POD Reduced-Order Modeling of Complex Fluid Flows

Zhu Wang
Department of Mathematics
University of South Carolina

ICERM - Algorithms for Dimension and Complexity Reduction
March 25, 2020
Collaborated with Jeff Borggaard, Traian Iliescu (VT), Xuping Xie (NYU)
flow control and optimization

challenges

many-query simulations of large-scale, time-dependent, nonlinear systems. However, short time, even real-time evaluation is needed

what to do?

computational efficient and reliable surrogate – reduced-order models
Outline

1. POD-ROM for Incompressible Fluid Flows
2. Closure Methods for POD-ROM
3. Implementation Improvements
4. Conclusions
Galerkin Projection-Based POD-ROM

Representative Data

POD

Optimal Basis

Galerkin

POD-G-ROM

Z. Wang, USC

POD-ROM of complex flows

ICERM 2020
Galerkin Projection-Based POD-ROM

- Navier-Stokes equations (NSE)
  \[
  \begin{aligned}
  \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\text{Re}} \Delta \mathbf{u} + \nabla p &= 0 \\
  \nabla \cdot \mathbf{u} &= 0
  \end{aligned}
  \]

- non-parametric case in following discussion

- FE, FD, FV ⇒ snapshots \( \mathbf{u}(\cdot, t_i) \)

- \( \mathcal{V} = \text{span} \{ \mathbf{u}(\cdot, t_1), \mathbf{u}(\cdot, t_2), \ldots, \mathbf{u}(\cdot, t_{n_s}) \} \)

- \( \mathbf{s}_{n_{\text{dof}} \times n_s} \xrightarrow{\text{POD}} \) POD basis \( \phi_1, \ldots, \phi_r, \phi_{r+1}, \ldots, \phi_d \)
Galerkin Projection-Based POD-ROM

- **Proper Orthogonal Decomposition (POD)**

\[
\min_{\|\phi_i\|_H^2 = 1} \frac{1}{n_s} \sum_{j=1}^{M} \left\| \mathbf{u}(\cdot, t_j) - \sum_{i=1}^{r} (\mathbf{u}(\cdot, t_j), \phi_i(\cdot))_H \phi_i(\cdot) \right\|_H^2
\]

- \( \mathcal{R}\phi(x) = \lambda\phi(x) \)

\[
\mathcal{R}_{k,j} = \frac{1}{n_s} (\mathbf{u}(\cdot, t_j), \mathbf{u}(\cdot, t_k))_H
\]

- **method of snapshots**
Galerkin Projection-Based POD-ROM

Representative Data → POD → Optimal Basis → Galerkin → POD-G-ROM

\( \phi_1 \)
Galerkin Projection-Based POD-ROM

Representative Data

POD

Optimal Basis

Galerkin

POD-G-ROM

\( \phi_3 \)
Galerkin Projection-Based POD-ROM

Representative Data → POD → Optimal Basis → Galerkin → POD-G-ROM

- \( \mathbf{u}(x, t) \approx \mathbf{u}_r = \mathbf{u}_c(x) + \sum_{i=1}^{r} a_i(t) \phi_i(x) \)

- POD-G ROM

\[
\left( \frac{\partial \mathbf{u}_r}{\partial t}, \phi_k \right) + ((\mathbf{u}_r \cdot \nabla) \mathbf{u}_r, \phi_k) + \left( \frac{2}{\text{Re}} \nabla (\mathbf{u}_r), \nabla \phi_k \right) = 0
\]

\( k = 1, \ldots, r \)

- \( r \sim O(10) << n_{dof} \)

- \( \frac{d\mathbf{a}}{dt} = \mathbf{A} + \mathbf{B}\mathbf{a} + \mathbf{C} (\mathbf{a} \otimes \mathbf{a}) \)

\( \mathbf{A}_{r \times 1}, \mathbf{B}_{r \times r}, \mathbf{C}_{r \times r^2} \) precomputed
Galerkin Projection-Based POD-ROM

\[ \frac{da}{dt} = \mathbf{A} + \mathbf{B}a + \mathbf{C}(a \otimes a) \]

Representative Data

POD

Optimal Basis

Galerkin

POD-G-ROM

→ a **handful** DOF

→ **low** CPU time

→ **same order** accuracy
Galerkin Projection-Based POD-ROM

\[
\frac{da}{dt} = \mathbf{A} + \mathbf{B}a + \mathbf{C}(a \otimes a)
\]

→ a **handful** DOF

→ **low** CPU time

→ **same order** accuracy
Galerkin Projection-Based POD-ROM

\[ \frac{da}{dt} = A + Ba + C(a \otimes a) \]

→ a **handful** DOF

→ **low** CPU time

→ **same order** accuracy
Galerkin Projection-Based POD-ROM

\[ \frac{da}{dt} = A + Ba + C(a \otimes a) \]

→ a **handful** DOF  
→ **low** CPU time  
→ **same order** accuracy

Z. Wang, USC
POD-ROM of complex flows
Galerkin Projection-Based POD-ROM

\[
\frac{da}{dt} = A + Ba + C(a \otimes a)
\]

→ a **handful** DOF  
→ **low** CPU time  
→ **same order** accuracy

- simple flow
- complex flow
Galerkin Projection-Based POD-ROM

\[
\frac{da}{dt} = \mathbf{A} + \mathbf{B}a + \mathbf{C}(a \otimes a)
\]

→ a **handful** DOF  

→ **low** CPU time  

→ **same order** accuracy

▶ simple flow  

▶ complex flow
Galerkin Projection-Based POD-ROM

\[
\frac{\text{d}a}{\text{d}t} = A + Ba + C(a \otimes a)
\]

→ a **handful** DOF

→ **low** CPU time

→ **same order** accuracy

- simple flow
- complex flow
Outline

1. POD-ROM for Incompressible Fluid Flows
2. Closure Methods for POD-ROM
3. Implementation Improvements
4. Conclusions
Energy Cascade

- to model the effect of truncated POD modes
- POD and Fourier are connected  
  Holmes, Lumley, Berkooz, 1996
- energy cascade in solutions of NSE  
  Richardson, 1922

**Big whirls have little whirls that feed on their velocity,**
and **little whirls have lesser whirls and so on to viscosity.**

Kolmogorov, 1941

- at high Reynolds numbers in inertial range, \( E(k) = \alpha \langle \epsilon \rangle^{5/3} k^{-5/3} \)
- beyond it, kinetic energy is negligible

Pope, 2000
Energy Cascade

- to model the effect of truncated POD modes
- POD and Fourier are connected \( \text{Holmes, Lumley, Berkooz, 1996} \)
- energy cascade in solutions of NSE

1. energy is input into the largest scales of the flow;
2. there is an intermediate range in which nonlinearity drives this energy into smaller and smaller scales and conserves the global energy because dissipation is negligible;
3. at small enough scales, dissipation is non negligible and the energy in those smallest scales decays to zero exponentially fast.

\( \text{Layton, 2008} \)
Energy Cascade

- to model the effect of truncated POD modes
- POD and Fourier are connected  \cite{Holmes1996}
- energy cascade for solutions of NSE in POD setting

Couplet, Sagaut, Basdevant,  
J. Fluid Mech., 2003
POD Filter

\[ (\mathbf{u} - \mathbf{u}_r, \phi) = 0 \ (\text{POD projection}) \iff (\mathbf{u} - \bar{\mathbf{u}}, \phi) = 0 \ (\text{Filter}) \]

\[ \text{POD-ROM} \quad \frac{\partial \bar{\mathbf{u}}}{\partial t} + \nabla \cdot (\bar{\mathbf{u}} \bar{\mathbf{u}}) - \text{Re}^{-1} \Delta \bar{\mathbf{u}} = 0 \]

\[ \text{NSE} \quad \frac{\partial \bar{\mathbf{u}}}{\partial t} + \nabla \cdot (\bar{\mathbf{u}} \bar{\mathbf{u}}) - \text{Re}^{-1} \Delta \bar{\mathbf{u}} = 0 \]

\[ \text{to close POD-ROM} \]

1. functional closure
   \[ \nabla \cdot \mathbf{\tau} := -\nu \Delta \bar{\mathbf{u}} \]

2. structural closure
   \[ \nabla \cdot \mathbf{\tau} := \nabla \cdot (\bar{\mathbf{u}}^* \bar{\mathbf{u}}^* - \bar{\mathbf{u}} \bar{\mathbf{u}}) \]
POD Filter

» \( (\mathbf{u} - \mathbf{u}_r, \phi) = 0 \) (POD projection) \iff (\mathbf{u} - \overline{\mathbf{u}}, \phi) = 0 \) (Filter)

» POD-ROM

\[
\frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}} \overline{\mathbf{u}}) - \text{Re}^{-1} \Delta \overline{\mathbf{u}} = 0
\]

» NSE

\[
\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \mathbf{u}) - \text{Re}^{-1} \Delta \mathbf{u} = 0
\]

\[
\frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}} \overline{\mathbf{u}}) + \nabla \cdot \mathbf{\tau} - \text{Re}^{-1} \Delta \overline{\mathbf{u}} = 0
\]

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

to close POD-ROM

1 functional closure \( \nabla \cdot \mathbf{\tau} := -\nu_* \Delta \overline{\mathbf{u}} \)

2 structural closure \( \nabla \cdot \mathbf{\tau} := \nabla \cdot (\overline{\mathbf{u}^* \mathbf{u}^*} - \overline{\mathbf{u}} \overline{\mathbf{u}}) \)
POD Filter

- \((\mathbf{u} - \mathbf{u}_r, \phi) = 0\) (POD projection) \iff \((\mathbf{u} - \bar{\mathbf{u}}, \phi) = 0\) (Filter)

- **POD-ROM**

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

- **NSE**

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

\[
\frac{\partial \bar{\mathbf{u}}}{\partial t} + \nabla \cdot (\bar{\mathbf{u}} \bar{\mathbf{u}}) + \nabla \cdot \mathbf{\tau} - Re^{-1} \Delta \bar{\mathbf{u}} = 0
\]

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

- to close POD-ROM
  1. functional closure
     \[
     \nabla \cdot \mathbf{\tau} := -\nu^* \Delta \bar{\mathbf{u}}
     \]
  2. structural closure
     \[
     \nabla \cdot \mathbf{\tau} := \nabla \cdot (\bar{\mathbf{u}}^* \bar{\mathbf{u}}^* - \bar{\mathbf{u}} \bar{\mathbf{u}})
     \]

Z. Wang, USC
POD Filter

1. (\(u - u_r, \phi\)) = 0 (POD projection) \(\iff\) (\(u - \bar{u}, \phi\)) = 0 (Filter)

2. POD-ROM

\[
\frac{\partial \bar{u}}{\partial t} + \nabla \cdot (\bar{u} \bar{u}) - Re^{-1} \Delta \bar{u} = 0
\]

3. NSE

\[
\frac{\partial \bar{u}}{\partial t} + \nabla \cdot (\bar{u} \bar{u}) - Re^{-1} \Delta \bar{u} = 0
\]

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

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

to close POD-ROM

1. functional closure

\[
\nabla \cdot \tau := -\nu_\ast \Delta \bar{u}
\]

2. structural closure

\[
\nabla \cdot \tau := \nabla \cdot (\bar{u}^* \bar{u}^* - \bar{u} \bar{u})
\]
ML-POD

- mixing-length model

\[
\left( \frac{\partial u_r}{\partial t}, \phi_k \right) + (u_r \cdot \nabla) u_r, \phi_k) + \left( \nu_{ML} + \frac{2}{Re} \right) \mathcal{D}(u_r), \nabla \phi_k \right) = 0, \\
\quad k = 1, \ldots, r
\]

- \( \nu_{ML} := \alpha U_{ML} L_{ML} \)

- Aubry, Holmes, Lumley, Stone, 1988
  Podvin, Lumley, 1998
  
  . . . . . . .
Smagorinsky POD model

\[
\left( \frac{\partial \mathbf{u}_r}{\partial t}, \phi_k \right) + ((\mathbf{u}_r \cdot \nabla) \mathbf{u}_r, \phi_k) + \left( \left( \nu_s + \frac{2}{\text{Re}} \right) \mathcal{D}(\mathbf{u}_r), \nabla \phi_k \right) = 0,
\]

\[k = 1, \ldots, r\]

\[\nu_s := 2(C_S\delta)^2 \|\mathcal{D}(\mathbf{u}_r)\|\]

- Ullmann, Lang, 2010
- Borggaard, Iliescu, Wang, 2011
VMS-POD

- variational multiscale $\iff$ locality energy transfer

Hughes et al. 2000; Guermond 1999; Layton 2002

- VMS-POD model $\vec{u}_r = \vec{u}_L + \vec{u}_S,$

$\vec{u}_L \in X_L = \text{span}\{\phi_1, \ldots, \phi_R\}, \quad \vec{u}_S \in X_S = \text{span}\{\phi_{R+1}, \ldots, \phi_r\}$
VMS-POD

- variational multiscale $\leftrightarrow$ locality energy transfer

Hughes et al. 2000; Guermond 1999; Layton 2002

- VMS-POD model $\mathbf{u}_r = \mathbf{u}_L + \mathbf{u}_S,$

\[\begin{align*}
\mathbf{u}_L &\in X_L = \text{span}\{\phi_1, \ldots, \phi_R\}, \quad \mathbf{u}_S \in X_S = \text{span}\{\phi_{R+1}, \ldots, \phi_r\} \\
\frac{\partial \mathbf{u}_L}{\partial t}, \phi_k + (\mathbf{u}_r \cdot \nabla)\mathbf{u}_r, \phi_k + \left(\frac{2}{\text{Re}} \mathbb{D}(\mathbf{u}_L), \nabla \phi_k\right) &= 0, \quad k = 1, \ldots, R \\
\frac{\partial \mathbf{u}_S}{\partial t}, \phi_k + (\mathbf{u}_r \cdot \nabla)\mathbf{u}_r, \phi_k + \left((\nu_{\text{VMS}} + \frac{2}{\text{Re}}) \mathbb{D}(\mathbf{u}_S), \nabla \phi_k\right) &= 0, \\
&\quad k = R + 1, \ldots, r
\end{align*}\]

- $\nu_{\text{VMS}} := 2(C_S\delta)^2\|\mathbb{D}(\mathbf{u}_S)\|$

- Wang, Akhtar, Borghaard, Iliescu, 2012

- Iliescu, Wang, 2013, 2014
dynamic subgrid-scale model

Germano, Piomelli, Moin, Cabot, Phys. Fluids A 1991

$C_S$ varies dynamically with $x$ and $t$

Filter I: $(u - \bar{u}, \phi) = 0$, $\forall \phi \in X_r =$span{$\phi_1, \ldots, \phi_r$}

$$\frac{\partial \bar{u}}{\partial t} + \nabla \cdot (\bar{u} \bar{u}) + \nabla \cdot (\bar{u} \bar{u} - \bar{u} \bar{u}) - Re^{-1} \Delta \bar{u} = 0$$

$$\tau := -2(C_s \delta)^2 \|D(\bar{u})\|D(\bar{u})$$

Filter II: $(u - \tilde{u}, \phi) = 0$, $\forall \phi \in X_R =$span{$\phi_1, \ldots, \phi_R$}

$$\frac{\partial \tilde{u}}{\partial t} + \nabla \cdot (\tilde{u} \tilde{u}) + \nabla \cdot (\tilde{u} \tilde{u} - \tilde{u} \tilde{u}) - Re^{-1} \Delta \tilde{u} = 0$$

$$T := -2(C_s \tilde{\delta})^2 \|D(\tilde{u})\|D(\tilde{u})$$

$$\tilde{u} \bar{u} - \tilde{u} \tilde{u} = (\tilde{u} \bar{u} - \tilde{u} \tilde{u}) - (\tilde{u} \bar{u} - \bar{u} \bar{u}) = T - \tilde{\tau}$$
DS-POD

- $C_S(x, t)$ determined dynamically

$$Q = 2(\delta)^2 \|\mathcal{D}(\tilde{u})\|_{\mathcal{D}(u)} - 2(\tilde{\delta})^2 \|\mathcal{D}(\tilde{u})\|_{\mathcal{D}(u)}$$

$$C_s^2(x, t) = \frac{[\tilde{u}u - \tilde{u}\tilde{u}] : [Q]}{[Q] : [Q]}$$

- DS-POD ROM

$$\left( \frac{\partial u_r}{\partial t}, \phi_k \right) + ((u_r \cdot \nabla) u_r, \phi_k) + \left( \left( \nu_{DS} + \frac{2}{Re} \right) \mathcal{D}(u_r), \nabla \phi_k \right) = 0,$$

$$k = 1, \ldots, r$$

- $\nu_{DS} := 2(C_S(x, t) \delta)^2 \|\mathcal{D}(u_r)\|$

- Wang, Akhtar, Borggaard, Iliescu, 2012
Numerical Experiments

3D turbulent flow past a cylinder $\text{Re} = 1000$
Energy Spectrum

VMS-POD; DS-POD – accurate
structural closure
\[ \nabla \cdot \tau = \nabla \cdot (\overline{u} \overline{u} - \overline{u} \overline{u}) \approx \nabla \cdot (\overline{u}^* \overline{u}^* - \overline{u} \overline{u}) \]

Stolz, Adam, 1999

dehconvolution

\[ \overline{u} := Gu \]

find \( u \)

image processing, inverse problems

exact deconvolution
\[ u^{ED} = G^{-1} \overline{u} \]

very bad idea

notoriously ill-posed: noise amplification

approximate deconvolution
\[ u^{AD} \approx u^{ED} = G^{-1} \overline{u} \]

Lavrentiev regularization
\[ u^{AD-L} = (G + \mu I)^{-1} \overline{u} \]

Bertero, Boccacci, 1998
approximate deconvolution for closure

define $\overline{u}_r \in X^r$ such that $(I - \alpha^2 \Delta) \overline{u}_r = u \iff \overline{u}_r = Gu$

denote $w_r := \overline{u}_r$

define $w^{AD-L}_r \in X^r$ such that $w^{AD-L}_r = (G + \mu I)^{-1} w_r$

AD-POD ROM

$$\left( \frac{\partial w_r}{\partial t}, \phi_k \right) + \frac{2}{\text{Re}} \left( \mathbb{D}(w_r), \nabla \phi_k \right) + \left( w^{AD-L}_r \cdot \nabla w^{AD-L}_r, \phi_k \right) = 0$$

Xie, Wells, Wang, Iliescu, 2017
Numerical Experiments

3D turbulent flow past a cylinder $\text{Re} = 1000$

- AD-POD $\alpha = 0.3$, $\mu = 0.03$
- energy spectrum
Outline

1. POD-ROM for Incompressible Fluid Flows
2. Closure Methods for POD-ROM
3. Implementation Improvements
4. Conclusions
Basis Generation

- high-fidelity, large-scale dynamic system simulations, e.g. DDM

  snapshot data huge, distributed over multiprocessors

  → POD basis, typically left singular vectors of data

  generation expensive in computation and communication

  → **partitioned methods of snapshots**  
    \[ W., \text{McBee and Iliescu, 2016} \]

Comparison of POD generation algorithms using

- computational complexity in terms of floating-point operations (flops)

- communication effort in terms of floating points to be transferred
POD Basis Generation

- **Singular Value Decomposition (SVD)**

  \[
  S = U\Sigma V^T \quad \rightarrow \quad \phi_j = U(\cdot, j)
  \]

  complexity: \( O(n^2 m + nm^2 + m^3) \) \quad communication: \( nm \)

- **Method of Snapshots (MOS)** \( Sirovich, 1987 \)

  \[
  S^T S z_j = \lambda_j z_j, \text{ for } j = 1, \ldots, r \quad \rightarrow \quad \phi_j = \frac{1}{\sqrt{\lambda_j}} \sum_{\ell=1}^{m} (z_j)_\ell S(\cdot, \ell)
  \]

  complexity: \( O(nm^2 + rmn + m^3) \) \quad communication: \( nm \)
Partitioned Singular Value Decomposition (PSVD)

given local data $S_i$, for $i = 1, \ldots, p$

$$[U_i, \Sigma_i, V_i] = \text{svd}(S_i) \text{ locally}, \quad \mathbf{V} = [V_1^q, V_2^q, \ldots, V_p^q];$$

$$[\hat{V}, \sim, \sim] = \text{svd}(\mathbf{V}); \quad \hat{V}^r = \hat{V}(\cdot, 1 : r);$$

calculate $S\hat{V}^r = [S_1\hat{V}^r; \ldots; S_p\hat{V}^r] \text{ locally};$

$$\text{do } [\hat{U}, \sim, \sim] = \text{svd}(S\hat{V}^r) \quad \rightarrow \quad \phi_j = \hat{U}(\cdot, j)$$

- complexity: $O \left( \sum_{i=1}^p (n_i^2 m + n_i m^2 + m^3 + n_i rm) \right) + O \left( r^3 p^3 + n^2 r + nr^2 + r^3 \right)$

- communication: $nr + mr + mpq$
Partitioned Method of Snapshots (I)

\[ S^T S = \sum_{i=1}^{p} S_i^T S_i \]

**Algorithm 1** Partitioned Method of Snapshots (PMOS)

Let \( S_i \) be local data on the \( i \)-th processor.

\begin{algorithm}
\textbf{for} \( i = 1 \) to \( p \) \textbf{do}
\hspace{1em} Evaluate \( D_i = S_i^T S_i \) \textbf{locally}
\textbf{end for}
\textbf{do} \( D = \sum_{i=1}^{p} D_i \)
\textbf{do} \([V, \Sigma] = \text{eig}(D)\)
\textbf{choose} \( r \) s.t. \( 1 - \sum_{i=1}^{r} \lambda_i / \sum_{i=1}^{d} \lambda_i < \epsilon_1 \)
\textbf{for} \( i = 1 \) to \( p \) \textbf{do}
\hspace{1em} \textbf{for} \( j = 1 \) to \( r \) \textbf{do}
\hspace{2em} calculate POD basis functions \( \phi_j^i = \frac{1}{\sqrt{\lambda_j}} S_i V(:,j) \) \textbf{locally}
\hspace{1em} \textbf{end for}
\textbf{end for}
\textbf{end for}
\end{algorithm}
Partitioned Method of Snapshots (II)

\[ S^T S = \sum_{i=1}^{p} S_i^T S_i = \sum_{i=1}^{p} V_i \Sigma_i^2 V_i^T \]

\[ = [V_1 \Sigma_1^T, V_2 \Sigma_2^T, \ldots, V_p \Sigma_p^T] \begin{bmatrix} \Sigma_1 V_1^T \\ \Sigma_2 V_2^T \\ \vdots \\ \Sigma_p V_p^T \end{bmatrix} = WW^T \]

\[ W \approx W' = \begin{bmatrix} V_1^{r_1} (\Sigma_1^{r_1})^T, V_2^{r_2} (\Sigma_2^{r_2})^T, \ldots, V_p^{r_p} (\Sigma_p^{r_p})^T \end{bmatrix} \]

\[ W' = XY^T \]

\[ V \approx X, \text{ and } \Sigma \approx \Lambda \]
Algorithm 2 Approximate Partitioned Method of Snapshots (APMOS)

Let $S_i$ be local data on the $i$-th processor.

for $i = 1$ to $p$ do

$[U_i, \Sigma_i, V_i] = \text{svd}(S_i)$ locally

select $r_i$, s.t., $\sigma_i^{r_i+1} < \epsilon_0$

take $V_i^{r_i} = V_i(\cdot, 1:r_i)$ and $\Sigma_i^{r_i} = \Sigma_i(1:r_i, 1:r_i)$

end for

assemble $W^r = \begin{bmatrix} V_1^{r_1}(\Sigma_1^{r_1})^T, \ldots, V_p^{r_p}(\Sigma_p^{r_p})^T \end{bmatrix}$

do $[X, \Lambda, Y] = \text{svd}(W^r)$

choose $r$, s.t. $1 - \sum_{i=1}^r \lambda_i / \sum_{i=1}^d \lambda_i < \epsilon_1$

for $i = 1$ to $p$ do

for $j = 1$ to $r$ do

calculate POD basis functions $\tilde{\phi}_j^i = \frac{1}{\sqrt{\lambda_j}} S_i X(\cdot,j)$ locally

end for

end for
### Comparison

<table>
<thead>
<tr>
<th>Method</th>
<th>Complexity (flops)</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMOS</td>
<td>$O\left(\sum_{i=1}^{p} n_i m^2 + \sum_{i=1}^{p} n_i r m\right)$ + $O(m^3)$</td>
<td>$m^2 p + m r$</td>
</tr>
<tr>
<td>APMOS</td>
<td>$O\left(\sum_{i=1}^{p} (n_i^2 m + n_i m^2 + m^3 + n_i r m)\right)$ + $O\left(\left(\sum_{i=1}^{p} r_i\right)^3\right)$</td>
<td>$m \sum_{i=1}^{p} r_i + m r + r$</td>
</tr>
</tbody>
</table>

* Underline represents operations to be executed in parallel.

### Theorem

Let $\lambda_j$ be the $j$-th largest eigenvalue of $W^r (W^r)^\top$ and $\overline{\lambda}_j$ the $j$-th largest eigenvalue of $W (W)^\top$. For $1 \leq j \leq m$,

$$|\lambda_j - \overline{\lambda}_j| \leq p \epsilon_0^2.$$
Numerical Experiment

Gravity current (such as the Red Sea overflow entering the Tadjura Rift)

\[
\begin{align*}
\frac{\partial u}{\partial t} - \Delta u + (u \cdot \nabla)u + \nabla p - Ra T k &= 0, \\
\nabla \cdot u &= 0, \\
\frac{\partial T}{\partial t} + (u \cdot \nabla) T - Pr^{-1} \Delta T &= 0,
\end{align*}
\]

- driven by velocity and temperature forcing profiles at the inlet
- \(Ra = 5 \times 10^6\) and \(Pr = 7\)
- horizontal length of \(L = 10\) km
- depth of the water column ranges from \(K = 400\) m at \(x = 0\) to \(H = 1000\) m at \(x = 10\) km over a constant slope (\(\theta = 3.5^\circ\))
- BC of velocity: homogeneous Dirichlet on the bottom, nonhomogeneous Dirichlet at the inlet, free slip on top, and zero normal flux at the outlet
- \(S \approx 14, 400 \times 900\)
Numerical Experiment

- $\epsilon_0 = 10^{-1}$, the shape of $W'$ is $900 \times 357$;
- $\epsilon_0 = 10^{-2}$, the shape of $W'$ is $900 \times 389$;
- $\epsilon_0 = 10^{-3}$, the shape of $W'$ is $900 \times 406$.

Figure 1: Gravity current. From the top: the first, fifth and tenth POD basis functions of the temperature generated by APMOS with $\epsilon_0 = 0.1$ and $p = 10$. 
Numerical Experiment

- APMOS with $p = 10$ and different values of $\epsilon_0$

<table>
<thead>
<tr>
<th>error</th>
<th>$\epsilon_0 = 0.1$</th>
<th>$\epsilon_0 = 0.01$</th>
<th>$\epsilon_0 = 0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$|\phi_1 - \tilde{\phi}_1|$</td>
<td>1.3773e-07</td>
<td>1.4887e-10</td>
<td>8.0819e-13</td>
</tr>
<tr>
<td>$|\phi_2 - \tilde{\phi}_2|$</td>
<td>3.1577e-07</td>
<td>3.9002e-09</td>
<td>1.3181e-11</td>
</tr>
<tr>
<td>$|\phi_3 - \tilde{\phi}_3|$</td>
<td>8.4477e-07</td>
<td>1.6343e-08</td>
<td>8.4092e-11</td>
</tr>
<tr>
<td>$|\phi_4 - \tilde{\phi}_4|$</td>
<td>1.5442e-06</td>
<td>5.1908e-08</td>
<td>2.0944e-10</td>
</tr>
<tr>
<td>$|\phi_5 - \tilde{\phi}_5|$</td>
<td>1.9823e-06</td>
<td>6.5121e-08</td>
<td>2.7802e-10</td>
</tr>
<tr>
<td>$|\phi_6 - \tilde{\phi}_6|$</td>
<td>2.3384e-06</td>
<td>4.5175e-08</td>
<td>5.5131e-10</td>
</tr>
<tr>
<td>$|\phi_7 - \tilde{\phi}_7|$</td>
<td>4.5517e-06</td>
<td>2.9049e-08</td>
<td>2.6552e-09</td>
</tr>
<tr>
<td>$|\phi_8 - \tilde{\phi}_8|$</td>
<td>2.2137e-05</td>
<td>3.9492e-08</td>
<td>2.9162e-09</td>
</tr>
<tr>
<td>$|\phi_9 - \tilde{\phi}_9|$</td>
<td>2.3534e-05</td>
<td>4.2363e-08</td>
<td>1.1833e-09</td>
</tr>
<tr>
<td>$|\phi_{10} - \tilde{\phi}_{10}|$</td>
<td>5.0482e-06</td>
<td>2.5171e-08</td>
<td>3.1499e-10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>error</th>
<th>$\epsilon_0 = 0.1$</th>
<th>$\epsilon_0 = 0.01$</th>
<th>$\epsilon_0 = 0.001$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\max_{j \in [1,10]}</td>
<td>\lambda_j - \tilde{\lambda}_j</td>
<td>$</td>
<td>3.9457e-03</td>
</tr>
</tbody>
</table>
Nonlinear Closure

\[ \frac{\text{d} \mathbf{a}}{\text{d} t} = \mathbf{A} + \tilde{\mathbf{A}}(\mathbf{u}_r(\cdot, t)) + \left( \mathbf{B} + \tilde{\mathbf{B}}(\mathbf{u}_r(\cdot, t)) \right) \mathbf{a} + C(\mathbf{a} \otimes \mathbf{a}) \]

- \( \tilde{\mathbf{A}}, \tilde{\mathbf{B}} \) nonlinear
General Nonlinear ROMs

\[ \frac{\text{da}}{\text{dt}} = \mathcal{A} + \mathcal{B} \text{a} + (\mathcal{N}(u_r), \Phi) \]

- \(\mathcal{N} = (\mathcal{N}(u_r), \Phi) : r \times 1\) with \(\mathcal{N}_k = - \left( \mathcal{N}(\sum_{j=1}^r \phi_j a_j(t)), \phi_k \right)\)

\(\mathcal{N}\) needs to be assembled at each time step/iteration

computational complexity depends on \(n_e n_q\) of full-order model

- hyper-reduction, e.g., DEIM/QDEIM

Chaturantabut and Sorensen, 2010; Drmac and Gugercin 2015
Difficulties of DEIM in CG

- online simulations \((P^T F(u))\) involve integrations in FE discretization
Difficulties of DEIM in CG

- online simulations ($P^T F(u)$) involve integrations in FE discretization
DEIM in CG

- replace nonlinear functions with their finite element interpolant (FEIC)

\[ \mathcal{I}^h N(u) = \sum_{i=1}^{n} N(u_i(t)) h_i(x) \]

(POD-FEIC) \[ \mathcal{N}_{FEIC} = (\mathcal{I}^h N(u_r), \phi) = \Phi^T M^h N(\Phi a(t)) \]

\( \Phi^T M^h \) pre-computable, \( N(\Phi a(t)) \) evaluated at FE nodes

(POD-FEIC-DEIM) \[ \mathcal{N}_{FEIC-DEIM} = \Phi^T M^h \psi (P^T \psi)^{-1} P^T N(\Phi a(t)) \]

\[ [\Phi^T M^h \psi (P^T \psi)^{-1}]_{r \times p} \) precomputable

online simulation \( N(\Phi a(t)) \) evaluated at selected DEIM pts, \( P^T N(\Phi a(t)) \)

- computational complexity at each step/iteration is \( \mathcal{O}(4rp + \varphi(p)) \) flops \( W., 2015 \)

No need for this in the DG setting
POD-FEIC-DEIM for Nonlinear Closure

\[
\begin{cases}
    u_t - \nu u_{xx} + uu_x = 0, & \text{in } \Omega \times [0, T], \\
    u(x, t) = 0, & \text{on } \partial\Omega \times [0, T], \\
    u(x, 0) = u_0(x), & \text{in } \Omega.
\end{cases}
\]

- \( \nu = 10^{-3} \)
- \( u_0 = \begin{cases} 1 & 0 < x \leq 0.5 \\ 0 & \text{others} \end{cases} \)
- \( \Omega = [0, 1], \ T = 1 \)
- nonlinear closure term \(-C_s |u_x| u_{xx}\)
$r = 20$

$p = 40$ in DEIM

speed-up-factor $> 100$

<table>
<thead>
<tr>
<th>model</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-POD-FE</td>
<td>119.50</td>
</tr>
<tr>
<td>S-POD-FEIC</td>
<td>2.05</td>
</tr>
<tr>
<td>S-POD-FEIC-DEIM</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Outline

1. POD-ROM for Incompressible Fluid Flows
2. Closure Methods for POD-ROM
3. Implementation Improvements
4. Conclusions
Conclusions

- closure methods for POD-ROMs of turbulent flows
  - eddy viscosity; approximate deconvolution
- partitioned methods of snapshots for improving POD basis generation
- FE with interpolated coefficients for nonlinear ROMs
- future work
  - deal with parameterized problems using adaptivity
  - seeking offline adaptive basis construction: Chellappa, Feng and Benner, 2019; · · ·
  - seeking online adaptivity: Peherstorfer and Willcox 2015; Peherstorfer 2019; Carlberg, 2015; Etter and Carlberg, 2019; · · ·
Thank You!