\
\dot{a}_L^S \\
\dot{a}_S^L \\
\dot{a}_S^S
\end{bmatrix} =
\begin{bmatrix}
b_L^L \\
b_L^S \\
b_S^L \\
b_S^S
\end{bmatrix} +
A^r
\begin{bmatrix}
a_L^L \\
a_L^S \\
a_S^L \\
a_S^S
\end{bmatrix} +
\begin{bmatrix}
A_L^L \\
0 \\
A_S^L + \tilde{A}_S^S(a_S^S) \\
0
\end{bmatrix}
\begin{bmatrix}
a_L^L \\
a_L^S \\
a_S^L \\
a_S^S
\end{bmatrix} +
\begin{bmatrix}
a_L^L \\
a_L^S \\
a_S^L \\
a_S^S
\end{bmatrix}^T
B
\begin{bmatrix}
a_L^L \\
a_L^S \\
a_S^L \\
a_S^S
\end{bmatrix}.
\tag{38}
\]

- **Remarks:**
  - EV term applied to small scales only, following principle of energy transfer locality:

\[
\tilde{A}_k^S(a) = -2(C_s\delta)^2 \left( \nabla \phi_k, \|D(u_r^S + U)\| \frac{\nabla \phi_j + \nabla \phi_j^T}{2} \right).
\]

- Unlike S-POD-ROM, VMS-POD-ROM acts only on small resolved scales, whereas in S-POD-ROM, it acts on all (small and large) resolved scales.

- (38) is coupled through two terms:
  - (i) \(a^T B a\): represents nonlinear convective term \((u^T \cdot \nabla)u^r\)
  - (ii) \(A^T a\): represents nonlinear term \((u^r \cdot \nabla)u^r\) linearized around base flow \(U\).

- EV terms are nonlinear and need to be handled efficiently – discussed in Section 4.
Section 3.3.4 : DS*-POD-ROM

- **Dynamic subgrid model**: \( \nu_T = \nu_{DS} = (C_S(x, t)\delta)^2||D(u_r)||_F \).
- **DS-POD-ROM** is of form (24) with:

\[
\begin{align*}
\tilde{b}_k(a) &= -2(C_S\delta)^2 \left( \nabla \varphi_k, \|D(u_r)\| \frac{\nabla U + \nabla U^T}{2} \right), \quad (30) \\
\tilde{A}_{km}(a) &= -2(C_S\delta)^2 \left( \nabla \varphi_k, \|D(u_r)\| \frac{\nabla \varphi_m + \nabla \varphi_m^T}{2} \right). \quad (31)
\end{align*}
\]

- **Least-squares problem** for \( C_S(x, t) \) is obtained by applying filtering twice to ROM equations, assuming \( C_S(x, t) \) is constant under double filtering, and equating terms.

\[
C_S^2(x, t) \Rightarrow \frac{[\tilde{u}_r\tilde{u}_r - \tilde{u}_r\tilde{u}_r]}{\left[ 2\delta^2 \|D(\tilde{u}_r)\|D(\tilde{u}_r) - 2\tilde{\delta}^2 \|D(\tilde{u}_r)\|D(\tilde{u}_r) \right]} : \left[ 2\delta^2 \|D(\tilde{u}_r)\|D(\tilde{u}_r) - 2\tilde{\delta}^2 \|D(\tilde{u}_r)\|D(\tilde{u}_r) \right].
\]

- **Remarks**:
  - \( \nu_{DS} \) can take on negative values – can be interpreted as **backscatter** (inverse transfer of energy from high index POD modes to low index modes).
  - Notion of backscatter is well-established in LES.
  - EV terms are nonlinear and need to be handled efficiently – discussed in Section 4.

* DS = Dynamic Subgrid
Outline

1. Motivation/Background
2. Section 2: POD/Galerkin ROMs for Incompressible Flows
3. Background on Turbulence Modeling/Large Eddy Simulation
4. Section 3: POD Closure Models
   • Mixing Length (ML)
   • Smagorinsky (S)
   • Variational Multi-Scale (VMS)
   • Dynamic Subgrid (DS)
5. Section 4.1: Computational Efficiency
6. Section 4.2: Numerical Results for 3D Flow Around Cylinder
7. Section 5 and Beyond: Future/Follow-Up Work
[8. Basis Rotation (My Work – an Alternative Approach)]
Ensuring Computational Efficiency (Section 4.1)

- All EV-POD-ROMs have **nonlinear closure model** terms except ML-POD-ROMs.
- Two approaches used to ensure **computational efficiency**:
  1. Instead of updating closure terms in ROMs at every time step, re-compute them every 1.5 time units (every 20K time steps for numerical example considered).
  2. Two-level algorithm: create 2 meshes (coarse and fine); discretize/compute closure terms $\tilde{b}(a^l), \tilde{A}(a^l)$ on coarse mesh.

\[ M = \text{total number of time steps.} \]

- Both (i) and (ii) were applied to all ROMs for fair comparison (even ML-POD-ROM).
- In [50], it was shown that two-level algorithm achieves same level of accuracy as one-level algorithm while decreasing computational cost by order of magnitude.
Outline

1. Motivation/Background
2. Section 2: POD/Galerkin ROMs for Incompressible Flows
3. Background on Turbulence Modeling/Large Eddy Simulation
4. Section 3: POD Closure Models
   • Mixing Length (ML)
   • Smagorinsky (S)
   • Variational Multi-Scale (VMS)
   • Dynamic Subgrid (DS)
5. Section 4.1: Computational Efficiency
6. Section 4.2: Numerical Results for 3D Flow Around Cylinder
7. Section 5 and Beyond: Future/Follow-Up Work
[8. Basis Rotation (My Work – an Alternative Approach)]
Numerical Example: 3D Flow Past Re = 1000 Cylinder (Section 4.1)

- **FOM** = finite difference DNS solver + 2\textsuperscript{nd} order Crank-Nicolson & Adams-Bashforth time-integration scheme (\(dt = 7.5\times10^{-4}\)).
- 1000 snapshots collected of velocity field over time \([0,75]\); ROM run for time \([0,300]\).
- **POD basis** of size \(r = 6\) created using snapshots; modes capture 84\% of snapshot energy.
- POD ROMs created using continuous projection and P2 finite elements + forward Euler time-integration scheme (\(dt = 2\times10^{-3}\)).
- Projection of quadratic nonlinearities in incompressible NS equations pre-computed.
- **Closure model terms** computed using (i) and (ii) on previous slide; coarse mesh for (ii) had 37x49x17 grid points (coursing factor \(R_c = 4\) in both radial and azimuthal directions).
Numerical Example: Estimation of Parameters (Section 4.1)

- Length scales in EV-POD-ROMs estimated using *dimensional analysis* (Section 3.2).

- “Correct” values for *EV constants* $\alpha$ in ML-POD-ROM and $C_S$ in S-POD-ROM and VMS-POD-ROM are not known in POD context. These parameters are estimated as follows:
  - Run POD-ROM on short time interval $[0,15]$ with several different values of EV constants.
  - Choose value that gives closest result to DNS.
  - **Note:** these EV constants are optimal only on short time interval tested – might be non-optimal for (longer) time-interval where ROM is run.

- For *VMS-POD-ROM*, only first mode considered large resolved scale, so $r_L = 1$.

- For *DS-POD-ROM*, since $\nu_{DS}$ can be negative, a standard “clipping” procedure is used to ensure numerical stability of discretization: let $C_S(x,t) = \max\{C_S(x,t), -0.2\}$, where -0.2 is determined numerically (see paper for details).
Numerical Example: Estimation of Parameters (Section 4.1)

- Length scales in EV-POD-ROMs estimated using *dimensional analysis* (Section 3.2).

- “Correct” values for *EV constants* $\alpha$ in ML-POD-ROM and $C_S$ in S-POD-ROM and VMS-POD-ROM are not known in POD context. These parameters are estimated as follows:
  - Run POD-ROM on short time interval $[0,15]$ with several different values of EV constants.
  - Choose value that gives closest result to DNS.
  - *Note:* these EV constants are optimal only on short time interval tested – might be non-optimal for (longer) time-interval where ROM is run.

- For **VMS-POD-ROM**, only first mode considered large resolved scale, so $r_L = 1$.

- For **DS-POD-ROM**, since $\nu_{DS}$ can be negative, a standard “clipping” procedure is used to ensure numerical stability of discretization: let $C_S(x,t) = \max\{C_S(x,t), -0.2\}$, where -0.2 is determined numerically (see paper for details).

*Comment:* authors consider one basis size $r$. Approach could be modified so $\nu_T \to 0$ as $r \to K$ for consistency.
Criteria for Evaluating ROMs (Section 4.1)

• *Five criteria* are used for evaluating ROMs:

  (i) Kinetic energy spectrum.

  (ii) Mean velocity.

  (iii) Reynolds stresses.

  (iv) Root mean square (rms) values of velocity and fluctuations.

  (v) Time-evolution of POD coefficients.
Criteria for Evaluating ROMs (Section 4.1)

• **Five criteria** are used for evaluating ROMs:
  
  (i) Kinetic energy spectrum.
  
  (ii) Mean velocity.
  
  (iii) Reynolds stresses.
  
  (iv) Root mean square (rms) values of velocity and fluctuations.
  
  (v) Time-evolution of POD coefficients.

*Comment:* These are similar criteria to those we are interested in!
Kinetic Energy Spectrum (Section 4.1)

• All energy spectra calculated from average kinetic energy (KE) at a single point.

• POD-G-ROM: over-estimates energy spectrum.

• ML-POD-ROM: underestimates energy spectrum, especially at higher frequencies.

• S-POD-ROM: more accurate than ML-POD-ROM, but displays high oscillations at higher frequencies.

• VMS-POD-ROM: improvement over S-POD-ROM, with smaller oscillations at higher frequencies.

DS-POD-ROM and VMS-POD-ROM yield most accurate energy spectra, with DS-POD-ROM slightly better than VMS-POD-ROM.
Mean Velocity Components (Section 4.1)

- \( \langle u \rangle = \) mean streamwise velocity
- \( \langle v \rangle = \) mean normal velocity
- \( \langle w \rangle = \) mean spanwise velocity
- \( \langle \cdot \rangle \) denotes time-averaging for \( t=[0,300] \) and spatial averaging performed in \( yz \)-direction.

- Mean streamwise velocity computed accurately by all POD-ROMs.
- POD-G-ROM yields inaccurate results for mean normal velocity; all other POD-ROMs performed significantly better.

Mean spanwise velocity results similar for all EV-ROMs; POD-G-ROM performed better than all EV-ROMs over certain regions worse over others.
Reynolds Stresses (Section 4.1)

- \( \langle u - \langle u \rangle, v - \langle v \rangle \rangle \): \( xy \)-component of Reynolds stress.

- \( \langle u - \langle u \rangle, w - \langle w \rangle \rangle \): \( xz \)-component of Reynolds stress.

- \( \langle v - \langle v \rangle, w - \langle w \rangle \rangle \): \( yz \)-component of Reynolds stress.

- POD-G-ROM Reynolds stresses are consistently most inaccurate.

EV-ROMs have similar behaviors; no clear “winner”.

(a)

(b)

(c)
Root Mean Square Values (Section 4.1)

- $\langle u \rangle_{rms} = \langle u - \langle u \rangle, u - \langle u \rangle \rangle$: rms of streamwise velocity fluctuations.
- $\langle v \rangle_{rms} = \langle v - \langle v \rangle, v - \langle v \rangle \rangle$: rms of normal velocity fluctuations.
- $\langle w \rangle_{rms} = \langle w - \langle w \rangle, w - \langle w \rangle \rangle$: rms of spanwise velocity fluctuations.
- POD-G-ROM rms values of velocity fluctuations are consistently the most inaccurate.
- DS-POD-ROM and VMS-POD-ROMs consistently outperformed other two EV-ROMs.

S-POD-ROM consistently performs worse than DS-POD-ROM and VMS-POD-ROM, but is clearly more accurate than ML-POD-ROM.
Time Evolution of POD Coefficients (Section 4.1)

- Left: $a_1$, right: $a_4$.
- POD-G-ROM’s time evolutions of $a_1$ and $a_4$ are clearly inaccurate.
- ML-POD-ROMs time evolutions also inaccurate.
- S-POD-ROM more accurate than ML-POD-ROM.
- VMS-POD-ROM more accurate than S-POD-ROM.
- DS-POD-ROM yields accurate results.
- More variability in coefficients for DS-POD-ROM due to $C_S$ varying with space and time.

VMS-POD-ROM and DS-POD-ROM perform best.
CPU Times (Section 4.1)

• To measure computational efficiency of the four POD-ROMs, define *speed-up factor*:

\[
S_f = \frac{\text{CPU time of DNS}}{\text{CPU time of POD-ROM}}
\]  

(68)

• Table 1 gives speed-up factors for POD-ROMs:

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Speed-up factors of POD-ROMs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>POD-G-ROM</td>
<td>ML-POD-ROM</td>
</tr>
<tr>
<td>(S_f)</td>
<td>665</td>
</tr>
</tbody>
</table>

• POD-G-ROM is most **efficient**; as sophistication of turbulence model increases, model becomes **more expensive**, not surprisingly.
CPU Times (Section 4.1)

- To measure computational efficiency of the four POD-ROMs, define **speed-up factor**:

\[ S_f = \frac{\text{CPU time of DNS}}{\text{CPU time of POD-ROM}} \]  

- Table 1 gives speed-up factors for POD-ROMs:

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Speed-up factors of POD-ROMs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>POD-G-ROM</td>
<td>665</td>
</tr>
<tr>
<td>ML-POD-ROM</td>
<td>659</td>
</tr>
<tr>
<td>S-POD-ROM</td>
<td>36</td>
</tr>
<tr>
<td>VMS-POD-ROM</td>
<td>41</td>
</tr>
<tr>
<td>DS-POD-ROM</td>
<td>23</td>
</tr>
</tbody>
</table>

- POD-G-ROM is most **efficient**; as sophistication of turbulence model increases, model becomes **more expensive**, not surprisingly.

**Comment:** it may be possible to improve these speed-up factors using hyper-reduction to handle non-linear terms (e.g., DEIM, gappy POD).
Summary of Results (Section 4.1)

• **VMS** and **DS** approaches yield **most accurate** POD closure models (i.e., give most accurate average and instantaneous numerical results):

  • Best energy spectra.
  • Best rms values.
  • Best time evolution of POD coefficients $a_1$ and $a_4$.
  • With respect to other criteria (mean velocity components, Reynolds stresses), DS-POD-ROM and VMS-POD-ROM perform at least as well as other POD-ROMs.
Summary of Results (Section 4.1)

• **VMS** and **DS** approaches yield **most accurate** POD closure models (i.e., give most accurate average and instantaneous numerical results):
  
  • Best energy spectra.
  • Best rms values.
  • Best time evolution of POD coefficients $a_1$ and $a_4$.
  • With respect to other criteria (mean velocity components, Reynolds stresses), DS-POD-ROM and VMS-POD-ROM perform at least as well as other POD-ROMs.

**Comment:** one cannot make these definitive conclusions from data presented because all the ROMs do not have the same computational cost (see previous slide)... but it is possible to improve computational efficiency of EV-ROMs using hyper-reduction like DEIM or gappy POD.
Outline

1. Motivation/Background
2. Section 2: POD/Galerkin ROMs for Incompressible Flows
3. Background on Turbulence Modeling/Large Eddy Simulation
4. Section 3: POD Closure Models
   • Mixing Length (ML)
   • Smagorinsky (S)
   • Variational Multi-Scale (VMS)
   • Dynamic Subgrid (DS)
5. Section 4.1: Computational Efficiency
6. Section 4.2: Numerical Results for 3D Flow Around Cylinder
7. Section 5 and Beyond: Future/Follow-Up Work
[8. Basis Rotation (My Work – an Alternative Approach)]
Future Work (Section 5)

- Study **more efficient time-discretization approaches** and take advantage of parallel computing in POD-ROMs.

- Investigate **hybrid approach**: using DS-POD-ROM to calculate $\alpha$ only when flow displays high level of variability, use this value in ML-POD-ROM (linear) as long as flow does not experience sudden transitions.

- **Higher Reynolds number** structurally dominated turbulent flows.

- Combining two-level algorithms in conjunction with **EIM** and **DEIM** to handle efficiently nonlinearities in turbulence models.

- Application to problems in **optimal control, optimization, data assimilation**.
Follow Up Work by Iliescu et al.

- **Filtered ROMs (F-ROMs)**
  - ROM terms are explicitly filtered using projection or differential filter.
  - E.g., “Evolve-then-filter” approach: do one step of ROM, filter ROM amplitudes $a_i(t)$, repeat.

- **Calibrated ROMs (C-ROMs)**
  - Turbulence model terms, e.g., $\tilde{A}$, are obtained by solving optimization problem.

$$
\min_{\tilde{A}} \sum_{j=1}^{M} ||a(t_j) - a^{snap}(t_j)||^2
$$

Unlike LES-ROMs, F-ROMs and C-ROMs are **consistent** and more **computationally efficient** (no nonlinear turbulence model terms to compute).

- **References:**
  - Technical seminar by X. Xie in March 2017: “LES ROMs”.
Other Follow Up Work

• **Closure models with auto-tuned data-driven coefficients:**
  • Free parameters in closure models are “learned” online using data-driven multi-parameter extremum seeking (MES) algorithm.
  • Takes into account parametric uncertainties
  • Employs robust Lyapunov control theory.

Outline

1. Motivation/Background
2. Section 2: POD/Galerkin ROMs for Incompressible Flows
3. Background on Turbulence Modeling/Large Eddy Simulation
4. Section 3: POD Closure Models
   • Mixing Length (ML)
   • Smagorinsky (S)
   • Variational Multi-Scale (VMS)
   • Dynamic Subgrid (DS)
5. Section 4.1: Computational Efficiency
6. Section 4.2: Numerical Results for 3D Flow Around Cylinder
7. Section 5 and Beyond: Future/Follow-Up Work
[8. Basis Rotation (My Work – an Alternative Approach)]
We start with the 3D compressible Navier-Stokes equations in *primitive specific volume form*:

\[
\begin{align*}
\zeta, t + \zeta, j u_j - \zeta u_{j,j} &= 0 \\
u_{i,t} + u_{i,j} u_j + \zeta p_i - \frac{1}{Re} \zeta \tau_{i,j} &= 0 \\
p, t + u_j p, j + \gamma u_{j,j} p - \left( \frac{\gamma}{PrRe} \right) \kappa(p \zeta), j - \left( \frac{\gamma - 1}{Re} \right) u_{i,j} \tau_{ij} &= 0
\end{align*}
\]
3D Compressible Navier-Stokes Equations

- We start with the 3D compressible Navier-Stokes equations in *primitive specific volume form*:

\[
\begin{align*}
\zeta_t + \zeta_j u_j - \zeta u_{j,j} &= 0 \\
u_{i,t} + u_{i,j} u_j + \zeta p_i - \frac{1}{Re} \zeta \tau_{ij,j} &= 0 \\
p_t + u_j p_j + \gamma u_{j,j} p - \left(\frac{\gamma}{PrRe}\right)(\kappa(p\zeta),j)_j - \left(\frac{\gamma - 1}{Re}\right) u_{i,j} \tau_{ij} &= 0
\end{align*}
\]

\[\text{[PDEs]}\]  

- Spectral discretization \((q(x, t) \approx \sum_{i=1}^{n} a_i(t) U_i(x)) + \text{Galerkin projection applied to (1) yields a system of } n \text{ coupled quadratic ODEs:}\]

\[
\frac{d\mathbf{a}}{dt} = \mathbf{C} + \mathbf{L} \mathbf{a} + [\mathbf{a}^T \mathbf{Q}^{(1)} \mathbf{a} + \mathbf{a}^T \mathbf{Q}^{(2)} \mathbf{a} + \cdots + \mathbf{a}^T \mathbf{Q}^{(n)} \mathbf{a}]^T
\]

\[\text{[ROM]}\]

where \(\mathbf{C} \in \mathbb{R}^n, \mathbf{L} \in \mathbb{R}^{n \times n} \text{ and } \mathbf{Q}^{(i)} \in \mathbb{R}^{n \times n} \text{ for all } i = 1, \ldots, n.\]
ROM Instability Problem

Stability can be a real problem for compressible flow ROMs!

- A compressible fluid POD/Galerkin ROM might be stable for a given number of modes, but **unstable** for other choices of basis size (Bui-Tanh *et al.* 2007).

- Some* remedies:

<table>
<thead>
<tr>
<th>Change projection (a priori)</th>
<th>Continuous Projection</th>
<th>Discrete Projection</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Change ROM equations (a posteriori)</th>
<th>Continuous Projection</th>
<th>Discrete Projection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear/nonlinear turbulence modeling (Iliescu, Borggaard, Xie, Wang, ...)</td>
<td>Eigenvalue reassignment (IKT <em>et al.</em>)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change ROM basis (a posteriori)</th>
<th>Continuous Projection</th>
<th>Discrete Projection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basis rotation (Balajewicz, IKT, <em>et al.</em>)</td>
<td>Optimization-based right basis modification (Amsallem <em>et al.</em>)</td>
<td></td>
</tr>
</tbody>
</table>
ROM Instability Problem

Stability can be a real problem for compressible flow ROMs!

- A compressible fluid POD/Galerkin ROM might be stable for a given number of modes, but **unstable** for other choices of basis size (Bui-Tanh *et al.* 2007).
- Some* remedies:

<table>
<thead>
<tr>
<th>Remedies</th>
<th>Continuous Projection</th>
<th>Discrete Projection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change ROM equations <em>(a posteriori)</em></td>
<td>Linear/nonlinear turbulence modeling (Iliescu, Borggaard, Xie, Wang, ...)</td>
<td>Eigenvalue reassignment (<strong>IKT</strong> <em>et al.</em>)</td>
</tr>
<tr>
<td>Change ROM basis <em>(a posteriori)</em></td>
<td>Basis rotation (<strong>Balajewicz</strong>, <strong>IKT</strong>, <em>et al.</em>)</td>
<td>Optimization-based right basis modification (Amsallem <em>et al.</em>)</td>
</tr>
</tbody>
</table>

- **Yellow box:** Wang, Akhtar, Borggaard, Iliescu.
- **Blue box:** Balajewicz, Tezaur, Dowell.
ROM Instability Problem

Stability can be a real problem for compressible flow ROMs!

- A compressible fluid POD/Galerkin ROM might be stable for a given number of modes, but **unstable** for other choices of basis size (Bui-Tanh et al. 2007).

- Some* remedies:

<table>
<thead>
<tr>
<th>Change projection (a priori)</th>
<th>Continuous Projection</th>
<th>Discrete Projection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy inner products (Rowley et al., Serre et al., IKT et al.)</td>
<td>Energy inner products (Rowley, et al.), Petrov-Galerkin Projection (Carlberg et al.)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change ROM equations (a posteriori)</th>
<th>Continuous Projection</th>
<th>Discrete Projection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear/nonlinear turbulence modeling (Iliescu, Borggaard, Xie, Wang, ...)</td>
<td></td>
<td>Eigenvalue reassignment (IKT et al.)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Change ROM basis (a posteriori)</th>
<th>Continuous Projection</th>
<th>Discrete Projection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basis rotation (Balajewicz, IKT, et al.)</td>
<td>Optimization-based right basis modification (Amsallem et al.)</td>
<td></td>
</tr>
</tbody>
</table>

- **Yellow box:** Wang, Akhtar, Borggaard, Iliescu.
- **Blue box:** Balajewicz, Tezaur, Dowell.  
  
Aimed at remedying **mode truncation instability.**
Mode Truncation Instability

- Projection-based MOR necessitates **truncation**.
- POD is, by definition and design, biased towards the **large, energy producing** scales of the flow (i.e., modes with large POD eigenvalues).
- Truncated/unresolved modes are negligible from a **data compression** point of view (i.e., small POD eigenvalues) but are crucial for the **dynamical equations**.
- For fluid flow applications, higher-order modes are associated with energy **dissipation**
  ⇒ low-dimensional ROMs (Galerkin *and* Petrov-Galerkin) can be **inaccurate** and **unstable**.

For a low-dimensional ROM to be stable and accurate, the **truncated/unresolved subspace** must be accounted for.

**Turbulence Modeling** (traditional approach)

**Subspace Rotation** (our approach)
Proposed new approach*: basis rotation

Instead of modeling truncation via additional linear term, model the truncation \textit{a priori} by “rotating” the projection subspace into a more dissipative regime.
Proposed new approach*: basis rotation

Instead of modeling truncation via additional linear term, model the truncation \textit{a priori} by “rotating” the projection subspace into a more dissipative regime.

### Illustrative example

- **Standard approach**: retain only the most energetic POD modes, i.e., \(U_1, U_2, U_3\).
- **Proposed approach**: add some higher order basis modes to increase dissipation, i.e., \(a_1U_1 + b_1U_6 + c_1U_8, a_2U_2 + b_2U_{11} + c_2U_{18}, a_3U_3 + b_3U_{21} + c_3U_{28}\).

---

*M. Balajewicz, E. Dowell. Stabilization of projection-based reduced order models of the Navier-Stokes equation. *Nonlinear Dynamics* 70 (2), 1619-1632 (2012).*
Proposed new approach*: basis rotation

Instead of modeling truncation via additional linear term, model the truncation \textit{a priori} by “rotating” the projection subspace into a more dissipative regime.

**Illustrative example**

- **Standard approach**: retain only the most energetic POD modes, i.e., $U_1, U_2, U_3$.
- **Proposed approach**: add some higher order basis modes to increase dissipation, i.e., \[ a_1 U_1 + b_1 U_6 + c_1 U_8, a_2 U_2 + b_2 U_{11} + c_2 U_{18}, a_3 U_3 + b_3 U_{21} + c_3 U_{28} \]

- **More generally**: approximate the solution using a linear superposition of $n + p$ (with $p > 0$) most energetic modes:

\[
\widetilde{U}_i = \sum_{j=1}^{n+p} X_{ij} U_j, \quad i = 1, \ldots, n, \quad (3)
\]

where $X \in \mathbb{R}^{(n+p) \times n}$ is an orthonormal ($X^T X = I_{n \times n}$) “rotation” matrix.

Goals of proposed new approach to account for modal truncation

Find $X$ such that:

1. New modes remain good approximations of the flow → minimize the "rotation" angle, i.e., minimize $X - I_M + P_M F$,

2. New modes produce stable and accurate ROMs. → ensure appropriate balance between energy production and energy dissipation.
Goals of proposed new approach to account for modal truncation

Find $X$ such that:

1. New modes $\tilde{\Phi}$ remain *good approximations* of the flow
   → minimize the “rotation” angle, i.e., minimize $\|X - I_{(M+P),M}\|_F$
Goals of proposed new approach to account for modal truncation

Find $X$ such that:

1. New modes $\tilde{\phi}$ remain *good approximations* of the flow
   \[ \rightarrow \text{minimize the “rotation” angle, i.e., minimize } \|X - I_{(M+P),M}\|_F \]

2. New modes produce *stable* and *accurate* ROMs.
   \[ \rightarrow \text{ensure appropriate balance between energy production and energy dissipation.} \]
Goals of proposed new approach to account for modal truncation

Find $X$ such that:

1. New modes $\tilde{\phi}$ remain *good approximations* of the flow
   → minimize the “rotation” angle, i.e., minimize $\|X - I_{(M+P),M}\|_F$

2. New modes produce *stable* and *accurate* ROMs.
   → ensure appropriate balance between energy production and energy dissipation.

• Once $X$ is found, the result is a system of the form (3) with:

$$Q^{(i)}_{jk} \leftarrow \sum_{s,q,r=1}^{M+P} X_{si} Q^{(s)}_{qr} X_{qr} X_{rk}, \quad L \leftarrow X^T LX, \quad C \leftarrow X^T C^*$$
Minimal subspace rotation

- Trace minimization problem on the Stiefel manifold:

\[
\begin{align*}
\text{minimize } & \min_{X \in \mathcal{V}_{(M+P),M}} - \text{tr}(X^T I_{(M+P) \times M}) \\
\text{subject to } & \text{tr}(X^T LX) = \eta
\end{align*}
\]

- \( \mathcal{V}_{(M+P),M} \in \{X \in \mathbb{R}^{(M+P) \times M} : X^T X = I_M, P > 0\} \) is the Stiefel manifold.

- Constraint is traditional linear eddy-viscosity closure model ansatz → involves overall balance between linear energy production and dissipation / vanishing of averaged total power (= tr\((X^T LX) + \) energy transfer).
  - \( \eta \in \mathbb{R} \): proxy for the balance between linear energy production and energy dissipation (calculated iteratively using modal energy).

- Equation (5) is solved efficiently offline using the method of Lagrange multipliers (Manopt MATLAB toolbox).

- See (Balajewicz, IKT, Dowell, 2016) and Appendix slide for Algorithm.
Basis rotation: remarks

Proposed approach may be interpreted as an *a priori consistent* formulation of the eddy-viscosity turbulence modeling approach.
Basis rotation: remarks

Proposed approach may be interpreted as an *a priori consistent* formulation of the eddy-viscosity turbulence modeling approach.

- Advantages of proposed approach:
  1. Retains *consistency* between ROM and Navier-Stokes equations → no additional turbulence terms required.
Basis rotation: remarks

Proposed approach may be interpreted as an \textit{a priori consistent} formulation of the eddy-viscosity turbulence modeling approach.

- Advantages of proposed approach:
  1. Retains \textit{consistency} between ROM and Navier-Stokes equations \(\rightarrow\) no additional turbulence terms required.
  2. Inherently a \textit{nonlinear} model \(\rightarrow\) should be expected to outperform linear models.
Proposed approach may be interpreted as an *a priori consistent* formulation of the eddy-viscosity turbulence modeling approach.

- **Advantages of proposed approach:**
  1. Retains *consistency* between ROM and Navier-Stokes equations → no additional turbulence terms required.
  2. Inherently a *nonlinear* model → should be expected to outperform linear models.
  3. Works with *any* basis and Petrov-Galerkin projection.
Basis rotation: remarks

Proposed approach may be interpreted as an *a priori consistent* formulation of the eddy-viscosity turbulence modeling approach.

- **Advantages of proposed approach:**
  1. Retains *consistency* between ROM and Navier-Stokes equations → no additional turbulence terms required.
  2. Inherently a *nonlinear* model → should be expected to outperform linear models.
  3. Works with *any* basis and Petrov-Galerkin projection.

- **Disadvantages of proposed approach:**
  1. Off-line calibration of free parameter $\eta$ is required.
Basis rotation: remarks

Proposed approach may be interpreted as an *a priori consistent* formulation of the eddy-viscosity turbulence modeling approach.

- **Advantages of proposed approach:**
  1. Retains *consistency* between ROM and Navier-Stokes equations → no additional turbulence terms required.
  2. Inherently a *nonlinear* model → should be expected to outperform linear models.
  3. Works with *any* basis and Petrov-Galerkin projection.

- **Disadvantages of proposed approach:**
  1. Off-line calibration of free parameter $\eta$ is required.
  2. Stability cannot be proven like for incompressible case.
Numerical results: low Re number cavity

Flow over square cavity at Mach 0.6, Re = 1453.9, Pr = 0.72 ⇒ \( M = 4 \) ROM (91% snapshot energy).

- **Above**: domain and mesh for viscous channel driven cavity problem.
Numerical results: low Re number cavity

- Figure (a) shows evolution of modal energy. Standard ROM is unstable.
- Figure (b) shows phase plot of first and second temporal basis $a_1(t)$ and $a_2(t)$. Stabilized ROM computes stable limit cycle; standard ROM computes unstable spiral.
- Figure (c) is an illustration of the stabilizing rotation matrix. Rotation is small: 
  $$\frac{\|X - I_{(M+P),M}\|_F}{M} = 0.188, X \approx I_{(M+P),M}$$
Numerical results: low Re number cavity

- Pressure power spectral density (PSD) at location $x = (2, -1)$.
Numerical results: moderate Re number cavity

Flow over square cavity at Mach 0.6, $Re = 5452.1$, $Pr = 0.72$ $\Rightarrow M = 20$ ROM (71.8% snapshot energy).

- Above: domain and mesh for viscous channel driven cavity problem.
Numerical results: moderate Re number cavity

- Figure (a) shows evolution of modal energy. Stabilized ROM energy closer to FOM.
- Figure (b) illustrates stabilizing rotation matrix. Rotation is small: $\frac{\|X - I_{(M+P),M}\|_F}{M} = 0.038, X \approx I_{(M+P),M}$
Numerical results: moderate Re number cavity

- Figures show pressure cross PSD of \( p(x_1, t) \) and \( p(x_2, t) \) where \( x_1 = (2, -0.5), x_2 = (0, -0.5) \). Left: power; right: phase lag.

Power and phase lag at fundamental frequency, and first two super harmonics are predicted accurately using the fine-tuned ROM (\( \Delta = \) stabilized ROM, \( \square = \) DNS)
Future work (basis rotation)

- Application to *higher Reynolds number* problems.
- Extension of the proposed approach to problems with *generic nonlinearities*, where the ROM involves some form of hyper-reduction (e.g., DEIM, gappy POD).
- Extension of the method to *minimal-residual-based* nonlinear ROMs.
- Extension of the method to *predictive applications*, e.g., problems with varying Reynolds number and/or Mach number.
- Selecting different *goal-oriented* objectives and constraints in our optimization problem:

\[
\begin{align*}
\text{minimize}_{\mathbf{X} \in \mathcal{V}_{(M+P),M}} & \quad f(\mathbf{X}) \\
\text{subject to} & \quad g(\mathbf{X}, L) = 0
\end{align*}
\]

- e.g.,
  - Maximize parametric robustness:
    \[
    f = \sum_{i=1}^{k} \beta_i \| \mathbf{U}^*(\mu_i)\mathbf{X} - \mathbf{U}^*(\mu_i) \|_F.
    \]
  - ODE constraints: \[ g = \| \mathbf{a}(t) - \mathbf{a}^*(t) \|. \]
Appendix: Continuous vs. discrete Galerkin projection

**Continuous Projection**

- Governing PDEs: \( \dot{q} = \mathcal{L}q \)
- High-fidelity model: \( \dot{q}_N = A_Nq_N \)
- Continuous modal basis: \( \phi_j(x) \)
- Projection of governing PDEs: \( \dot{a}_j = (\phi_j, \mathcal{L}\phi_k)a_k \)
- ROM: \( \dot{a}_M = \Phi^TA_N\Phi a_M \)

**Discrete Projection**

- Governing PDEs: \( \dot{q} = \mathcal{L}q \)
- High-fidelity model: \( \dot{q}_N = A_Nq_N \)
- Discrete modal basis: \( \Phi \)
- Projection of FOM: \( \dot{a}_M = \Phi^TA_N\Phi a_M \)

If PDEs are linear or have polynomial non-linearities, projection can be calculated in **offline stage** of MOR.

* Continuous function space defined using e.g., finite elements.
Appendix: Section 3.3.4 : DS*-POD-ROM

• Original DS models are in LES, where it is considered *state-of-the-art*.

• Derivation requires precise definition of *filtering operation*, unlike other models, which were phenomenological.
  • *LES*: filtering operation effected by convolving flow variables with a rapidly-decaying spatial filter.
  • *POD*: filtering operation is effected by using the POD Galerkin projection (Sec. 3.1).

• Apply *filtering* operation (14) to:

\[
\frac{\partial \mathbf{u}_r}{\partial t} - Re^{-1} \Delta \mathbf{u}_r + \nabla \cdot (\mathbf{u}_r \mathbf{u}_r) + \nabla p = 0. \tag{49}
\]

((49) is equivalent to momentum equation (1) since \(\nabla \cdot \mathbf{u}_r = 0\), one obtains:

\[
\frac{\partial \bar{\mathbf{u}}_r}{\partial t} - Re^{-1} \Delta \bar{\mathbf{u}}_r + \nabla \cdot (\bar{\mathbf{u}}_r \bar{\mathbf{u}}_r) + \nabla \bar{p} = 0. \tag{50}
\]

(assuming differentiation and POD filtering commute).

• **Remarks**:
  • If filtering and differentiation do not commute, one has to estimate commutation error [67-69].
  • Since POD filtering is Galerkin projection (14), we have that \(\bar{\mathbf{u}}_r = \mathbf{u}_r\).

* DS = Dynamic Subgrid