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)

Turbulent Flows

Photographer Christian Steiness. http://www.ict-aeolus.eu

- flow control and optimization

  computer aided design

- challenges

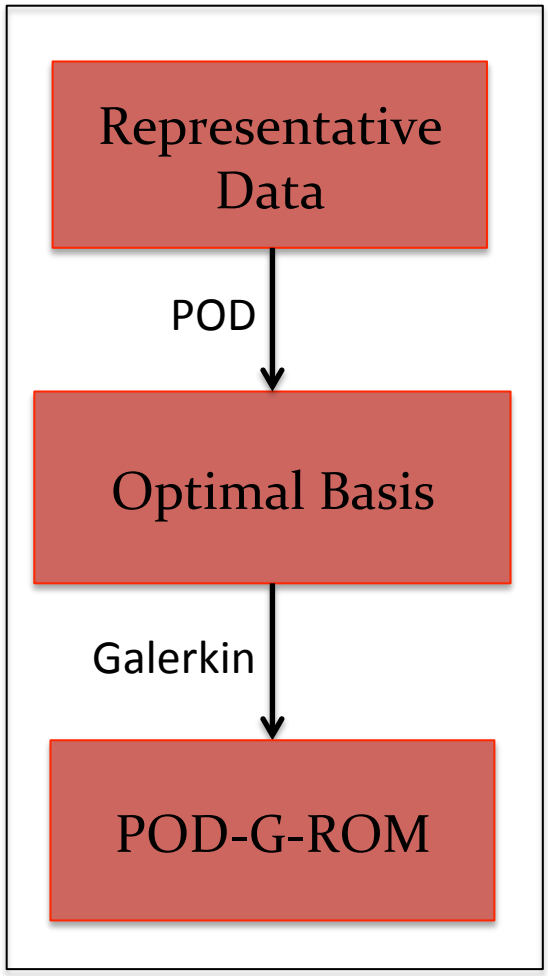
  many-query simulations of **large-scale, time-dependent, nonlinear** systems. However, **short time, even real-time** evaluation is needed

- what to do?

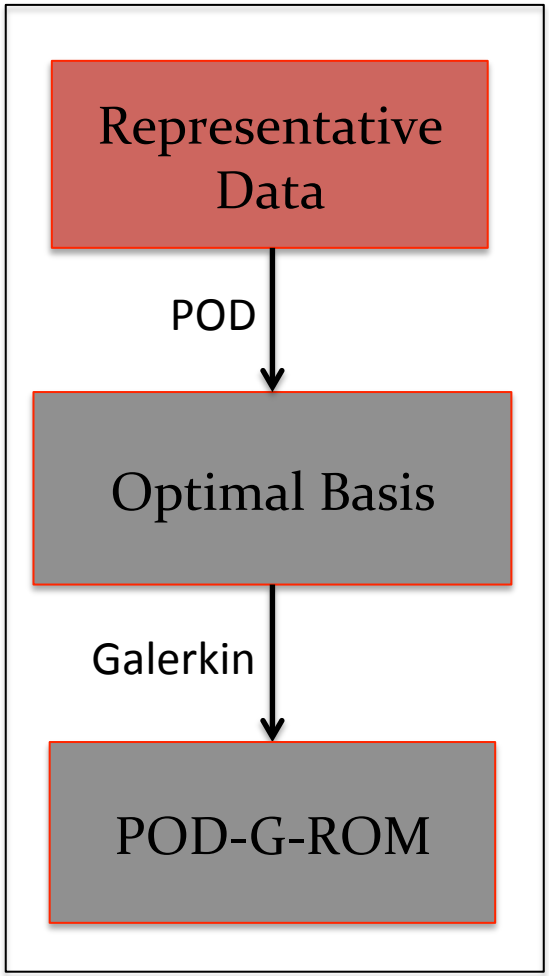
  computational efficient and reliable surrogate – reduced-order models

Outline

Galerkin Projection-Based POD-ROM



Galerkin Projection-Based POD-ROM

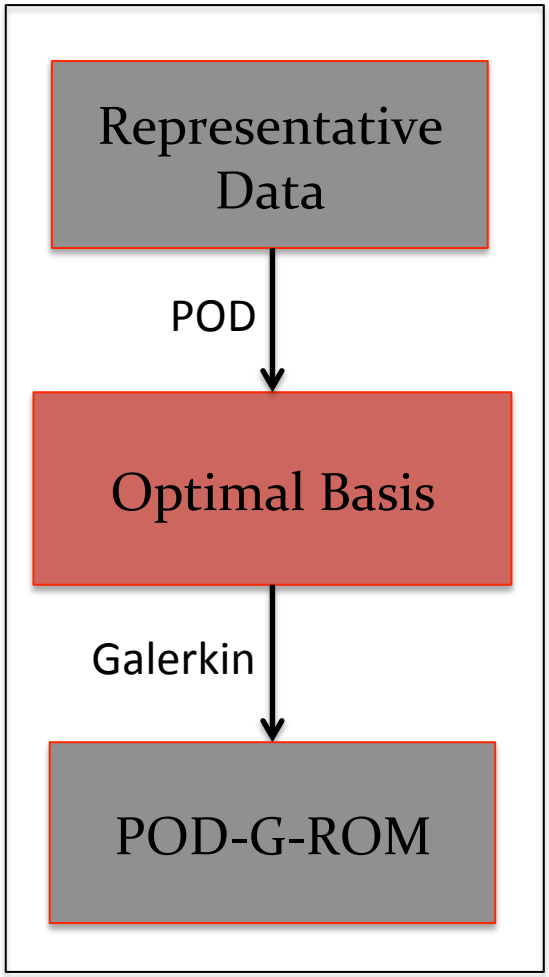


- Navier-Stokes equations (NSE)

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\mathrm{Re}}\Delta \mathbf{u} + \nabla p = 0 \\ \nabla \cdot \mathbf{u} = 0 \end{cases}$$

- non-parametric case in following discussion
- FE, FD, FV $\Rightarrow$ snapshots $\mathbf{u}(\cdot, t_i)$
- $\mathcal{V} = \mathrm{span}\,\{\mathbf{u}(\cdot, t_1), \mathbf{u}(\cdot, t_2), \ldots, \mathbf{u}(\cdot, t_{n_s})\}$
- $\mathbf{S}_{n_{dof} \times n_s} \overset{POD}{\Longrightarrow}$ POD basis $\phi_1, \ldots, \phi_r, \phi_{r+1}, \ldots, \phi_d$

Galerkin Projection-Based POD-ROM



- Proper Orthogonal Decomposition (POD)

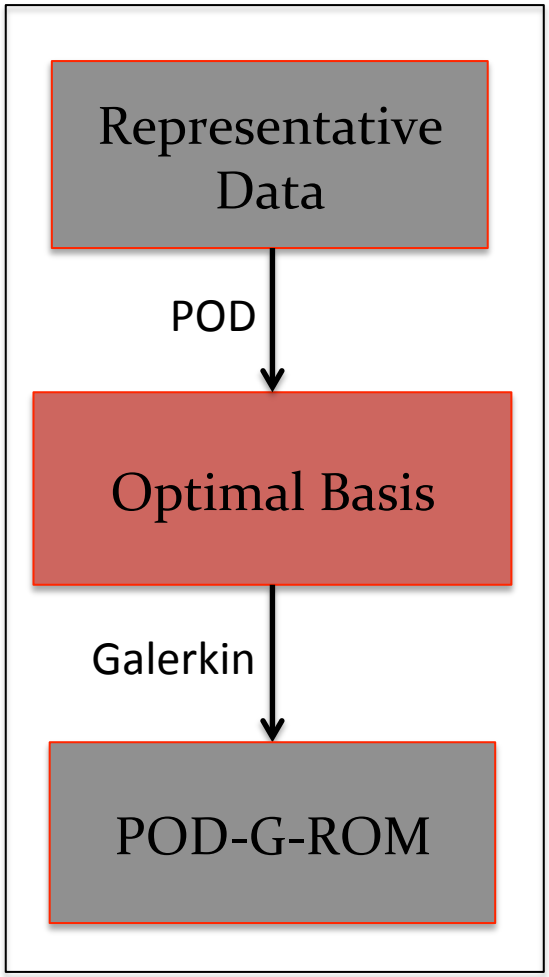
$$\min_{\|\phi_i\|_{\mathcal{H}}^2=1} \frac{1}{n_s}\sum_{j=1}^{M}\left\|\mathbf{u}(\cdot,t_j)-\sum_{i=1}^{r}(\mathbf{u}(\cdot,t_j),\phi_i(\cdot))_{\mathcal{H}}\,\phi_i(\cdot)\right\|_{\mathcal{H}}^2$$

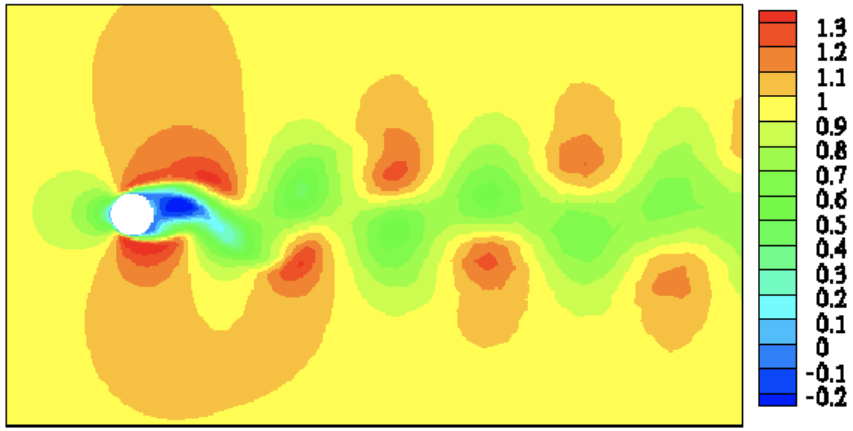
- $\mathcal{R}\phi(x)=\lambda\phi(x)$

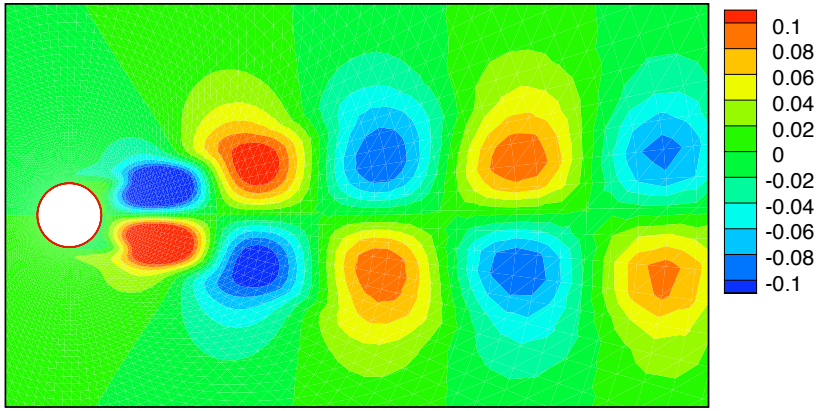
$$\mathcal{R}_{k,j}=\frac{1}{n_s}(\mathbf{u}(\cdot,t_j),\mathbf{u}(\cdot,t_k))_{\mathcal{H}}$$

- method of snapshots

Galerkin Projection-Based POD-ROM

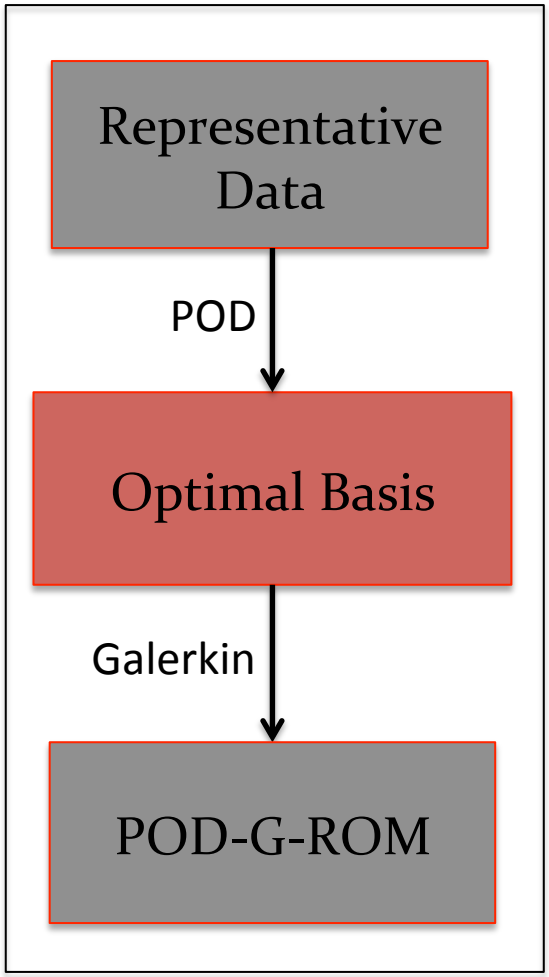


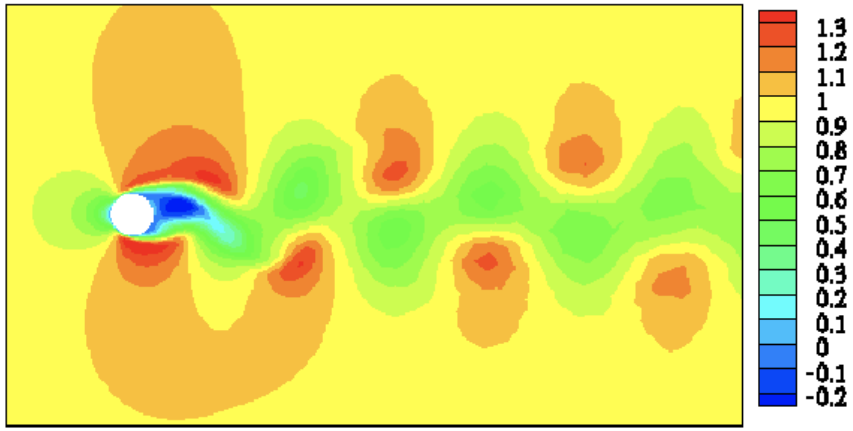


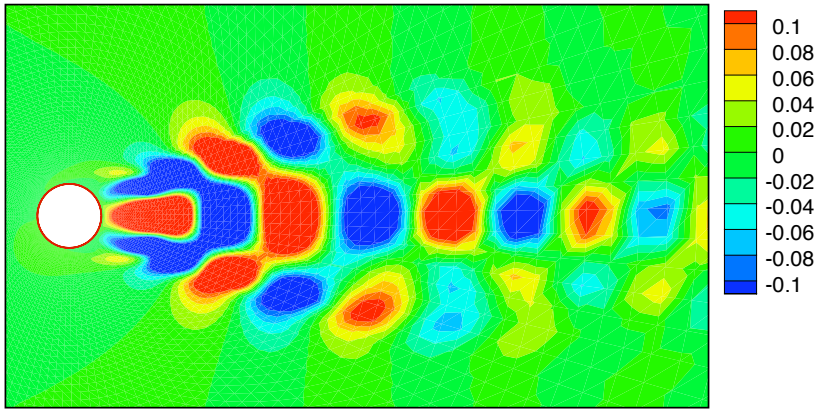


ϕ_1

Galerkin Projection-Based POD-ROM

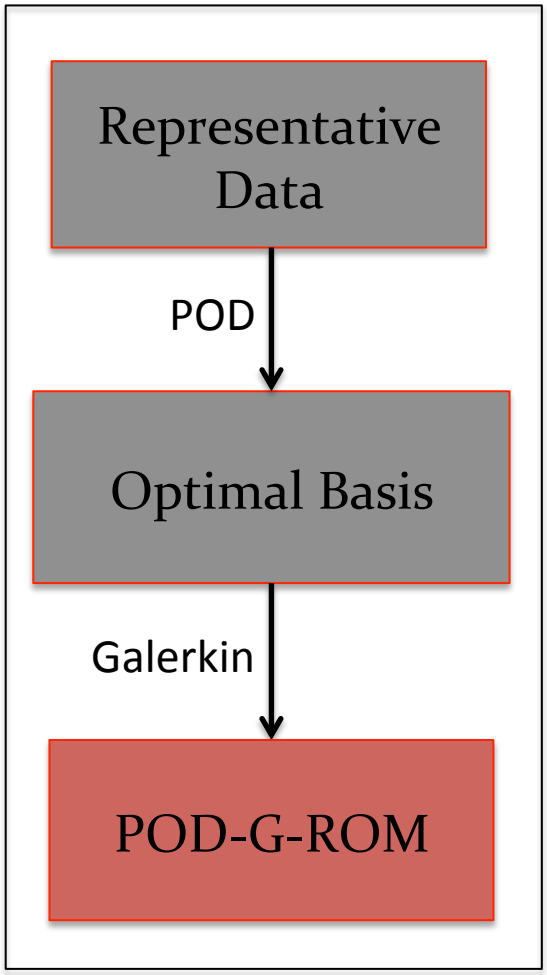






ϕ_3

Galerkin Projection-Based POD-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

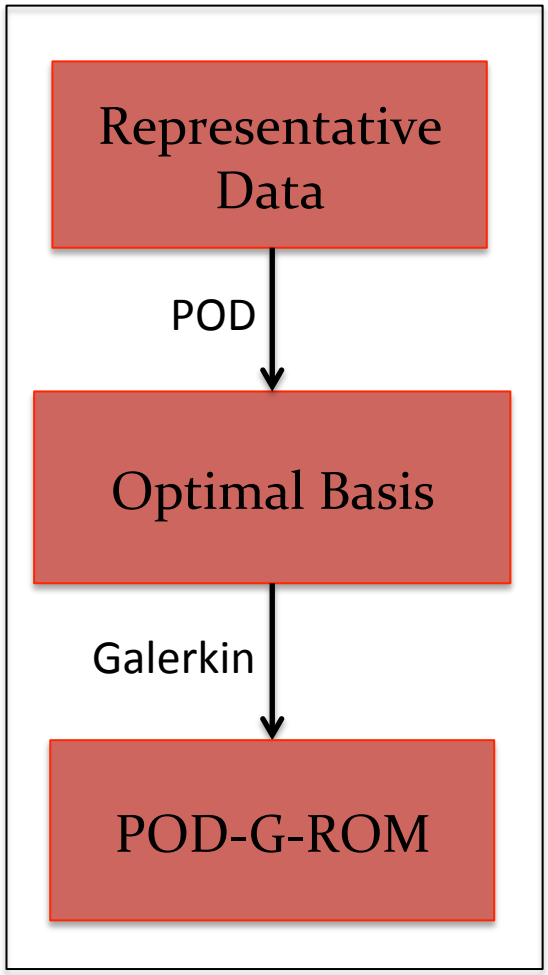
$$\left(\frac{\partial \mathbf{u}_r}{\partial t}, \phi_k\right) + ((\mathbf{u}_r \cdot \nabla)\mathbf{u}_r, \phi_k) + \left(\frac{2}{\mathrm{Re}}\mathbb{D}(\mathbf{u}_r), \nabla\phi_k\right) = 0$$

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

- r $\sim$ O(10) $<<$ n_{dof}
- $\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = \mathcal{A} + \mathcal{B}\mathbf{a} + \mathcal{C}(\mathbf{a} \otimes \mathbf{a})$

$\mathcal{A}_{r\times 1}, \mathcal{B}_{r\times r}, \mathcal{C}_{r\times r^2}$ precomputed

Galerkin Projection-Based POD-ROM



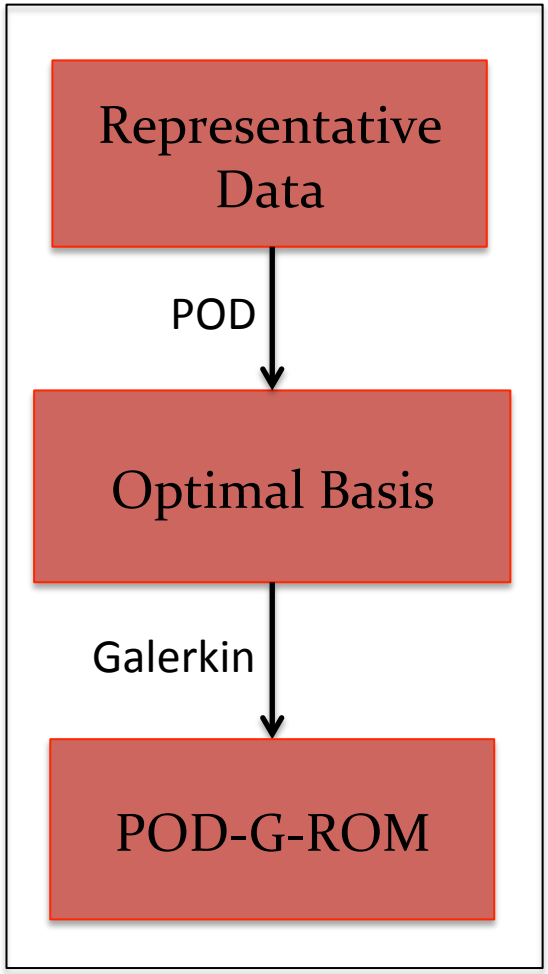
$$\frac{d\mathbf{a}}{dt} = \mathcal{A} + \mathcal{B}\mathbf{a} + \mathcal{C}(\mathbf{a} \otimes \mathbf{a})$$

$\rightarrow$ a **handful** DOF

$\rightarrow$ **low** CPU time

$\rightarrow$ **same order** accuracy

Galerkin Projection-Based POD-ROM



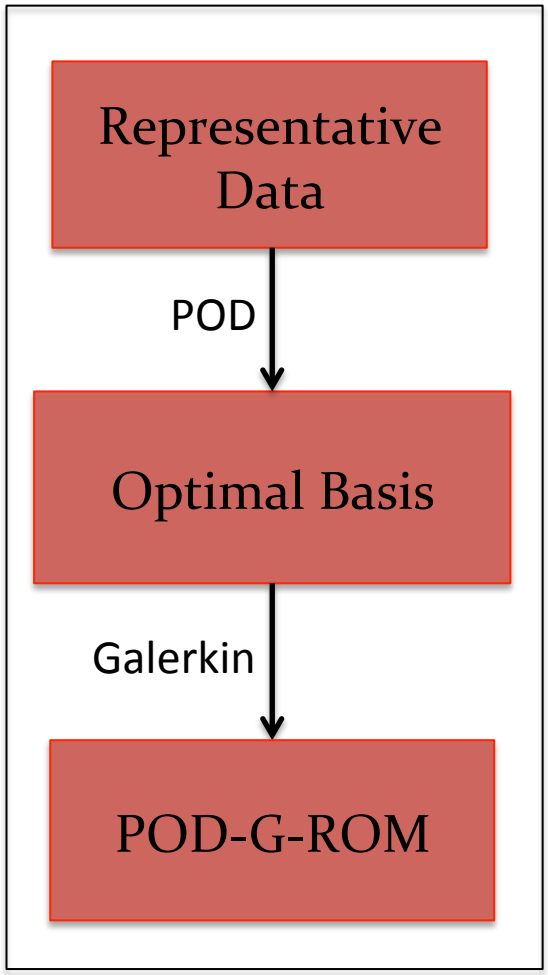
$$\frac{d\mathbf{a}}{dt} = \mathcal{A} + \mathcal{B}\mathbf{a} + \mathcal{C}(\mathbf{a} \otimes \mathbf{a})$$

$\rightarrow$ a **handful** DOF $\surd$

$\rightarrow$ **low** CPU time

$\rightarrow$ **same order** accuracy

Galerkin Projection-Based POD-ROM



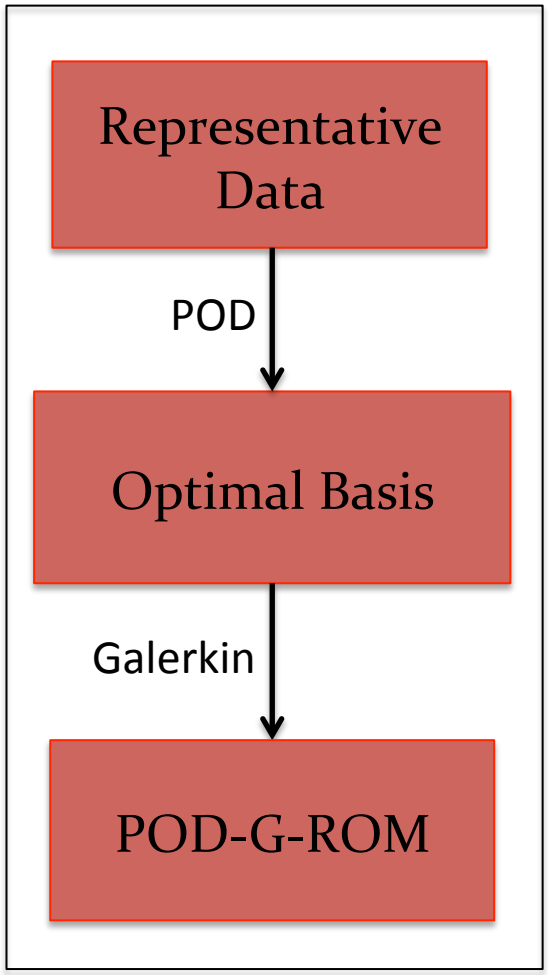
$$\boxed{\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = \mathcal{A} + \mathcal{B}\mathbf{a} + \mathcal{C}(\mathbf{a} \otimes \mathbf{a})}$$

$\rightarrow$ a **handful** DOF $\surd$

$\rightarrow$ **low** CPU time $\surd$

$\rightarrow$ **same order** accuracy

Galerkin Projection-Based POD-ROM



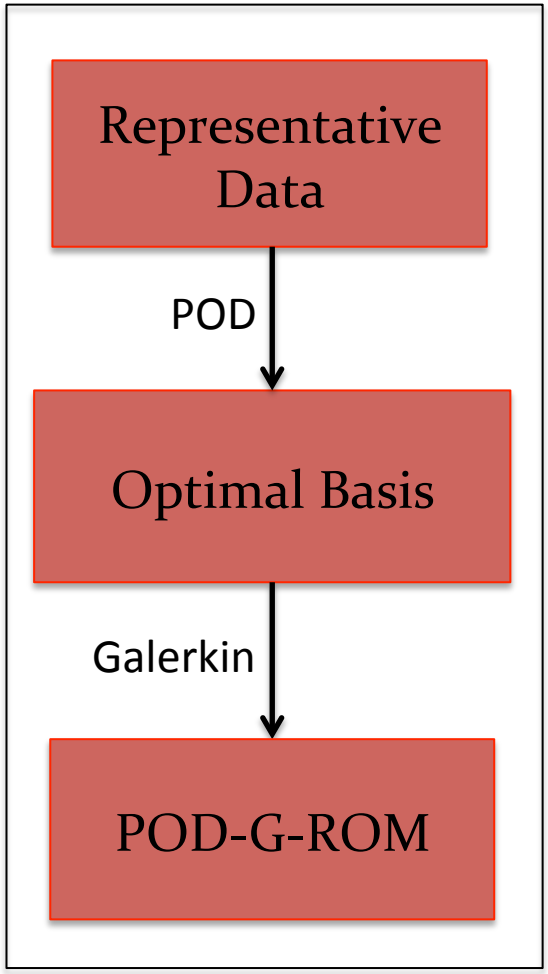
$$\frac{d\mathbf{a}}{dt} = \mathcal{A} + \mathcal{B}\mathbf{a} + \mathcal{C}(\mathbf{a} \otimes \mathbf{a})$$

$\rightarrow$ a **handful** DOF $\surd$

$\rightarrow$ **low** CPU time $\surd$

$\rightarrow$ **same order** accuracy ?

Galerkin Projection-Based POD-ROM



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

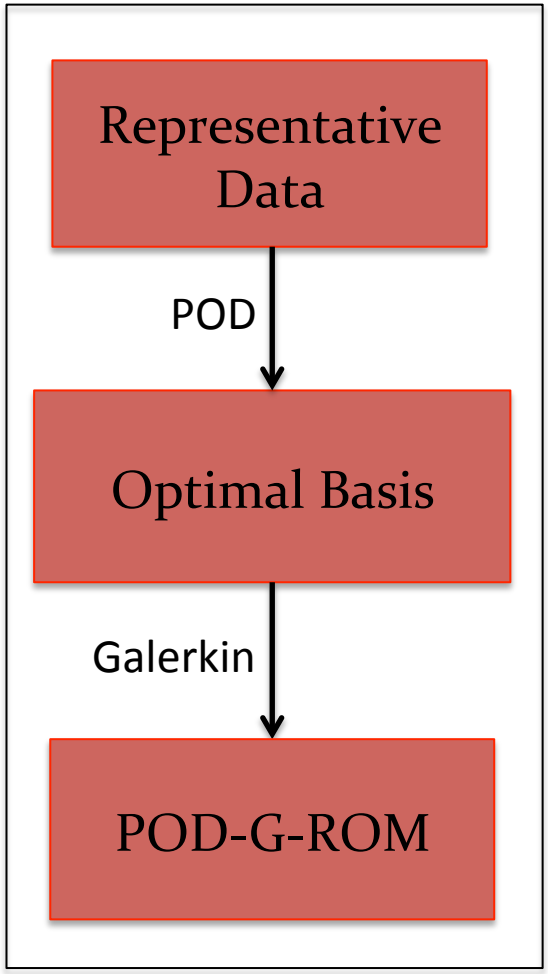
$\rightarrow$ a **handful** DOF $\surd$

$\rightarrow$ **low** CPU time $\surd$

$\rightarrow$ **same order** accuracy

- simple flow
- complex flow

Galerkin Projection-Based POD-ROM



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

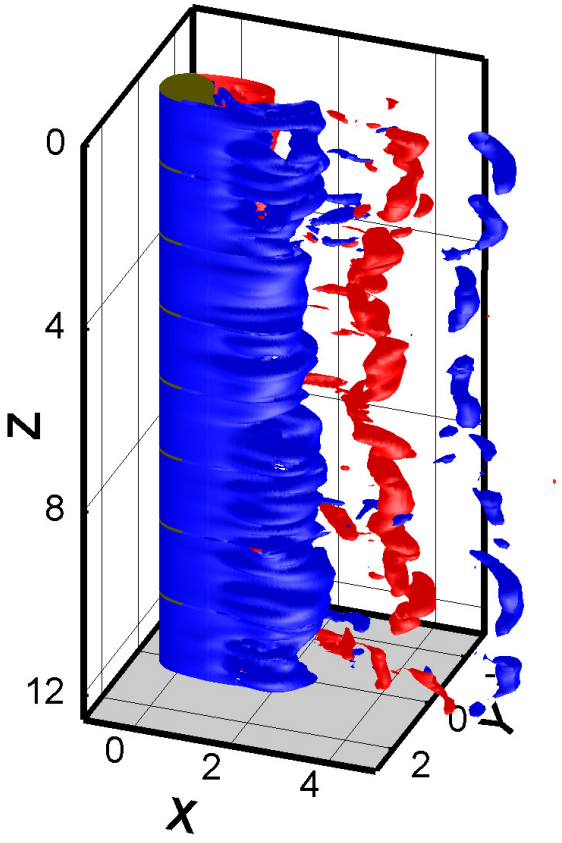
$\rightarrow$ a **handful** DOF $\surd$

$\rightarrow$ **low** CPU time $\surd$

$\rightarrow$ **same order** accuracy

- simple flow $\surd$
- complex flow

Galerkin Projection-Based POD-ROM



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

$\rightarrow$ a **handful** DOF $\surd$

$\rightarrow$ **low** CPU time $\surd$

$\rightarrow$ **same order** accuracy

- simple flow $\surd$
- complex flow $\times$

Outline

Energy Cascade

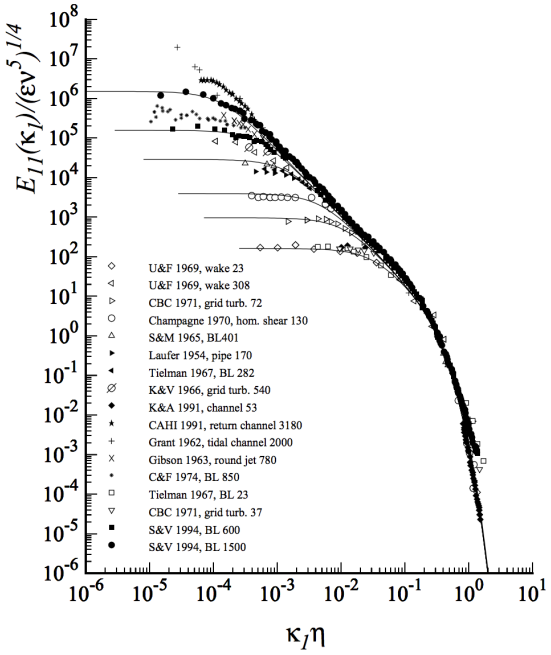
- to model the effect of truncated POD modes
- POD and Fourier are connected *Holmes, Lumley, Berkooz, 1996*
- energy cascade in solutions of NSE

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, $E(k) = \alpha \langle \epsilon \rangle^{\frac{2}{3}} k^{-\frac{5}{3}}$
beyond it, kinetic energy is negligible



Pope, 2000

Energy Cascade

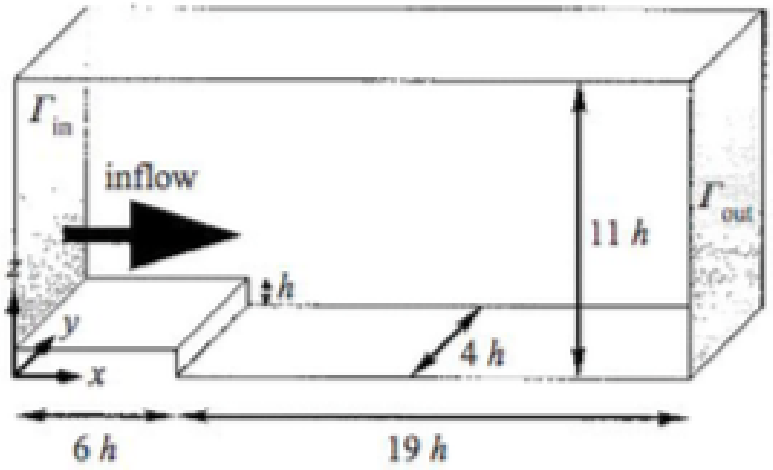
- to model the effect of truncated POD modes
- POD and Fourier are connected *Holmes, Lumley, Berkooz, 1996*
- energy cascade in solutions of NSE

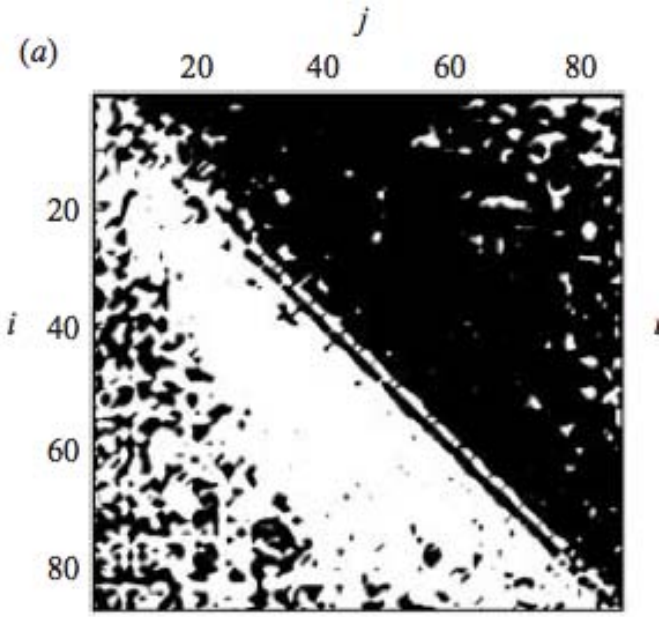
1. energy is input into the largest scales of the flow;
2. there is an intermediate range in which nonlinearity drives this energy into smaller and smaller scales and conserves the global energy because dissipation is negligible;
3. at small enough scales, dissipation is non negligible and the energy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

Layton, 2008

Energy Cascade

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





- Couplet, Sagaut, Basdevant,

  J. Fluid Mech., 2003

POD Filter

- $(\mathbf{u} - \mathbf{u}_r, \phi) = 0$ (POD projection) $\Longleftrightarrow$ $(\mathbf{u} - \overline{\mathbf{u}}, \phi) = 0$ (Filter)
- POD-ROM $\quad \frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}}\,\overline{\mathbf{u}}) - \mathrm{Re}^{-1} \Delta \overline{\mathbf{u}} = 0$
- $\overline{\text{NSE}}$ $\quad \frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}\,\mathbf{u}}) - \mathrm{Re}^{-1} \Delta \overline{\mathbf{u}} = 0$

$$\frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}}\,\overline{\mathbf{u}}) + \nabla \cdot \boldsymbol{\tau} - \mathrm{Re}^{-1} \Delta \overline{\mathbf{u}} = 0$$

$$\boldsymbol{\tau} = \overline{\mathbf{u}\,\mathbf{u}} - \overline{\mathbf{u}}\,\overline{\mathbf{u}}$$

- to close POD-ROM
  1. functional closure $\quad \boxed{\nabla \cdot \boldsymbol{\tau} := -\nu_* \Delta \overline{\mathbf{u}}}$
  2. structural closure $\quad \boxed{\nabla \cdot \boldsymbol{\tau} := \nabla \cdot (\overline{\mathbf{u}^* \mathbf{u}^*} - \overline{\mathbf{u}}\,\overline{\mathbf{u}})}$

POD Filter

- $(\mathbf{u} - \mathbf{u}_r, \phi) = 0$ (POD projection) $\Longleftrightarrow$ $(\mathbf{u} - \overline{\mathbf{u}}, \phi) = 0$ (Filter)
- POD-ROM $\quad \frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}}\,\overline{\mathbf{u}}) - \mathrm{Re}^{-1} \Delta \overline{\mathbf{u}} = 0$
- $\overline{\text{NSE}}$ $\quad \frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}\,\mathbf{u}}) - \mathrm{Re}^{-1} \Delta \overline{\mathbf{u}} = 0$

$$\frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}}\,\overline{\mathbf{u}}) + \nabla \cdot \boldsymbol{\tau} - \mathrm{Re}^{-1} \Delta \overline{\mathbf{u}} = 0$$

$$\boldsymbol{\tau} = \overline{\mathbf{u}\,\mathbf{u}} - \overline{\mathbf{u}}\,\overline{\mathbf{u}}$$

- to close POD-ROM
  1. functional closure $\quad \boxed{\nabla \cdot \boldsymbol{\tau} := -\nu_* \Delta \overline{\mathbf{u}}}$
  2. structural closure $\quad \boxed{\nabla \cdot \boldsymbol{\tau} := \nabla \cdot (\overline{\mathbf{u}^* \mathbf{u}^*} - \overline{\mathbf{u}}\,\overline{\mathbf{u}})}$

POD Filter

- $(\mathbf{u} - \mathbf{u}_r, \phi) = 0$ (POD projection) $\Longleftrightarrow$ $(\mathbf{u} - \overline{\mathbf{u}}, \phi) = 0$ (Filter)
- POD-ROM $\quad \frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}}\,\overline{\mathbf{u}}) - \mathrm{Re}^{-1} \Delta \overline{\mathbf{u}} = 0$
- $\overline{\text{NSE}}$ $\quad \frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}\,\mathbf{u}}) - \mathrm{Re}^{-1} \Delta \overline{\mathbf{u}} = 0$

$$\frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}}\,\overline{\mathbf{u}}) + \nabla \cdot \boldsymbol{\tau} - \mathrm{Re}^{-1} \Delta \overline{\mathbf{u}} = 0$$

$$\boldsymbol{\tau} = \overline{\mathbf{u}\,\mathbf{u}} - \overline{\mathbf{u}}\,\overline{\mathbf{u}}$$

- to close POD-ROM
  1. functional closure $\quad \boxed{\nabla \cdot \boldsymbol{\tau} := -\nu_* \Delta \overline{\mathbf{u}}}$
  2. structural closure $\quad \boxed{\nabla \cdot \boldsymbol{\tau} := \nabla \cdot (\overline{\mathbf{u}^* \mathbf{u}^*} - \overline{\mathbf{u}}\,\overline{\mathbf{u}})}$

POD Filter

- $(\mathbf{u} - \mathbf{u}_r, \phi) = 0$ (POD projection) $\Longleftrightarrow$ $(\mathbf{u} - \overline{\mathbf{u}}, \phi) = 0$ (Filter)
- POD-ROM $\quad \frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}}\ \overline{\mathbf{u}}) - \mathrm{Re}^{-1} \Delta \overline{\mathbf{u}} = 0$
- $\overline{\text{NSE}}$ $\quad \frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}\,\mathbf{u}}) - \mathrm{Re}^{-1} \Delta \overline{\mathbf{u}} = 0$

$$\frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}}\ \overline{\mathbf{u}}) + \nabla \cdot \boldsymbol{\tau} - \mathrm{Re}^{-1} \Delta \overline{\mathbf{u}} = 0$$

$$\boldsymbol{\tau} = \overline{\mathbf{u}\,\mathbf{u}} - \overline{\mathbf{u}}\ \overline{\mathbf{u}}$$

- to close POD-ROM
  1. functional closure $\quad \boxed{\nabla \cdot \boldsymbol{\tau} := -\nu_* \Delta \overline{\mathbf{u}}}$
  2. structural closure $\quad \boxed{\nabla \cdot \boldsymbol{\tau} := \nabla \cdot (\overline{\mathbf{u}^* \mathbf{u}^*} - \overline{\mathbf{u}}\ \overline{\mathbf{u}})}$

ML-POD

- mixing-length model

$$\left(\frac{\partial \mathbf{u}_r}{\partial t}, \phi_k\right) + ((\mathbf{u}_r \cdot \nabla)\mathbf{u}_r, \phi_k) + \left(\left(\nu_{ML} + \frac{2}{\mathrm{Re}}\right) \mathbb{D}(\mathbf{u}_r), \nabla \phi_k\right) = 0, \qquad k = 1, \ldots, r$$

- $\nu_{ML} := \alpha \, U_{ML} \, L_{ML}$

- *Aubry, Holmes, Lumley, Stone, 1988*

  Podvin, Lumley, 1998

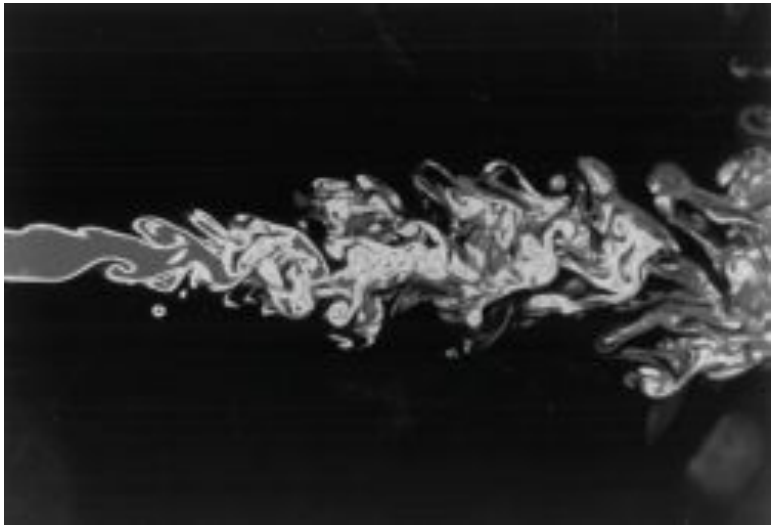
  .....

SMG-POD

- Smagorinsky POD model

$$\left(\frac{\partial \mathbf{u}_r}{\partial t}, \phi_k\right) + ((\mathbf{u}_r \cdot \nabla)\mathbf{u}_r, \phi_k) + \left(\left(\nu_S + \frac{2}{\mathrm{Re}}\right) \mathbb{D}(\mathbf{u}_r), \nabla \phi_k\right) = 0, \qquad k = 1, \ldots, r$$

- $\nu_S := 2(C_S\delta)^2 \|\mathbb{D}(\mathbf{u}_r)\|$
- *Borggaard, Duggleby, Hay, Iliescu, Wang, 2008*
- *Ullmann, Lang, 2010*
- *Borggaard, Iliescu, Wang, 2011*



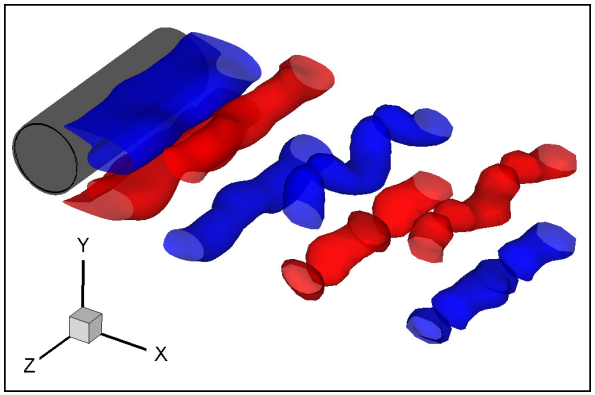
VMS-POD

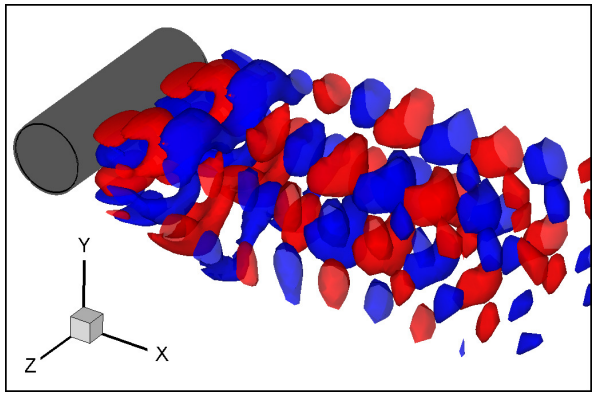
- variational multiscale $\Longleftarrow$ locality energy transfer

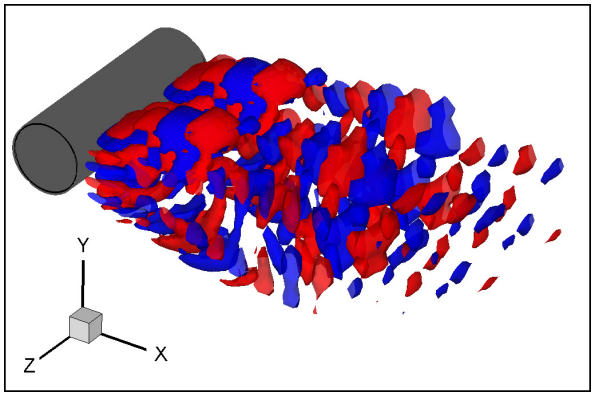
  Hughes et al. 2000; Guermond 1999; Layton 2002

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

  $\mathbf{u}_L \in X_L = \text{span}\{\phi_1, \ldots, \phi_R\}$, $\mathbf{u}_S \in X_S = \text{span}\{\phi_{R+1}, \ldots, \phi_r\}$







VMS-POD

- variational multiscale $\Longleftarrow$ locality energy transfer

  Hughes et al. 2000; Guermond 1999; Layton 2002

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

  $\mathbf{u}_L \in X_L = \text{span}\{\phi_1, \ldots, \phi_R\}$, $\mathbf{u}_S \in X_S = \text{span}\{\phi_{R+1}, \ldots, \phi_r\}$

$$\left(\frac{\partial \mathbf{u}_L}{\partial t}, \phi_k\right) + ((\mathbf{u}_r \cdot \nabla)\mathbf{u}_r, \phi_k) + \left(\frac{2}{\text{Re}}\mathbb{D}(\mathbf{u}_L), \nabla\phi_k\right) = 0, \quad k = 1, \ldots, R$$

$$\left(\frac{\partial \mathbf{u}_S}{\partial t}, \phi_k\right) + ((\mathbf{u}_r \cdot \nabla)\mathbf{u}_r, \phi_k) + \left(\left(\nu_{VMS} + \frac{2}{\text{Re}}\right)\mathbb{D}(\mathbf{u}_S), \nabla\phi_k\right) = 0,$$

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

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

- *Wang, Akhtar, Borggaard, Iliescu, 2012*

- *Iliescu, Wang, 2013, 2014*

DS-POD

- dynamic subgrid-scale model

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

- C_S varies dynamically with x and t

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

$$\frac{\partial \overline{\mathbf{u}}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}}\,\overline{\mathbf{u}}) + \nabla \cdot \overbrace{\cancel{(\overline{\mathbf{u}\,\mathbf{u}} - \overline{\mathbf{u}}\,\overline{\mathbf{u}})}}^{\tau} - Re^{-1}\Delta\overline{\mathbf{u}} = 0$$

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

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

$$\frac{\partial \widetilde{\overline{\mathbf{u}}}}{\partial t} + \nabla \cdot \left(\widetilde{\overline{\mathbf{u}}}\,\widetilde{\overline{\mathbf{u}}}\right) + \nabla \cdot \overbrace{\cancel{\left(\widetilde{\overline{\mathbf{u}\,\mathbf{u}}} - \widetilde{\overline{\mathbf{u}}}\,\widetilde{\overline{\mathbf{u}}}\right)}}^{\mathbf{T}} - Re^{-1}\Delta\widetilde{\overline{\mathbf{u}}} = 0$$

$$\mathbf{T} := -2(C_s\widetilde{\delta})^2 \|\mathbb{D}(\widetilde{\overline{\mathbf{u}}})\| \mathbb{D}(\widetilde{\overline{\mathbf{u}}})$$

- $\widetilde{\overline{\mathbf{u}}\,\overline{\mathbf{u}}} - \widetilde{\overline{\mathbf{u}}}\,\widetilde{\overline{\mathbf{u}}} = \left(\widetilde{\overline{\mathbf{u}\,\mathbf{u}}} - \widetilde{\overline{\mathbf{u}}}\,\widetilde{\overline{\mathbf{u}}}\right) - \left(\widetilde{\overline{\mathbf{u}\,\mathbf{u}}} - \widetilde{\overline{\mathbf{u}}}\,\widetilde{\overline{\mathbf{u}}}\right) = \mathbf{T} - \widetilde{\tau}$

DS-POD

- $C_S(x,t)$ determined dynamically

$$Q = 2(\delta)^2 \|\widetilde{\mathbb{D}(\overline{\mathbf{u}})}\| \mathbb{D}(\overline{\mathbf{u}}) - 2(\widetilde{\delta})^2 \|\mathbb{D}(\widetilde{\overline{\mathbf{u}}})\| \mathbb{D}(\widetilde{\overline{\mathbf{u}}})$$

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

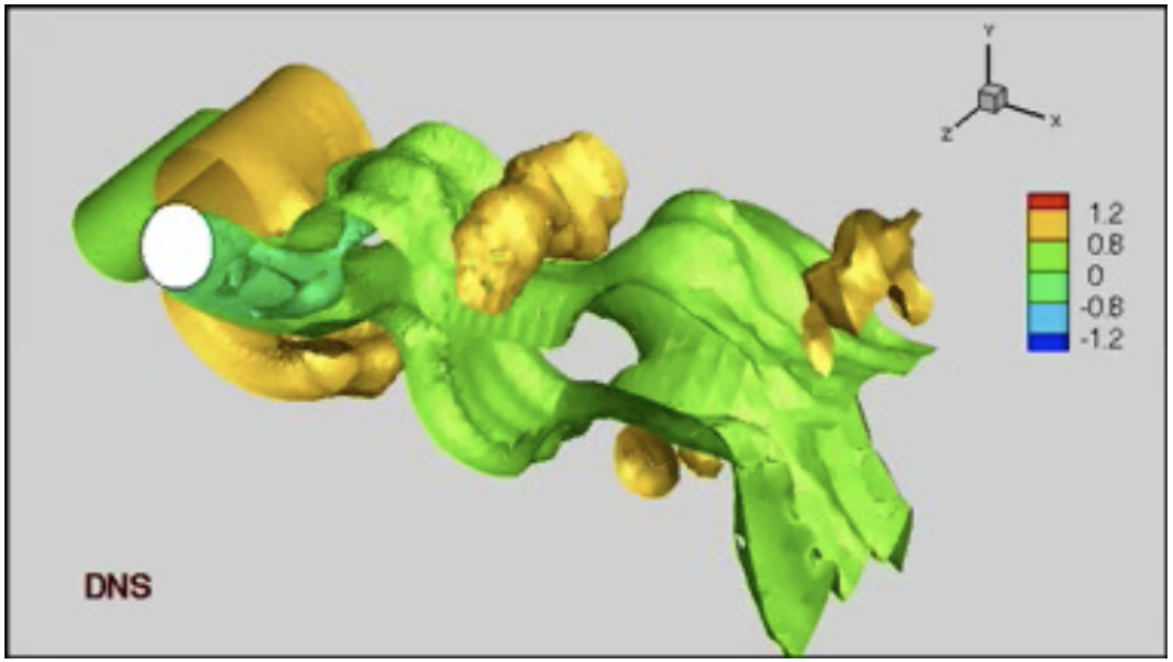
- DS-POD ROM

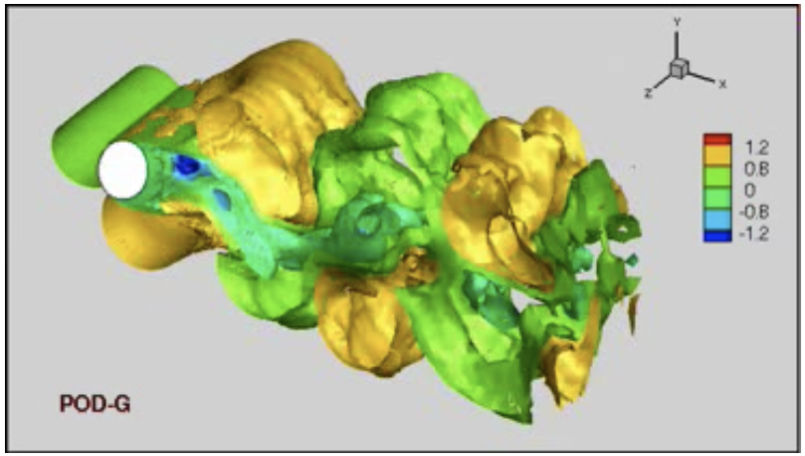
$$\left(\frac{\partial \mathbf{u}_r}{\partial t}, \phi_k\right) + ((\mathbf{u}_r \cdot \nabla)\mathbf{u}_r, \phi_k) + \left(\left(\nu_{DS} + \frac{2}{Re}\right) \mathbb{D}(\mathbf{u}_r), \nabla \phi_k\right) = 0, \qquad k = 1, \ldots, r$$

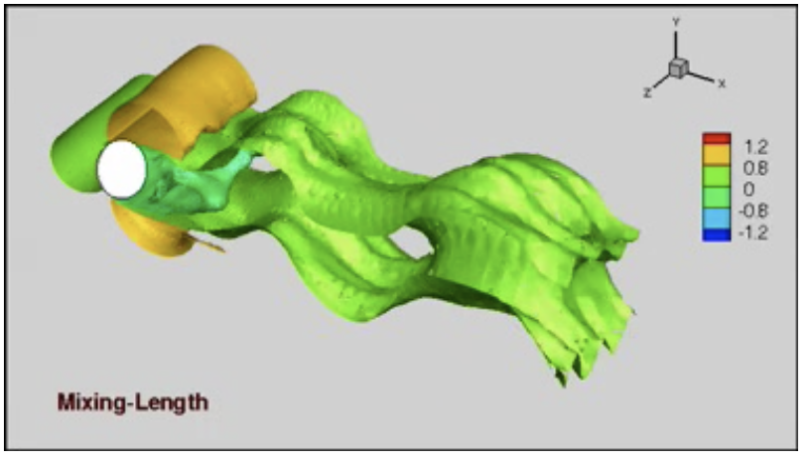
- $\nu_{DS} := 2\,(C_S(x,t)\,\delta)^2\,\|\mathbb{D}(\mathbf{u}_r)\|$
- *Wang, Akhtar, Borggaard, Iliescu, 2012*

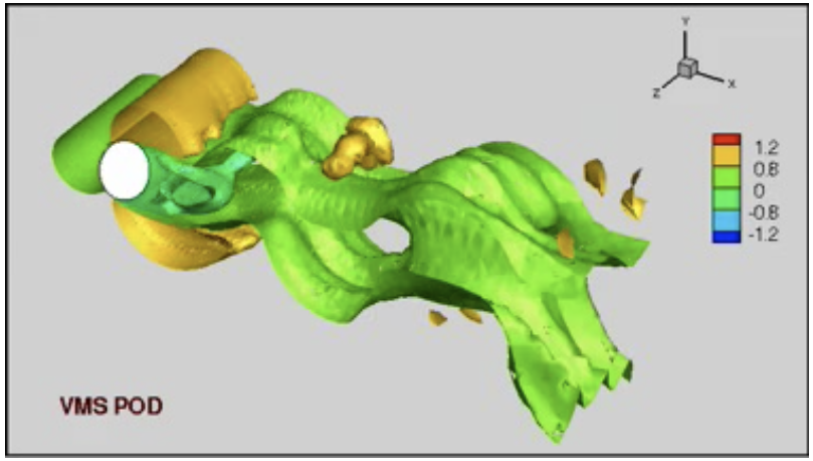
Numerical Experiments

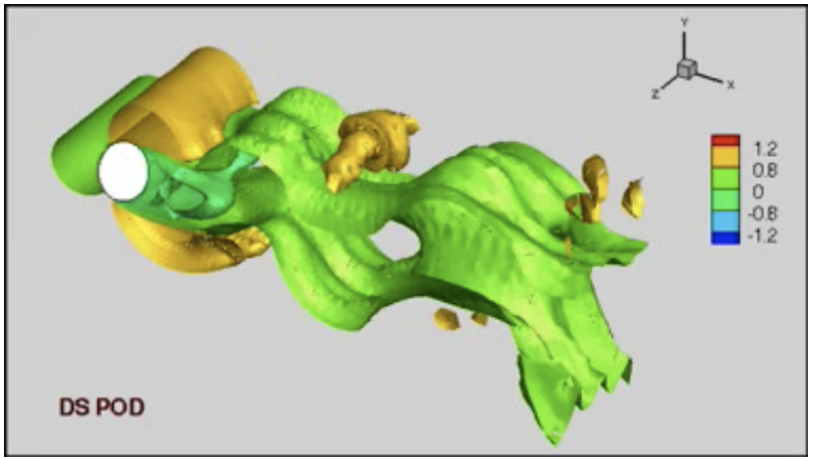
3D turbulent flow past a cylinder $\boxed{\mathrm{Re} = 1000}$



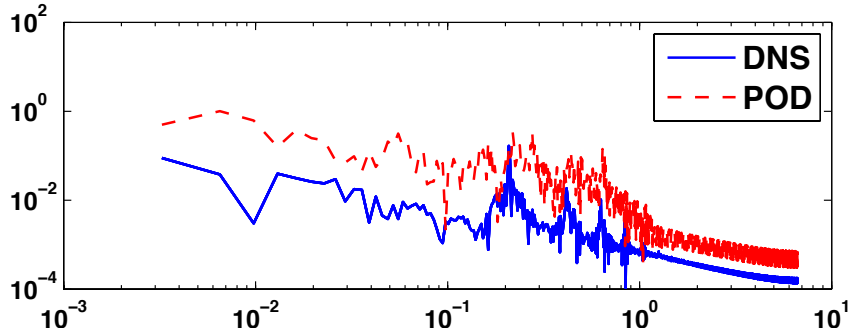

1.2
0.8
0
-0.8
-1.2
POD-G


1.2
0.8
0
-0.8
-1.2
Mixing-Length

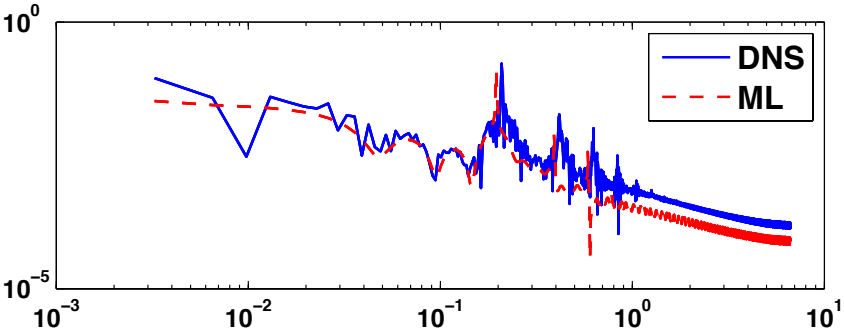

1.2
0.8
0
-0.8
-1.2
VMS POD

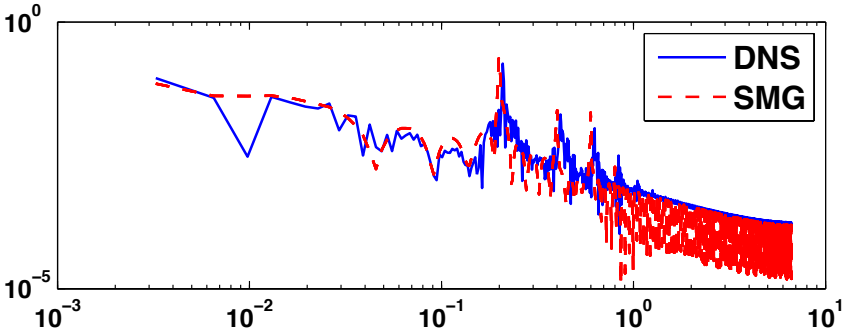

1.2
0.8
0
-0.8
-1.2
DS POD

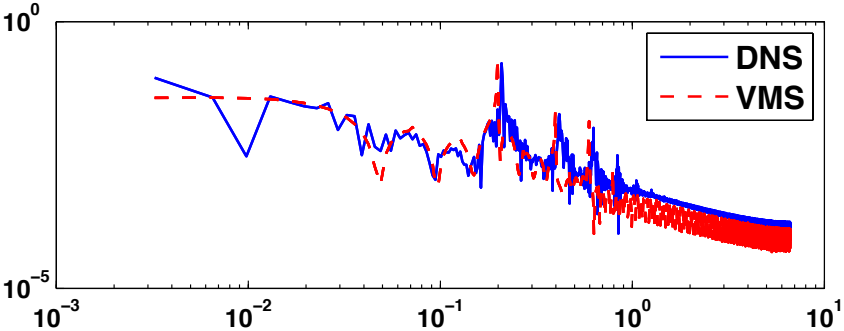
Energy Spectrum

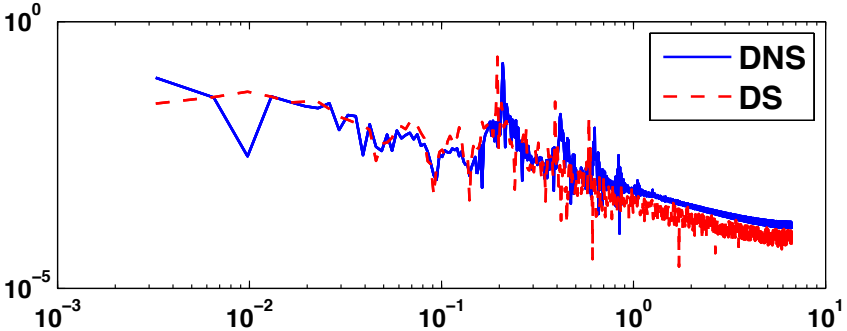


VMS-POD; DS-POD – accurate









AD-POD

- *structural* closure $\nabla \cdot \boldsymbol{\tau} = \nabla \cdot (\overline{\mathbf{u}\,\mathbf{u}} - \overline{\mathbf{u}}\,\overline{\mathbf{u}}) \approx \nabla \cdot (\overline{\mathbf{u}^*\,\mathbf{u}^*} - \overline{\mathbf{u}}\,\overline{\mathbf{u}})$

  Stolz, Adam, 1999

- deconvolution

  given $\overline{\mathbf{u}} := G\,\mathbf{u}$

  find $\mathbf{u}$

  image processing, inverse problems

- exact deconvolution $\mathbf{u}^{ED} = G^{-1}\,\overline{\mathbf{u}}$

  very bad idea

  notoriously ill-posed: noise amplification

- approximate deconvolution $\mathbf{u}^{AD} \approx \mathbf{u}^{ED} = G^{-1}\,\overline{\mathbf{u}}$

  Lavrentiev regularization $\boxed{\mathbf{u}^{AD-L} = (G + \mu\, I)^{-1}\,\overline{\mathbf{u}}}$ Bertero, Boccacci, 1998

AD-POD

- approximate deconvolution for closure

  define $\overline{\mathbf{u}}_r \in X^r$ such that $(I - \alpha^2 \Delta)\overline{\mathbf{u}}_r = \mathbf{u} \quad \leftrightarrow \quad \overline{\mathbf{u}}_r = G\,\mathbf{u}$

  denote $\mathbf{w}_r := \overline{\mathbf{u}}_r$

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

- AD-POD ROM

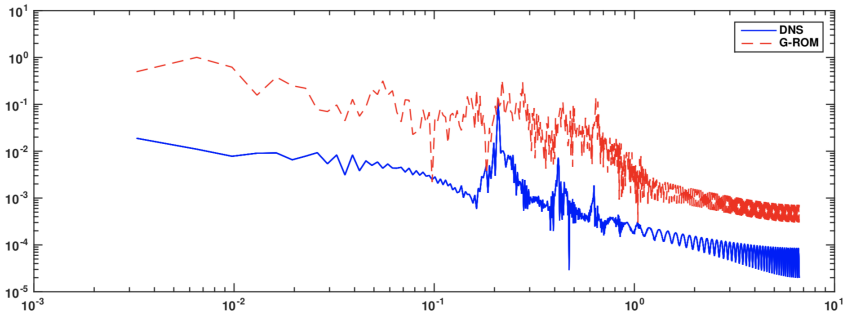
$$\boxed{\left(\frac{\partial \mathbf{w}_r}{\partial t}, \phi_k\right) + \frac{2}{\mathrm{Re}}\,(\mathbb{D}(\mathbf{w}_r), \nabla\phi_k) + \left(\overline{\mathbf{w}_r^{AD-L} \cdot \nabla \mathbf{w}_r^{AD-L}}, \phi_k\right) = 0}$$

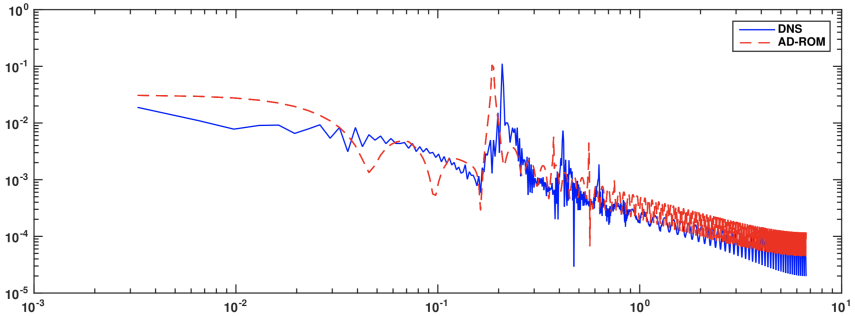
- *Xie, Wells, Wang, Iliescu, 2017*

Numerical Experiments

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

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





Outline

Basis Generation

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

  snapshot data huge, distributed over multiprocessors

  $\rightarrow$ POD basis, typically left singular vectors of data

  generation expensive in computation and communication

  $\rightarrow$ **partitioned methods of snapshots** *W., 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)

$$\mathbf{S} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{T}} \quad \rightarrow \quad \phi_j = \mathbf{U}(\cdot, j)$$

complexity: $\mathcal{O}(n^2 m + nm^2 + m^3)$ communication: nm

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

$$\mathbf{S}^{\mathsf{T}}\mathbf{S}\,\mathbf{z}_j = \lambda_j\,\mathbf{z}_j, \text{for } j = 1, \ldots, r \quad \rightarrow \quad \phi_j = \frac{1}{\sqrt{\lambda_j}} \sum_{\ell=1}^{m} (\mathbf{z}_j)_\ell\, \mathbf{S}_{(\cdot,\ell)}$$

complexity: $\mathcal{O}(nm^2 + rnm + m^3)$ communication: nm

POD Basis Generation

- Partitioned Singular Value Decomposition (PSVD)

  given local data $\mathbf{S}_i$, for $i = 1, \ldots, p$ *Beattie, Borggaard, Gugercin and Iliescu, 2006*

  $[\mathbf{U}_i, \mathbf{\Sigma}_i, \mathbf{V}_i] = \text{svd}(\mathbf{S}_i)$ <u>locally</u>, $\boldsymbol{\mathcal{V}} = \left[\mathbf{V}_1^q, \mathbf{V}_2^q, \ldots, \mathbf{V}_p^q\right]$;

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

  calculate $\mathbf{S}\widehat{\mathbf{V}}^r = [\mathbf{S}_1\widehat{\mathbf{V}}^r; \ldots; \mathbf{S}_p\widehat{\mathbf{V}}^r]$ <u>locally</u>;

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

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

  communication: $nr + mr + mpq$

Partitioned Method of Snapshots (I)

- $\mathbf{S}^\intercal\mathbf{S} = \sum_{i=1}^{p} \mathbf{S}_i^\intercal \mathbf{S}_i$

Algorithm 1 Partitioned Method of Snapshots (PMOS)

Let $\mathbf{S}_i$ be local data on the i-th processor.
for $i = 1$ to p **do**
  Evaluate $\mathbf{D}_i = \mathbf{S}_i^\intercal \mathbf{S}_i$ locally
end for
do $\mathbf{D} = \sum_{i=1}^{p} \mathbf{D}_i$
do $[\mathbf{V}, \mathbf{\Sigma}]$ = eig($\mathbf{D}$)
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 $\phi_j^i = \frac{1}{\sqrt{\lambda_j}} \mathbf{S}_i \mathbf{V}_{(\cdot,j)}$ locally
  end for
end for

Partitioned Method of Snapshots (II)

- $\mathbf{S}^\intercal\mathbf{S} = \sum_{i=1}^{p} \mathbf{S}_i^\intercal \mathbf{S}_i = \sum_{i=1}^{p} \mathbf{V}_i \mathbf{\Sigma}_i^2 \mathbf{V}_i^\intercal$

$$= \left[\mathbf{V}_1\mathbf{\Sigma}_1^\intercal, \mathbf{V}_2\mathbf{\Sigma}_2^\intercal, \ldots, \mathbf{V}_p\mathbf{\Sigma}_p^\intercal\right] \begin{bmatrix} \mathbf{\Sigma}_1\mathbf{V}_1^\intercal \\ \mathbf{\Sigma}_2\mathbf{V}_2^\intercal \\ \vdots \\ \mathbf{\Sigma}_p\mathbf{V}_p^\intercal \end{bmatrix} = \mathbf{W}\mathbf{W}^\intercal$$

- $\mathbf{W} \approx \mathbf{W}^r = \left[\mathbf{V}_1^{r_1}(\mathbf{\Sigma}_1^{r_1})^\intercal, \mathbf{V}_2^{r_2}(\mathbf{\Sigma}_2^{r_2})^\intercal, \ldots, \mathbf{V}_p^{r_p}(\mathbf{\Sigma}_p^{r_p})^\intercal\right]$
- $\mathbf{W}^r = \mathbf{X}\mathbf{\Lambda}\mathbf{Y}^\intercal$
- $\mathbf{V} \approx \mathbf{X}$, and $\mathbf{\Sigma} \approx \mathbf{\Lambda}$

Partitioned Method of Snapshots (II)

Algorithm 2 Approximate Partitioned Method of Snapshots (APMOS)

Let $\mathbf{S}_i$ be local data on the i-th processor.
for $i = 1$ to p **do**
  $[\mathbf{U}_i, \mathbf{\Sigma}_i, \mathbf{V}_i]$ = svd($\mathbf{S}_i$) locally
  select r_i, s.t., $\sigma_i^{r_i+1} < \epsilon_0$
  take $\mathbf{V}_i^{r_i} = \mathbf{V}_{i,(\cdot,1:r_i)}$ and $\mathbf{\Sigma}_i^{r_i} = \mathbf{\Sigma}_{i,(1:r_i,1:r_i)}$
end for
assemble $\mathbf{W}^r = \left[\mathbf{V}_1^{r_1}(\mathbf{\Sigma}_1^{r_1})^\intercal, \ldots, \mathbf{V}_p^{r_p}(\mathbf{\Sigma}_p^{r_p})^\intercal\right]$
do $[\mathbf{X}, \mathbf{\Lambda}, \mathbf{Y}]$ = svd($\mathbf{W}^r$)
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 $\widetilde{\phi}_j^i = \frac{1}{\sqrt{\lambda_j}} \mathbf{S}_i \mathbf{X}_{(\cdot,j)}$ locally
  end for
end for

Comparison

Method	Complexity (flops)	Communication
PMOS	$\underline{\mathcal{O}(\sum_{i=1}^{p} n_i m^2 + \sum_{i=1}^{p} n_i r m)}$ $+ \mathcal{O}(m^3)$	$m^2 p + mr$
APMOS	$\underline{\mathcal{O}\left(\sum_{i=1}^{p}(n_i^2 m + n_i m^2 + m^3 + n_i r m)\right)}$ $+ \mathcal{O}\left((\sum_{i=1}^{p} r_i)^3\right)$	$m \sum_{i=1}^{p} r_i + mr + r$

* Underline represents operations to be executed in parallel.

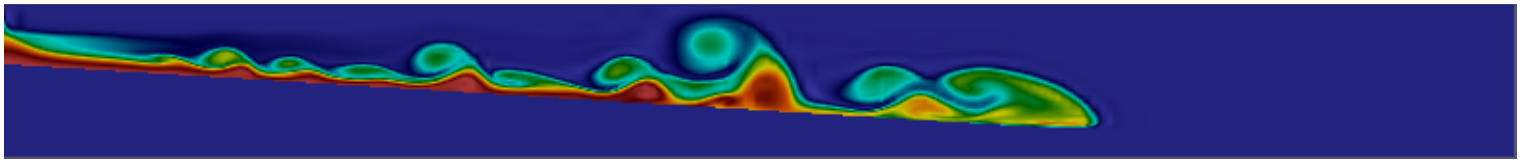
Theorem

Let λ_j be the j-th largest eigenvalue of $\mathbf{W}^r(\mathbf{W}^r)^\intercal$ and $\overline{\lambda}_j$ the j-th largest eigenvalue of $\mathbf{W}(\mathbf{W})^\intercal$. 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)



$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} - \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \mathrm{Ra}\, T\mathbf{k} &= 0, \\ \nabla \cdot \mathbf{u} &= 0, \\ \frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T - \mathrm{Pr}^{-1} \Delta T &= 0, \end{cases}$$

- driven by velocity and temperature forcing profiles at the inlet
- $\mathrm{Ra} = 5 \times 10^6$ and $\mathrm{Pr} = 7$
- horizontal length of $L = 10km$
- depth of the water column ranges from $K = 400m$ at $x = 0$ to $H = 1000m$ at $x = 10km$ over a constant slope ($\theta = 3.5°$)
- 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** – $14,400 \times 900$

Numerical Experiment

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

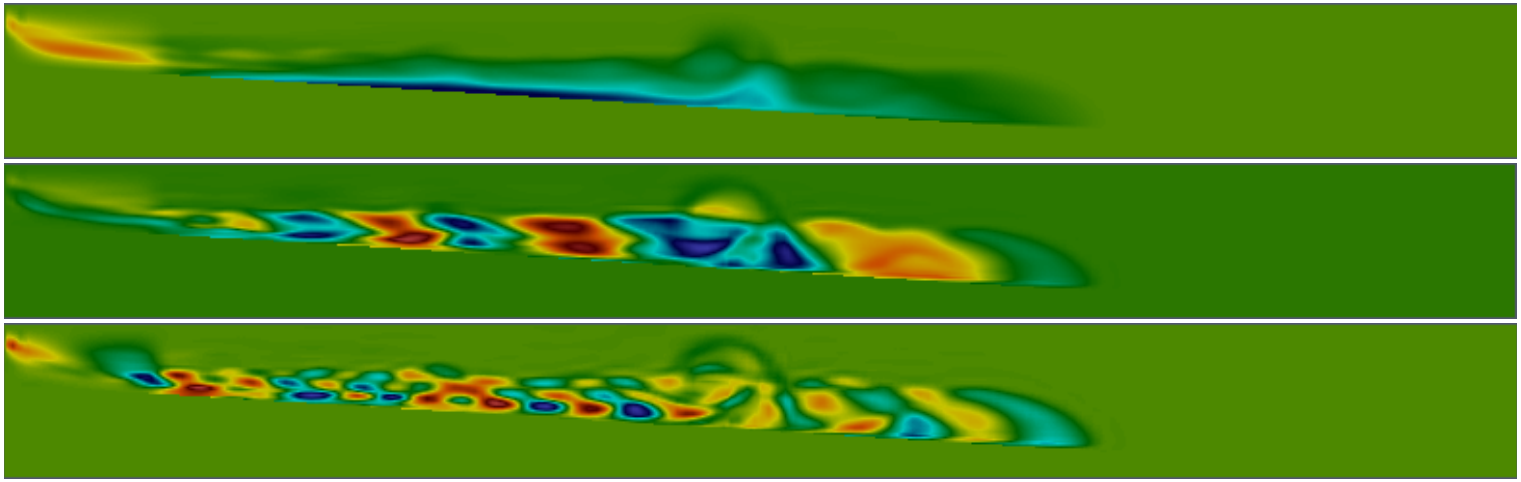


Figure 1: Gravity current. From the top: the first, fifth and tenth POD basis functions of the temperature generated by APMOS with $\epsilon_0 = 0.1$ and $p = 10$.

Numerical Experiment

- APMOS with $p = 10$ and different values of ϵ_0

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

error	$\epsilon_0 = 0.1$	$\epsilon_0 = 0.01$	$\epsilon_0 = 0.001$
$\max_{j\in[1,10]} \lvert\lambda_j - \overline{\lambda}_j\rvert$	3.9457e-03	2.3582e-05	1.3043e-07

Nonlinear Closure

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = \mathcal{A} + \widetilde{\mathcal{A}}\big(\mathbf{u}_r(\cdot, t)\big) + \Big(\mathcal{B} + \widetilde{\mathcal{B}}\big(\mathbf{u}_r(\cdot, t)\big)\Big)\mathbf{a} + \mathcal{C}(\mathbf{a} \otimes \mathbf{a})$$

- $\widetilde{\mathcal{A}}$, $\widetilde{\mathcal{B}}$ nonlinear

General Nonlinear ROMs

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = \mathcal{A} + \mathcal{B}\mathbf{a} + (N(u_r), \Phi)$$

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

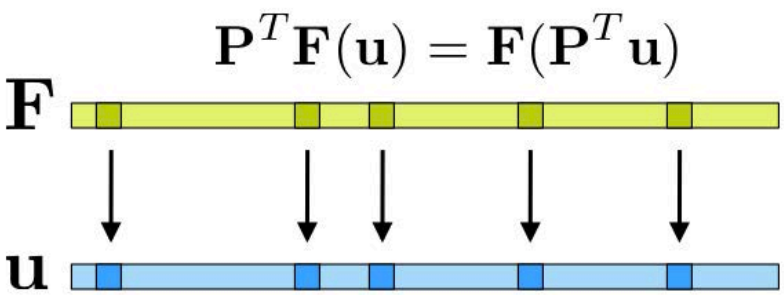
  $\mathcal{N}$ needs to be assembled at each time step/iteration

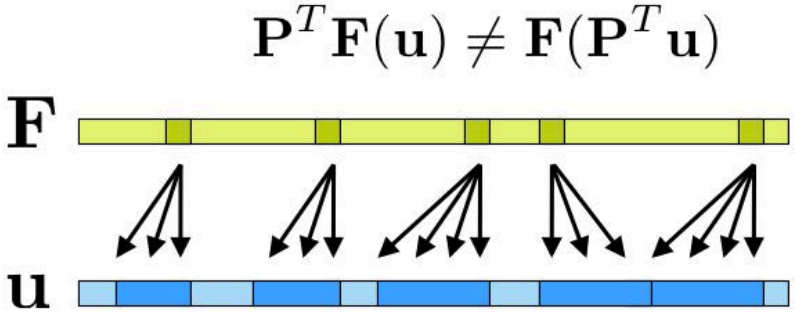
  computational complexity depends on $n_e n_q$ of full-order model
- hyper-reduction, e.g., DEIM/QDEIM

Chaturantabut and Sorensen, 2010; Drmac and Gugercin 2015

Difficulties of DEIM in CG

- online simulations ($\mathbf{P}^\intercal\mathbf{F}(\mathbf{u})$) involve integrations in FE discretization

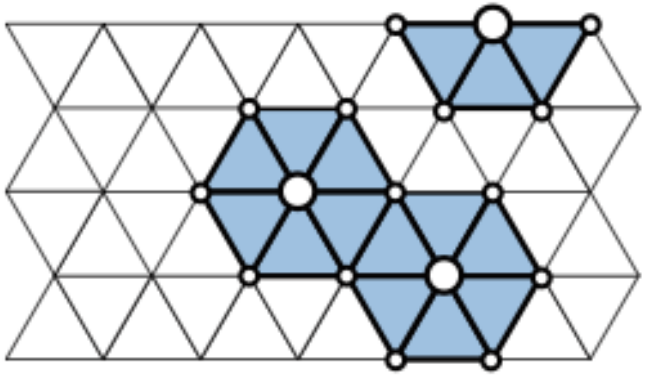


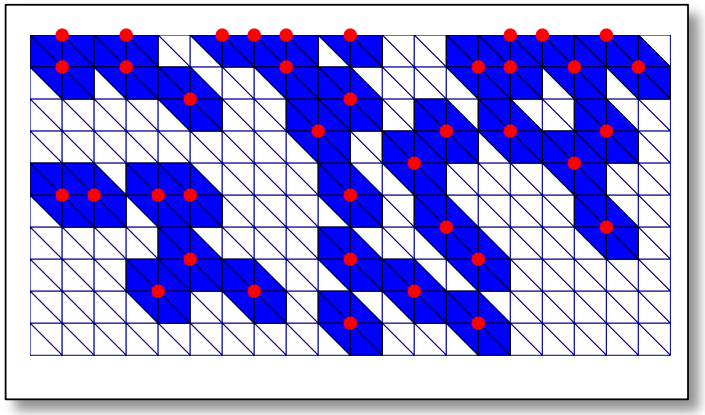


Difficulties of DEIM in CG

- online simulations ($\mathbf{P}^\intercal \mathbf{F}(\mathbf{u})$) involve integrations in FE discretization

regular FE





40 pts, 45% elements

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) $\mathcal{N}_{FEIC} = \left(\mathcal{I}^h N(u_r), \phi\right) = \Phi^\intercal \mathbf{M}^h N\left(\Phi \mathbf{a}(t)\right)$

$\Phi^\intercal \mathbf{M}^h$ pre-computable, $N\left(\Phi \mathbf{a}(t)\right)$ evaluated at FE nodes

(POD-FEIC-DEIM) $\mathcal{N}_{FEIC-DEIM} = \Phi^\intercal \mathbf{M}^h \Psi(\mathbf{P}^\intercal \Psi)^{-1} \mathbf{P}^\intercal \mathbf{N}\left(\Phi \mathbf{a}(t)\right)$

$[\Phi^\intercal \mathbf{M}^h \Psi(\mathbf{P}^\intercal \Psi)^{-1}]_{r \times p}$ precomputable

online simulation $N\left(\Phi \mathbf{a}(t)\right)$ evaluated at selected DEIM pts, $\mathbf{P}^\intercal \mathbf{N}\left(\Phi \mathbf{a}(t)\right)$

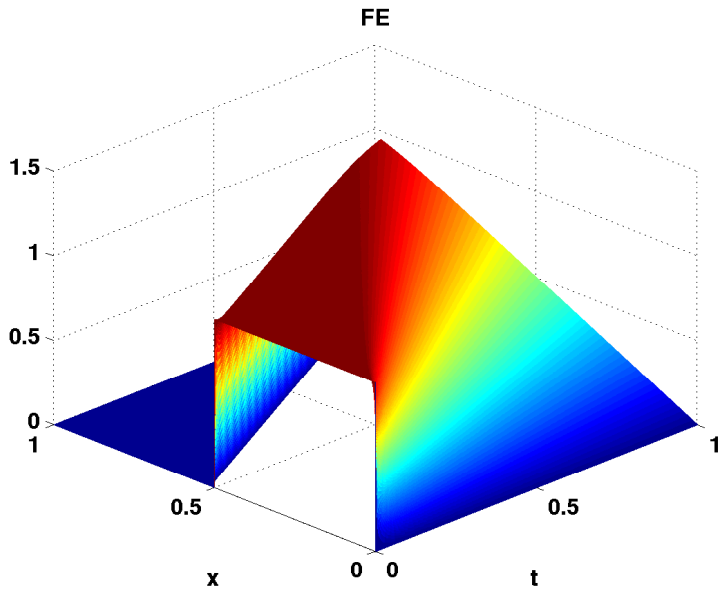
- computational complexity at each step/iteration is $\mathcal{O}\left(4rp + \varrho(p)\right)$ flops *W., 2015*

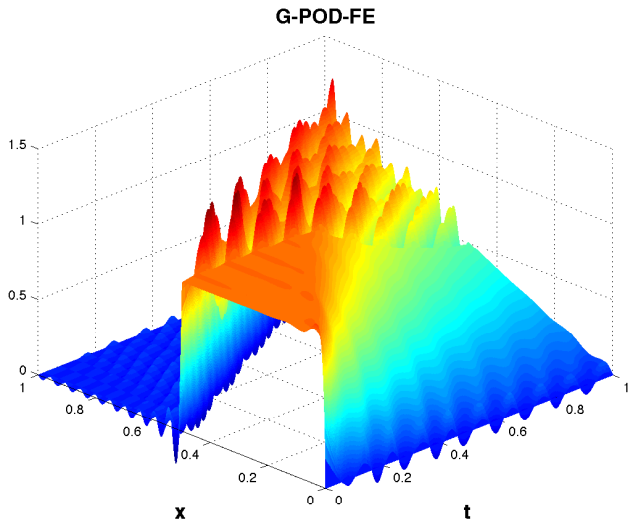
No need for this in the DG setting

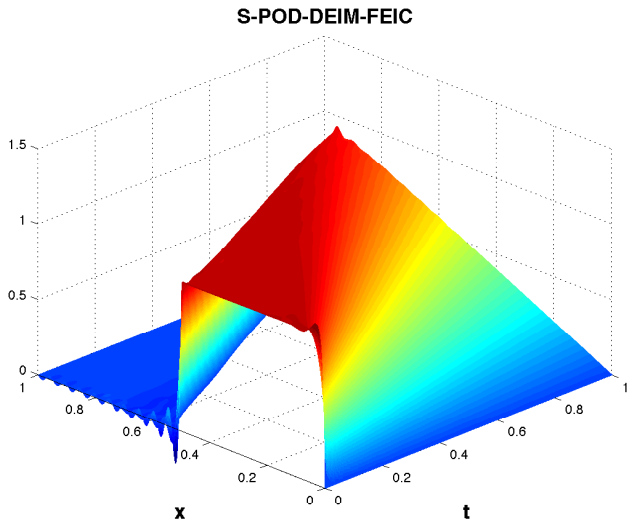
POD-FEIC-DEIM for Nonlinear Closure

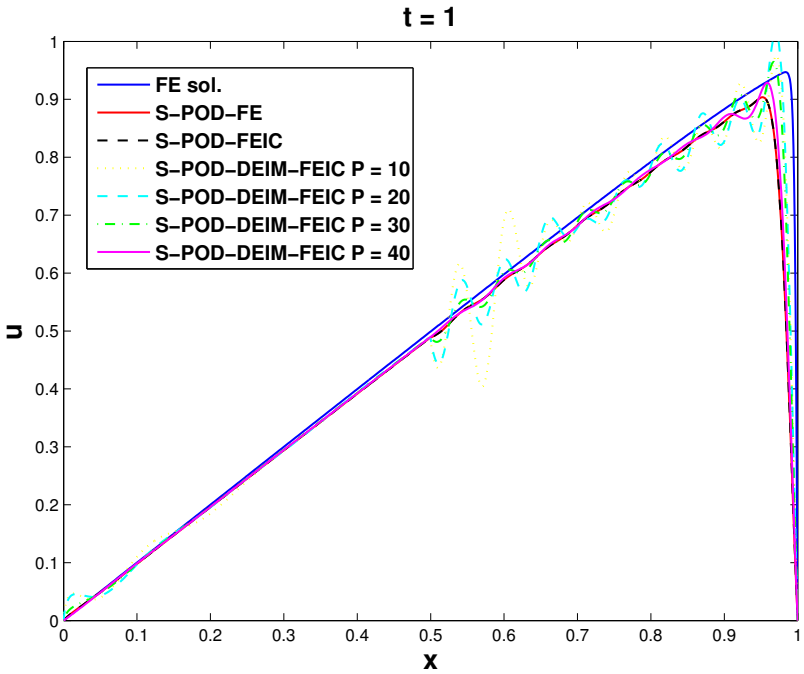
$$\begin{cases} u_t - \nu\, u_{xx} + u\, u_x = 0, & \text{in } \Omega \times [0, T], \\ u(x,t) = 0, & \text{on } \partial\Omega \times [0, T], \\ u(x,0) = u_0(x), & \text{in } \Omega. \end{cases}$$

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









- $r = 20$

  $p = 40$ in DEIM

  speed-up-factor $>$ 100

model	CPU time
S-POD-FE	119.50
S-POD-FEIC	2.05
S-POD-FEIC-DEIM	1.00

Outline

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;* $\cdots$
  - seeking online adaptivity: *Peherstorfer and Willcox 2015; Peherstorfer 2019; Carlberg, 2015; Etter and Carlberg, 2019;* $\cdots$

Thank You!