A Novel Method for Generating Reduced-order Models for Transport Phenomena Using Proper Orthogonal Decomposition

Elizabeth Krath, Forrest Carpenter, Paul Cizmas
Texas A&M University

August 2-4, 2022
Workshop on Multiphase Flow Science

Contributors:

Tao Yuan, Thomas Brenner, Raymond Fontenot, Sohail Reddy

Outline

- Motivation and Current Limitations
- Reduced-order Models
  - Standard Proper Orthogonal Decomposition (POD) Method
  - POD Method with Dynamic Basis Functions
  - Constrained POD
  - ς-variable POD Method
- Results
- Conclusions and Future Work

Motivation

- Reduce computational cost while improving model accuracy
- Current speedups for MFIX approx. 120
- Current speedups for single-phase flows modeled with RANS approx. 50,000

Models for Unsteady Transport Phenomena

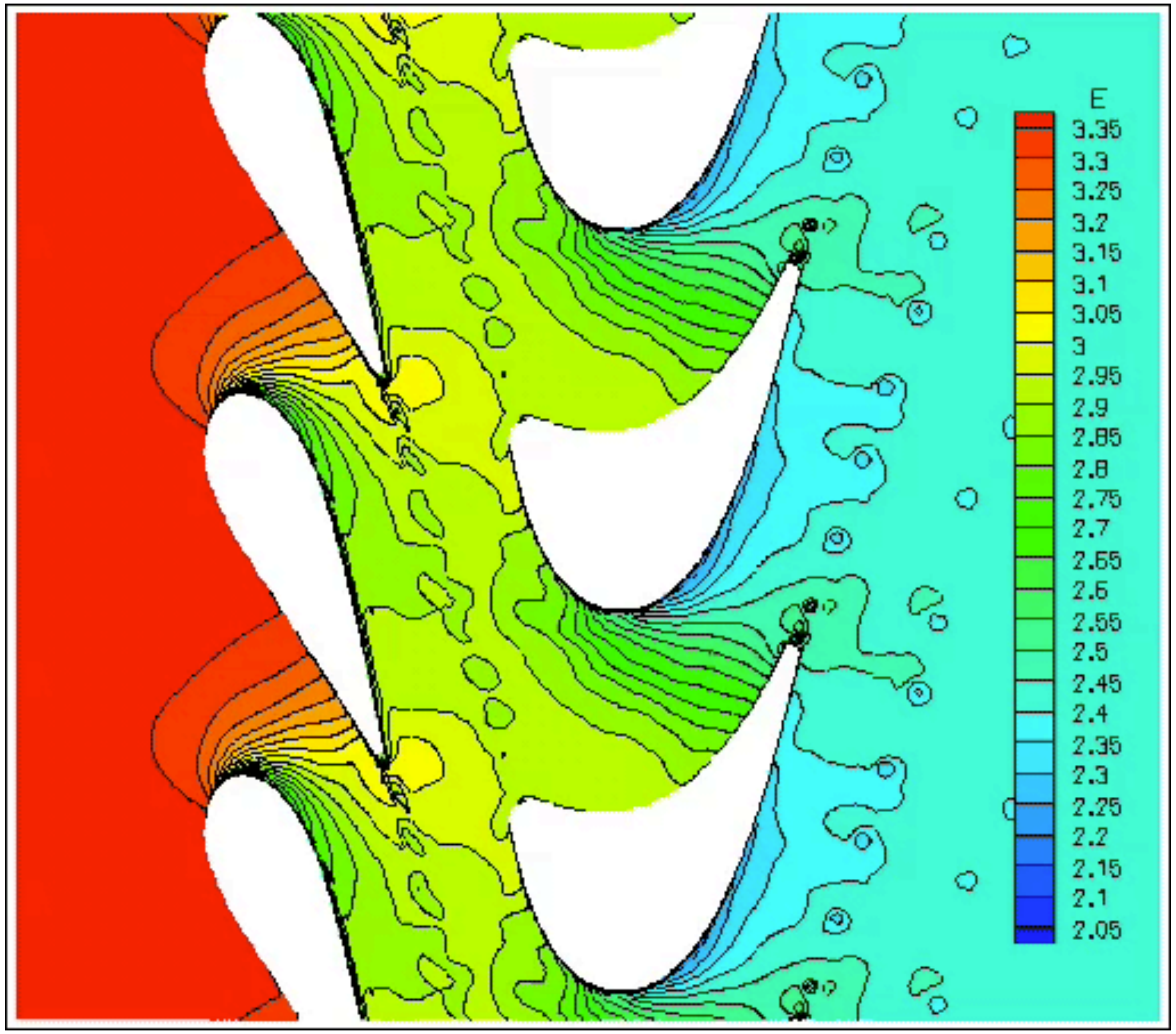
- Time linearization (Hall, 1993; Cizmas & Hall, 1995; Clark & Hall, 1999)
  - Small (linear) dynamic perturbation about a (nonlinear) mean steady flow $a(x,t) = a_0(x) + a_1(x)e^{j\omega t}$
- Harmonic balance (Kuo et al., 1972; Hwang & Lan, 1989; Greco et al. 1997; Hall et al. 2002)
  - Expand solution in terms of a Fourier series in time; transforms from time domain to frequency domain; single solution for each harmonic frequency included in model
- Reduced-order model (Dowell et al., 1999; Cizmas & Palacios, 2003; Amsallem & Farhat,2008)
  - Use dominant spatial modes to represent the flow
- Transfer functions and Volterra series (Silva, 1997; Beran & Silva, 2001; Lucia et al., 2004)

POD-Based Reduced-Order Model

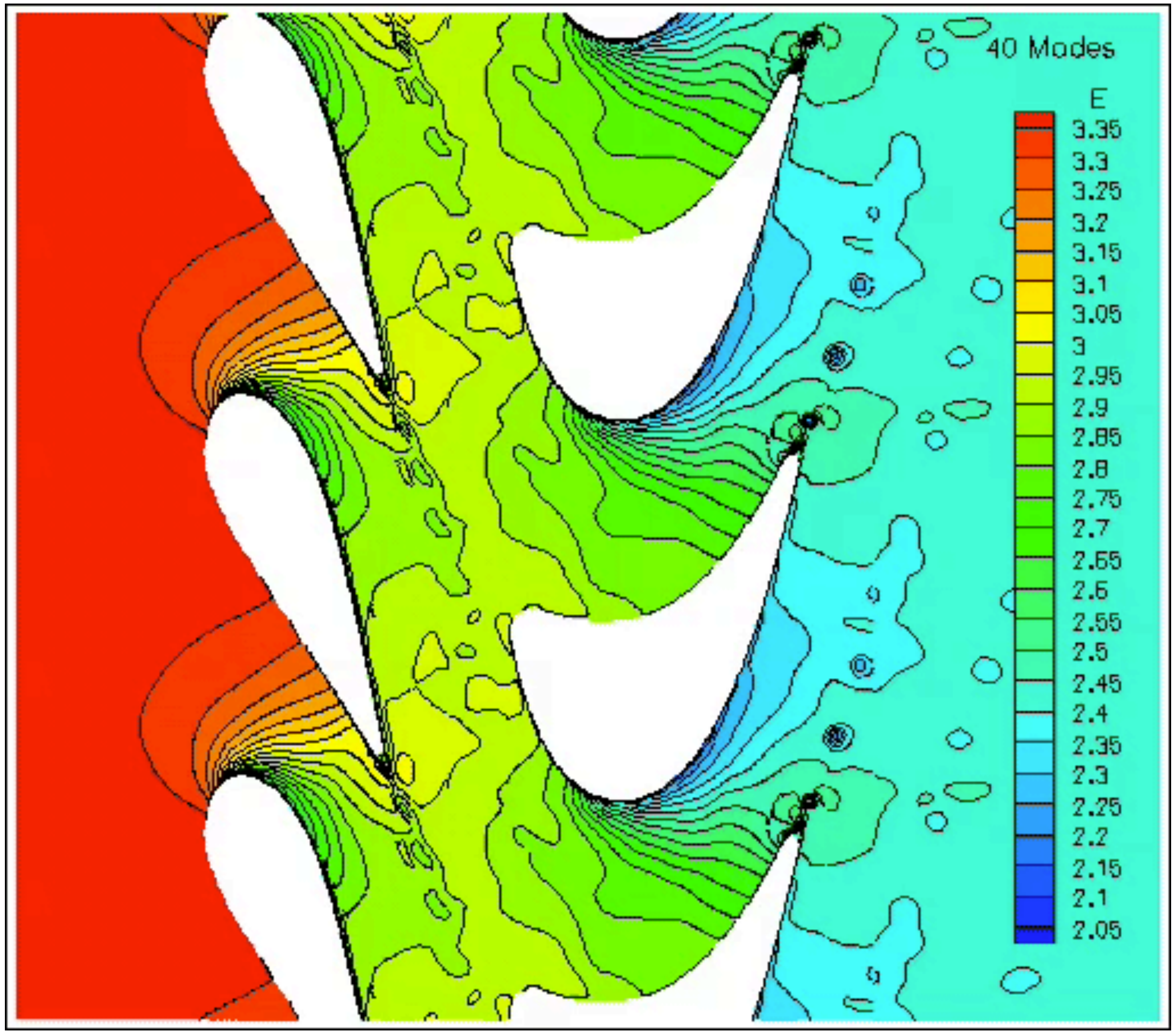
POD for Turbomachinery Aeroelastic Analysis

A reduced-order model is not necessarily a low-fidelity solution!

Full-Order Model

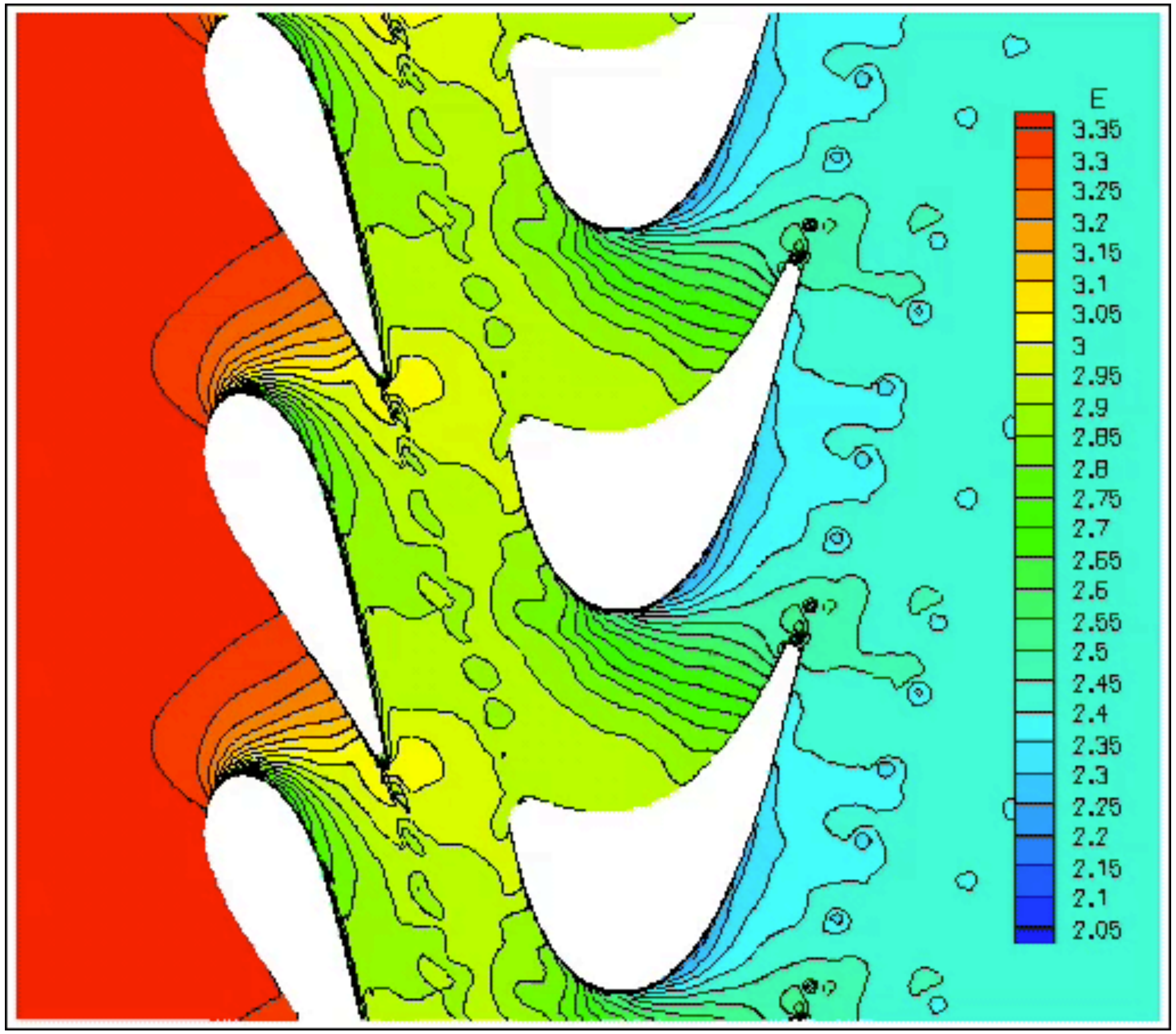


Reduced-Order Model, POD 40 modes

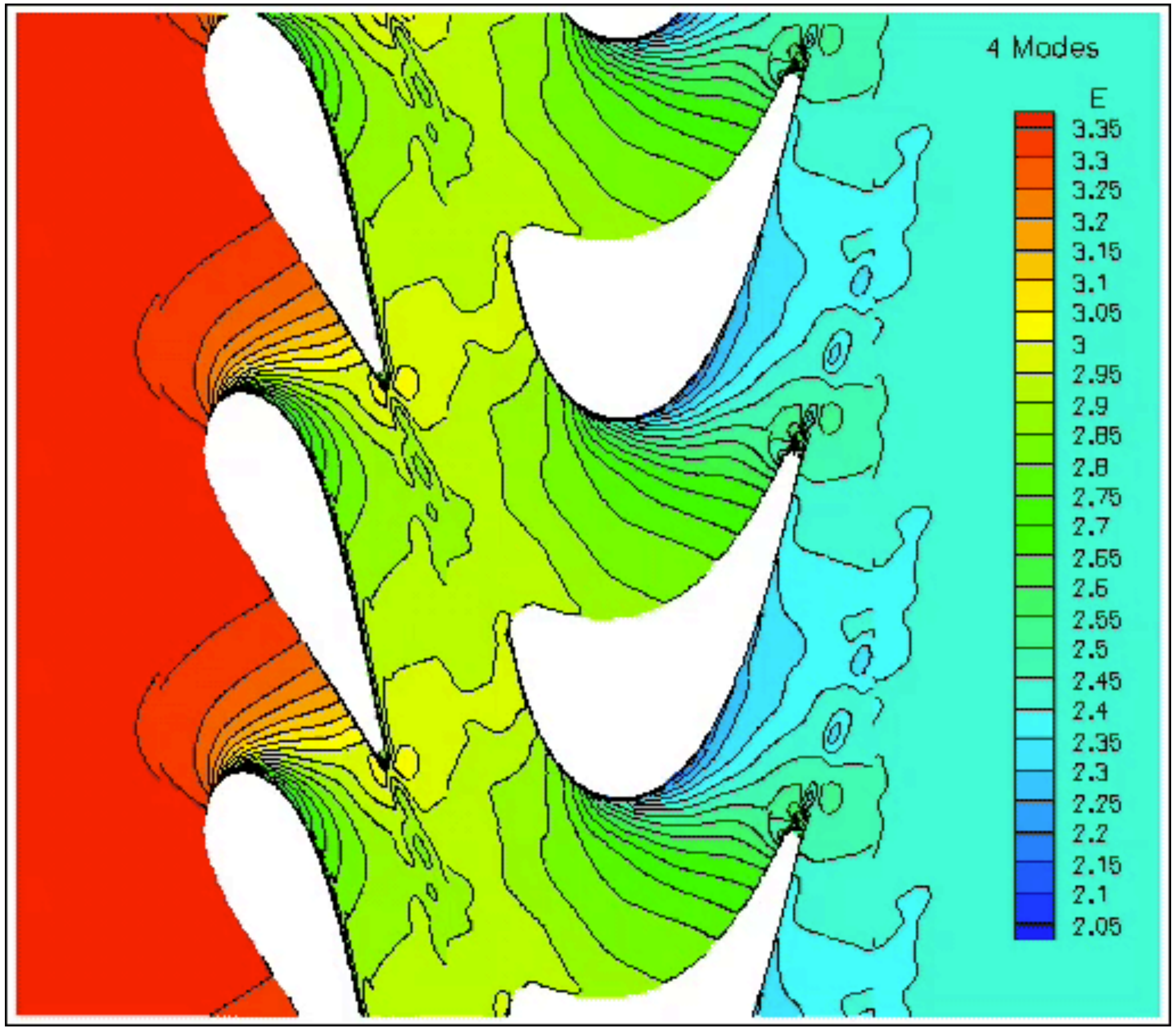


POD for Turbomachinery Aeroelastic Analysis

Full-Order Model

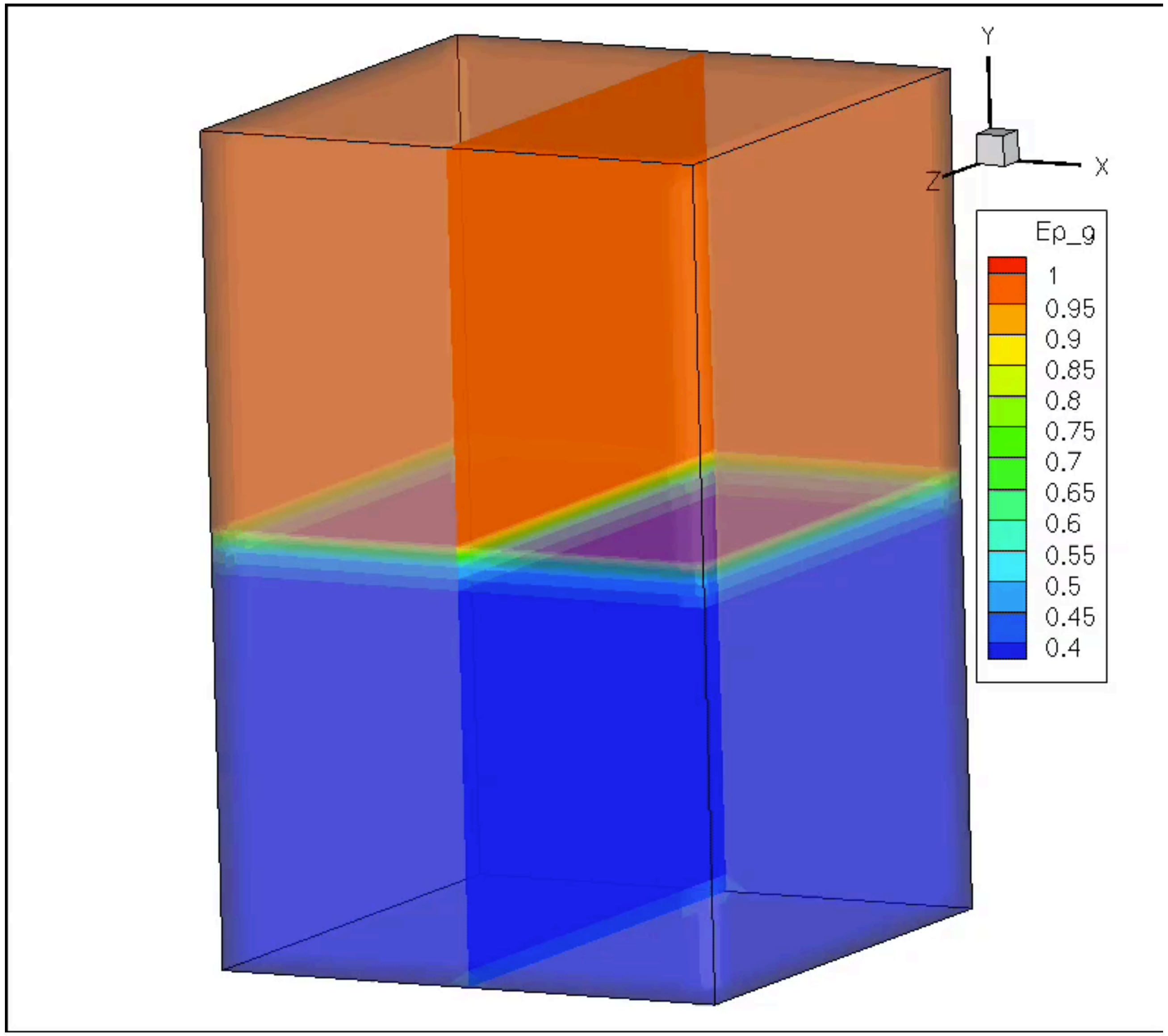


Reduced-Order Model, POD 4 modes

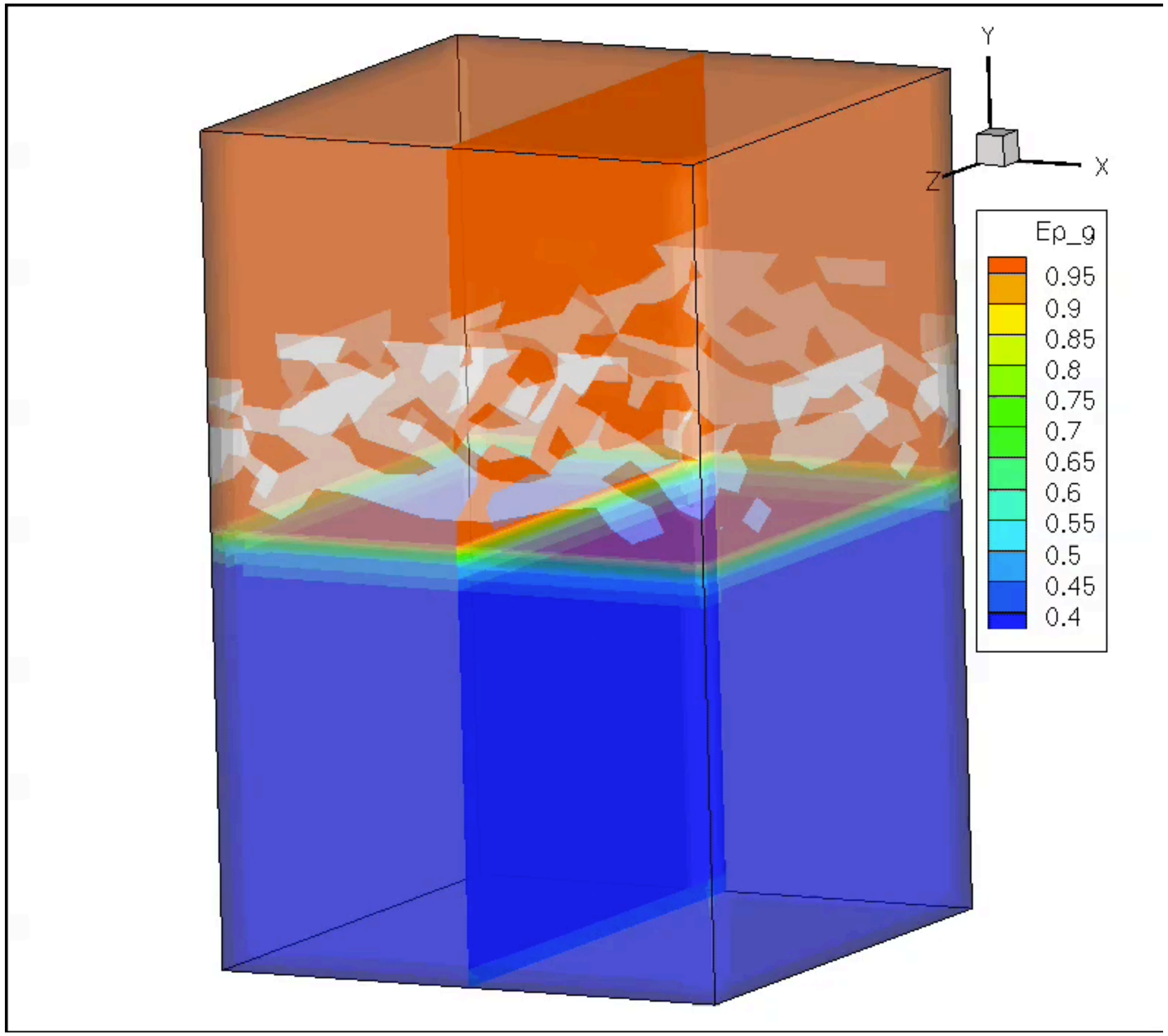


Void Fraction, ϵ_g

Full-order model



Reduced-order model



Proper Orthogonal Decomposition (POD) Method

- POD is also known as Singular Value Decomposition, Karhunen-Loeve Decomposition, Principal Components Analysis, and Singular Systems Analysis
- Provides optimal basis for modal decomposition of a data set
- Extracts key **spatial** features from physical systems with spatial and temporal characteristics
- Reduces a large set of governing PDEs to a much smaller set of ODEs

POD Method

- Extracts:
  - time-independent orthonormal basis functions $\Phi_k(x)$
  - time-dependent orthonormal amplitude coefficients $\alpha_k(t_i)$ such that the reconstruction

$$u(\mathbf{x}, t_i) = \sum_{k=1}^{M} \alpha_k(t_i)\varphi_k(\mathbf{x}), \qquad i = 1, \ldots, M$$

is optimal in the sense that the average least square truncation error

$$\varepsilon_m = \left\langle \left\| u(\mathbf{x}, t_i) - \sum_{k=1}^{m} \alpha_k(t_i)\, \varphi_k(\mathbf{x}) \right\|^2 \right\rangle \qquad (1)$$

is a minimum for any given number $m \leq M$ of basis functions over all possible sets of orthogonal functions

POD Method

- Optimal property (1) reduces to:

$$\int_D \langle u(\mathbf{x}) u^*(\mathbf{y}) \rangle \varphi(\mathbf{y}) dy = \lambda \varphi(\mathbf{x}) \qquad (2)$$

$\{\Phi_k\}$ are eigenfunctions of integral equation (2), whose kernel is the averaged autocorrelation function

$$\langle u(\mathbf{x}) u^*(\mathbf{y}) \rangle \equiv R(\mathbf{x}, \mathbf{y}) \qquad (3)$$

- For a finite-dimensional case, (3) replaced by tensor product matrix

$$\overline{\overline{R}}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{M} u(\mathbf{x}, t_i) u^T(\mathbf{y}, t_i)}{M}$$

POD Steps

- Generate database using full-order model
- Assembly autocorrelation matrix and extract eigenmodes
- Substitute approximation in governing equations and perform Galerkin projection
- Solve ODE system to obtain time coefficients and reconstruct solution

OTHER POD-Like Reduced-Order Models

- Bi-orthogonal Decomposition (Audry, 1991)
- Balanced Proper Orthogonal Decomposition (Rowley, 2005)
- Dynamic Mode Decomposition (Schmid, 2010)
- Dynamic Proper Orthogonal Decomposition (Freno & Cizmas, 2015)
- Constrained Proper Orthogonal Decomposition (Reddy *et al.*, 2017)

Dynamic POD

Motivation

- POD worked fine on NASA Rotor 67 with variable back pressure
- POD failed on NASA Rotor 67 with deforming blades
- POD worked fine on Tenth Standard Configuration, (moving blades) subsonic flow
- POD failed on Tenth Standard Configuration, (moving blades) transonic flow

POD - New Approach Dynamic Average

Dynamic average

$$\bar{\mathbf{U}}(\mathbf{x};\mathbf{\Gamma}) \equiv \bar{\mathbf{U}}_0(\mathbf{x}) + \sum_{k=1}^{d} \gamma_k \bar{\mathbf{U}}_k(\mathbf{x}), \quad \mathbf{\Gamma} \equiv \{\gamma_1, \ldots, \gamma_d\}^T$$

Optimal dynamic average if

$$J\left[\bar{\mathbf{U}}_0, \ldots, \bar{\mathbf{U}}_d\right] \equiv \left\langle \left\| \mathbf{U} - \bar{\mathbf{U}}_0 - \sum_{k=1}^{d} \gamma_k \bar{\mathbf{U}}_k \right\|^2 \right\rangle \text{ is minimized}$$

$$\frac{\partial J}{\partial \delta}\left[\bar{\mathbf{U}}_0, \ldots, \bar{\mathbf{U}}_{k-1}, \bar{\mathbf{U}}_k + \delta\boldsymbol{\phi}, \bar{\mathbf{U}}_{k+1}, \ldots, \bar{\mathbf{U}}_d\right]\Bigg|_{\delta=0} = 0, \quad 0 \le k \le d$$

$$\Rightarrow \left\langle \begin{Bmatrix} \gamma_0 \\ \vdots \\ \gamma_d \end{Bmatrix} \begin{Bmatrix} \gamma_0 \\ \vdots \\ \gamma_d \end{Bmatrix}^T \right\rangle \begin{Bmatrix} \bar{\mathbf{U}}_0 \\ \vdots \\ \bar{\mathbf{U}}_d \end{Bmatrix} = \begin{Bmatrix} \langle \gamma_0 \mathbf{U} \rangle \\ \vdots \\ \langle \gamma_d \mathbf{U} \rangle \end{Bmatrix}, \quad \gamma_0 = 1$$

POD - New Approach Dynamic Basis Functions

$$\boldsymbol{\varphi}_j(\mathbf{x};\boldsymbol{\Gamma}) \equiv \tilde{\boldsymbol{\varphi}}_0^j(\mathbf{x}) + \sum_{k=1}^{d} \gamma_k \tilde{\boldsymbol{\varphi}}_k^j(\mathbf{x}) \qquad \boldsymbol{\Gamma} = \{\gamma_1, \ldots, \gamma_d\}^T$$

$$\tilde{\mathbf{U}}_j \equiv \mathbf{U} - \bar{\mathbf{U}} - \sum_{i=1}^{j-1} a_i \boldsymbol{\varphi}_i$$

$$\boldsymbol{\varphi}^T \hat{\mathbf{A}} \boldsymbol{\varphi} \equiv \tilde{\boldsymbol{\varphi}}^T \mathbf{A} \tilde{\boldsymbol{\varphi}} \qquad (\boldsymbol{\varphi}, \boldsymbol{\varphi}) \equiv \tilde{\boldsymbol{\varphi}}^T \mathbf{B} \tilde{\boldsymbol{\varphi}}$$

$$\tilde{\boldsymbol{\varphi}} \equiv \begin{Bmatrix} \tilde{\boldsymbol{\varphi}}_0 \\ \vdots \\ \tilde{\boldsymbol{\varphi}}_d \end{Bmatrix}, \; \mathbf{A}(t) \equiv \begin{bmatrix} \gamma_0\gamma_0\hat{\mathbf{A}} & \cdots & \gamma_0\gamma_d\hat{\mathbf{A}} \\ \vdots & \ddots & \vdots \\ \gamma_d\gamma_0\hat{\mathbf{A}} & \ldots & \gamma_d\gamma_d\hat{\mathbf{A}} \end{bmatrix}, \; \mathbf{B}(t) \equiv \begin{bmatrix} \gamma_0\gamma_0\mathbf{I} & \cdots & \gamma_0\gamma_d\mathbf{I} \\ \vdots & \ddots & \vdots \\ \gamma_d\gamma_0\mathbf{I} & \cdots & \gamma_d\gamma_d\mathbf{I} \end{bmatrix}$$

$$J[\boldsymbol{\varphi}] \equiv \left\langle \frac{\boldsymbol{\varphi}^T \hat{\mathbf{A}}(t) \boldsymbol{\varphi}}{(\boldsymbol{\varphi}, \boldsymbol{\varphi})} \right\rangle \quad \rightarrow \quad J[\tilde{\boldsymbol{\varphi}}] = \left\langle \frac{\tilde{\boldsymbol{\varphi}}^T \mathbf{A}(t) \tilde{\boldsymbol{\varphi}}}{\tilde{\boldsymbol{\varphi}}^T \mathbf{B}(t) \tilde{\boldsymbol{\varphi}}} \right\rangle$$

POD - New Approach
Dynamic Basis Functions

$$\left| \left\langle \frac{\mathbf{A}(t)}{\tilde{\boldsymbol{\varphi}}^T \mathbf{B}(t) \tilde{\boldsymbol{\varphi}}} - \frac{\left(\tilde{\boldsymbol{\varphi}}^T \mathbf{A}(t) \tilde{\boldsymbol{\varphi}}\right) \mathbf{B}(t)}{\left(\tilde{\boldsymbol{\varphi}}^T \mathbf{B}(t) \tilde{\boldsymbol{\varphi}}\right)^2} \right\rangle \right| = 0$$

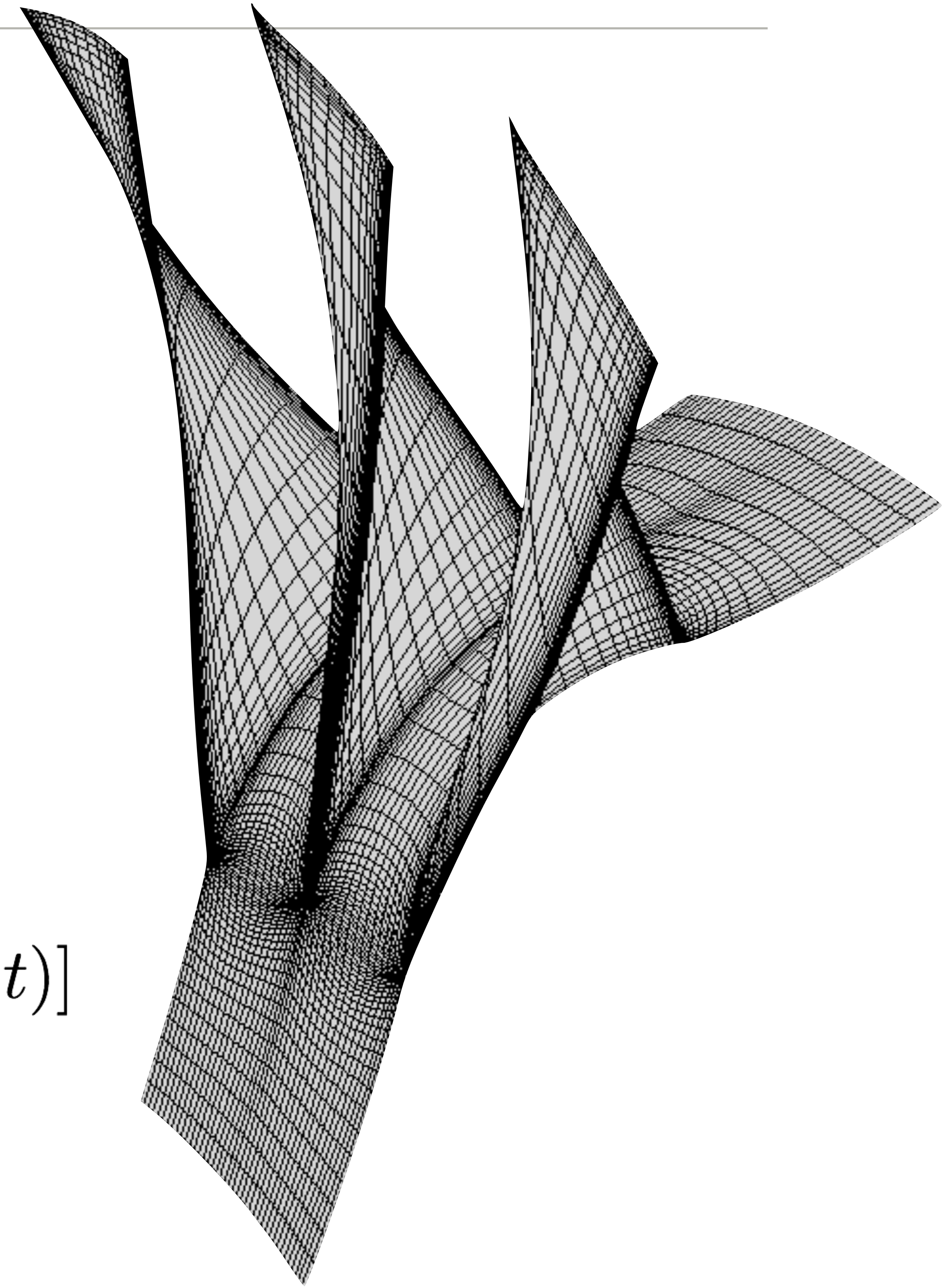
$$\tilde{\mathbf{A}} \tilde{\boldsymbol{\varphi}} = \lambda \tilde{\mathbf{B}} \tilde{\boldsymbol{\varphi}}$$

$$\tilde{\mathbf{A}} \equiv \left\langle \frac{\mathbf{A}(t)}{\tilde{\boldsymbol{\varphi}}^T \mathbf{B}(t) \tilde{\boldsymbol{\varphi}}} \right\rangle, \; \hat{\mathbf{B}} \equiv \left\langle \frac{\left(\tilde{\boldsymbol{\varphi}}^T \mathbf{A}(t) \tilde{\boldsymbol{\varphi}}\right) \mathbf{B}(t)}{\left(\tilde{\boldsymbol{\varphi}}^T \mathbf{B}(t) \tilde{\boldsymbol{\varphi}}\right)^2} \right\rangle,$$

$$\lambda \equiv \left\langle \frac{\tilde{\boldsymbol{\varphi}}^T \mathbf{A}(t) \tilde{\boldsymbol{\varphi}}}{\tilde{\boldsymbol{\varphi}}^T \mathbf{B}(t) \tilde{\boldsymbol{\varphi}}} \right\rangle, \tilde{\mathbf{B}} \equiv \frac{\hat{\mathbf{B}}}{\lambda}$$

Transonic Fan Flow Conditions

Parameter	Value	Unit
Stagnation Pressure	101,600	Pa
Stagnation Temperature	288	K
Nominal Back Pressure	102,300	Pa
Inlet Mach Number	0.6098	--
Wheel Speed	16,056	RPM
Back Pressure Amplitude, A_p	0.01 & 0.05	--
Tip Chord	92.18	mm
Tip Relative Velocity†	453.9	m/s

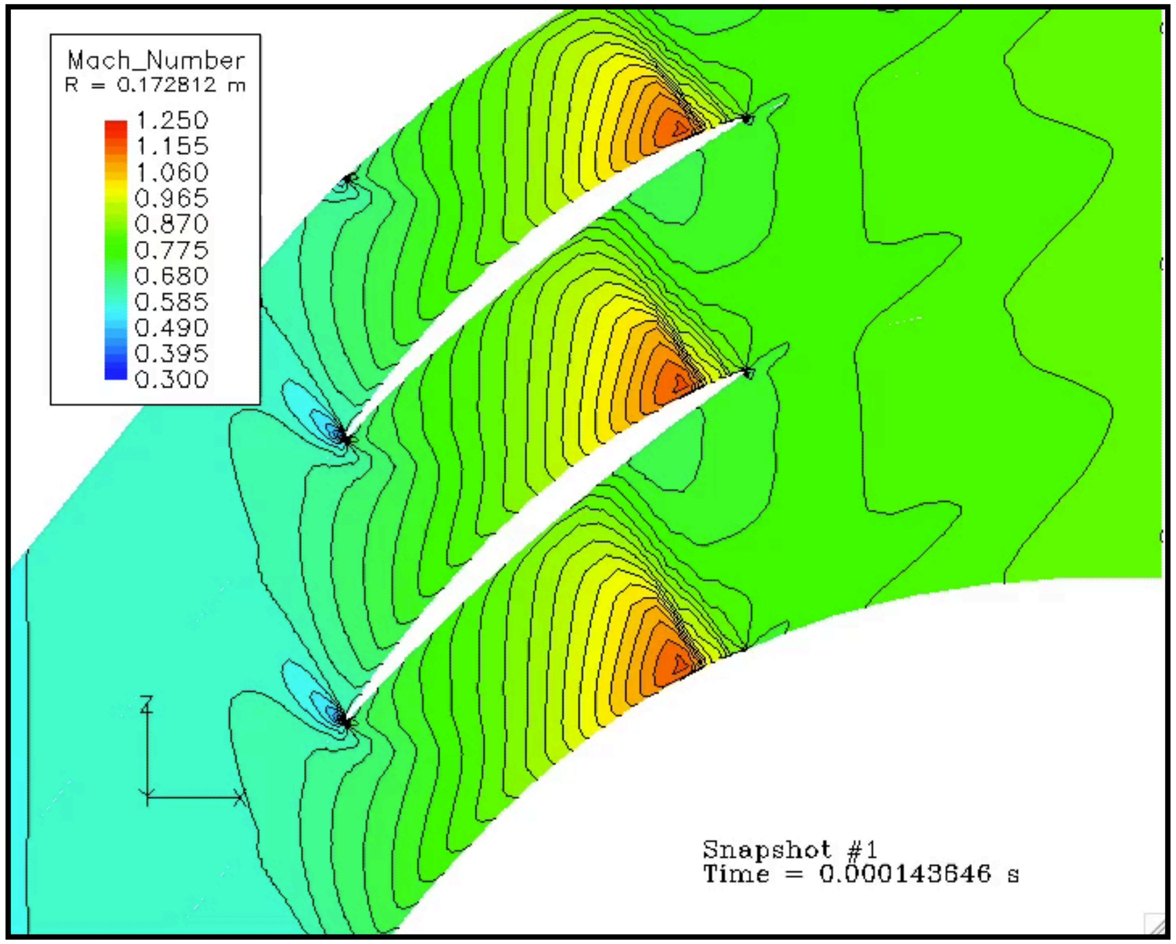


$$p_{back}(t) = p_{back}\big|_0 \left[1 - A_p \sin\left(984.81t\right)\right]$$

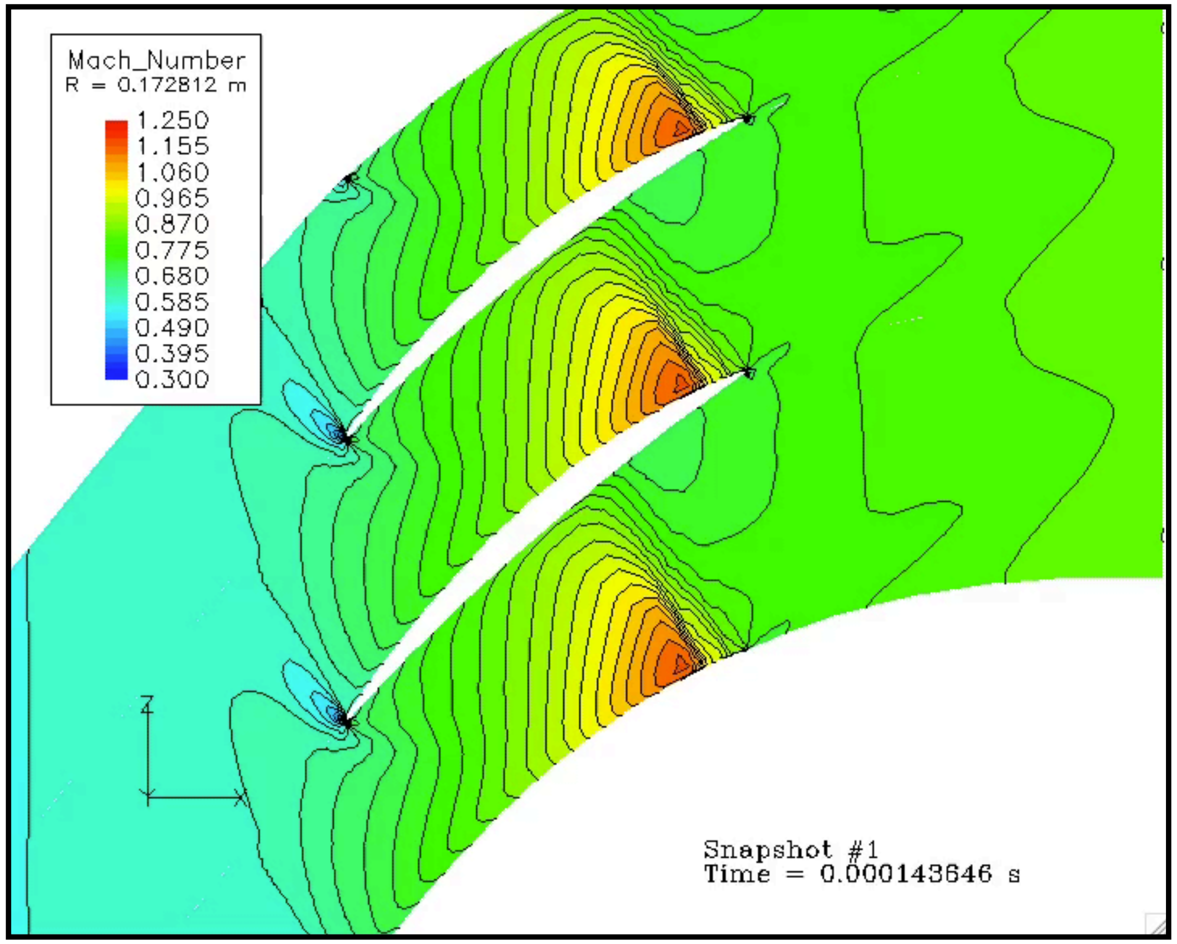
$$k := \frac{\omega}{2}\left(\frac{c}{U_{rel}}\right)\bigg|_{tip} = 0.1$$

†Taken from Table IX, reading 1393 in Urasek (1979)

Transonic Fan FOM/ROM Comparison: 10% Case



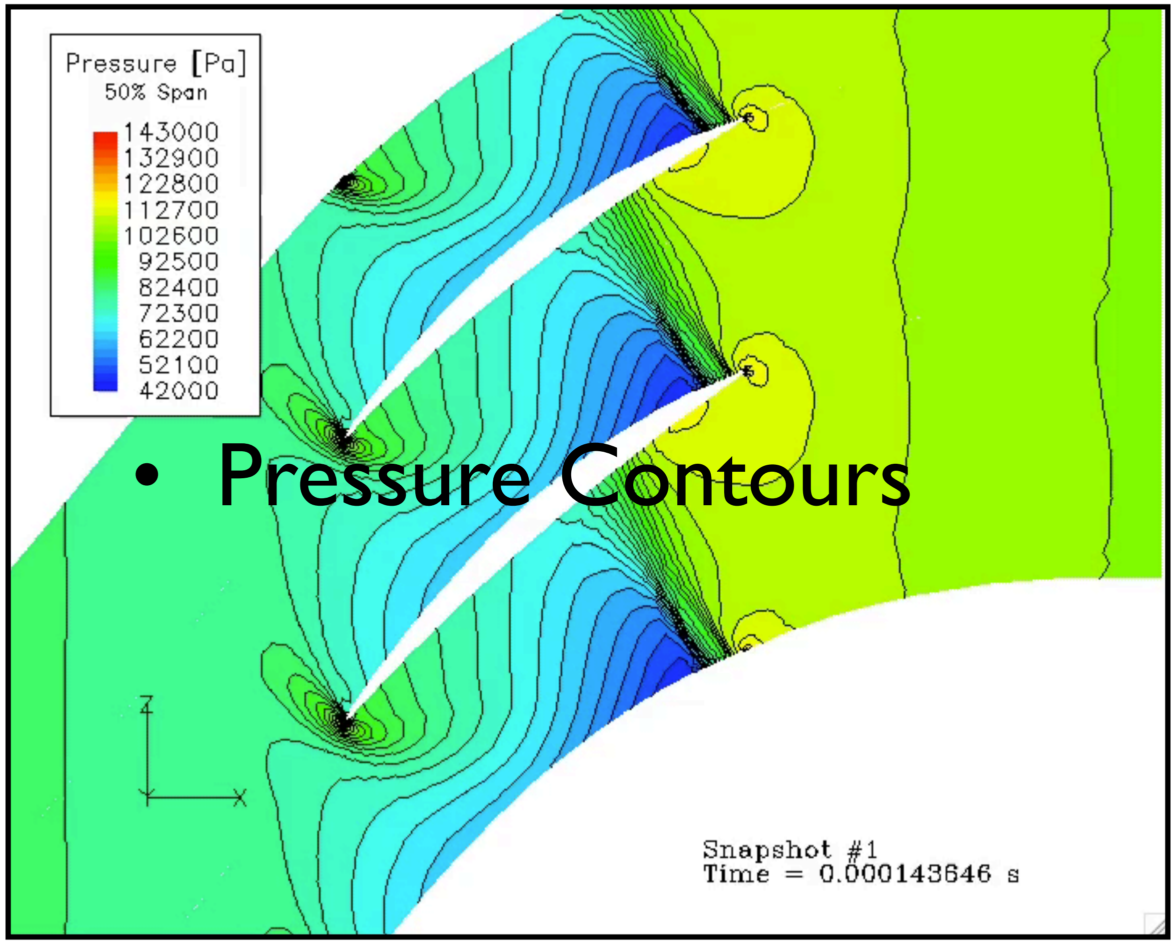
Full Order Model



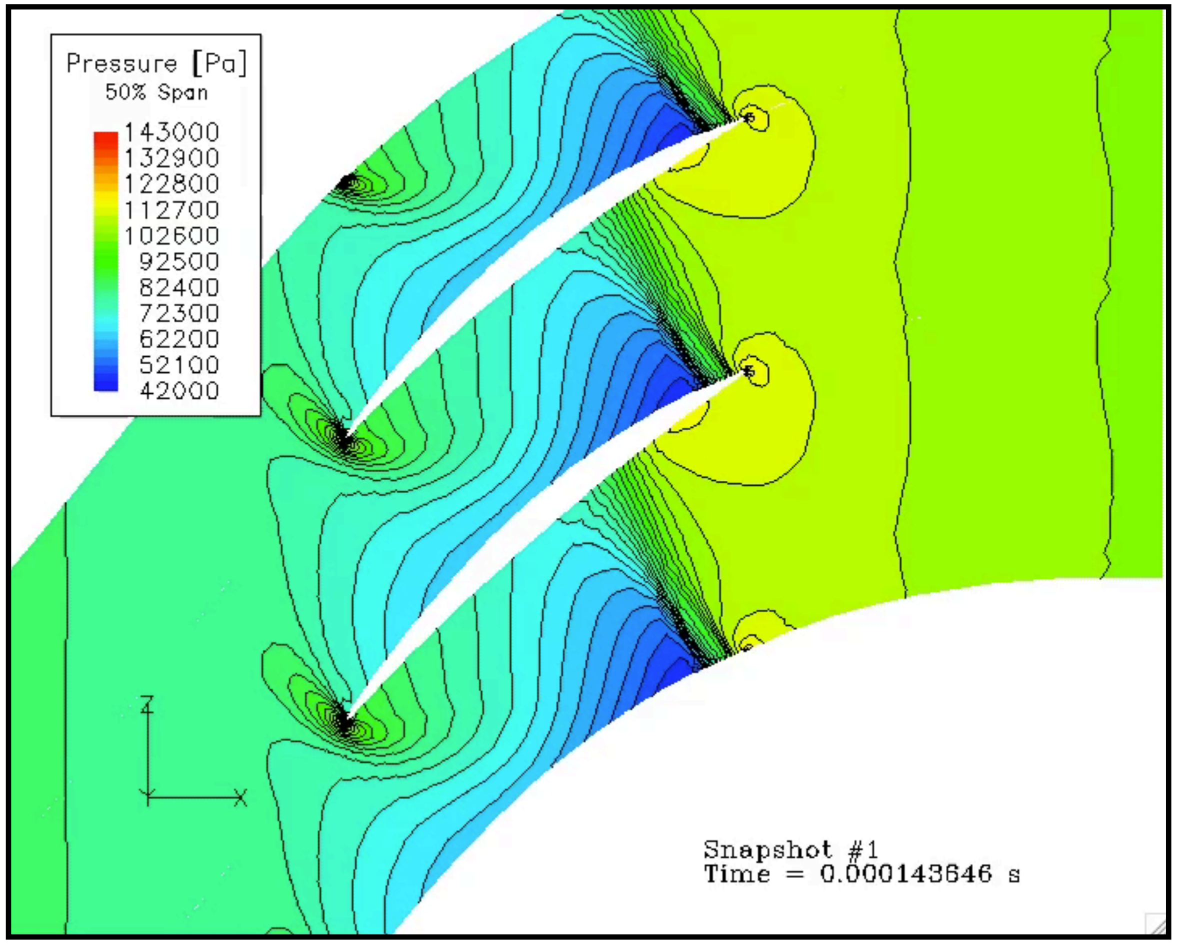
Reduced Order Model

Transonic Fan FOM/ROM Comparison: 10% Case

- Pressure Contours



Full Order Model

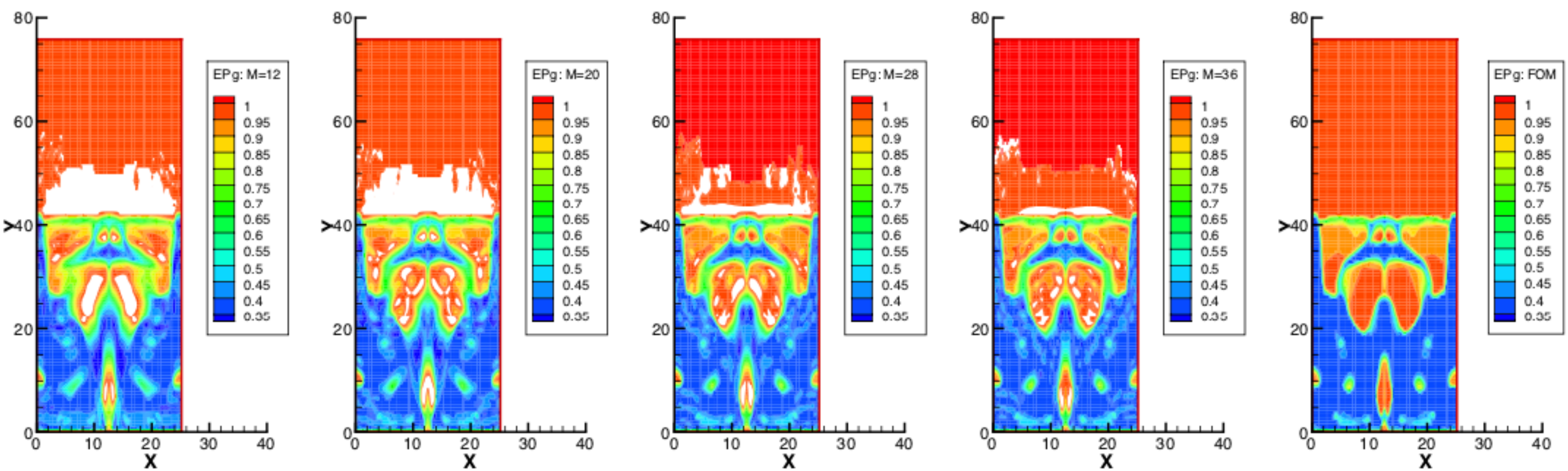


Reduced Order Model

Constrained POD

Gas Void Fraction
Bubbles with Jet

Gidaspow, Multiphase Flow and Fluidization, p.158



12 20 28 36 Mfix

Odex modes

Constrained POD with Karush-Kuhn-Tucker Conditions

Let us consider the first-order wave equation

$$u_t + cu_x = 0, \quad x \in [0,1], \quad c > 0, \tag{1}$$

with the initial condition

$$u(x,0) = f(x) \geq 0. \tag{2}$$

Approximate $u(x,t)$ using the POD method

$$u(x,t) \approx \sum_{i=1}^{m} a_i(t)\phi_i(x) \tag{3}$$

$$\dot{a}_i\phi_i + ca_i\phi_i' = 0 \tag{4}$$

Apply Galerkin projection

$$\int_0^1 \dot{a}_i\phi_i\phi_j dx + \int_0^1 ca_i\phi_i'\phi_j dx = 0 \tag{5}$$

which to due to orthonormality of POD basis functions reduces to

$$\dot{a}_j + c\left(\int_0^1 \phi_i'\phi_j dx\right) a_i = 0 \tag{6}$$

Constrained POD with Karush-Kuhn-Tucker Conditions

$$\dot{\underline{a}} + \boldsymbol{B}\underline{a} = \underline{0} \tag{1}$$

where $\underline{a} \in \mathbb{R}^m$, $\boldsymbol{B} \in \mathbb{R}^{m \times m}$, and the elements of the $\boldsymbol{B}$ matrix are

$$B_{ij} = c\left(\int_0^1 \phi_i' \phi_j dx\right) \tag{2}$$

Using an implicit time integration scheme

$$\left(\mathbf{I} + \Delta t\ \boldsymbol{B}\right)\underline{a}^{n+1} - \underline{a}^n = \underline{0} \tag{3}$$

where $\mathbf{I}$ is the identity matrix of rank m, Δt is the time step, and $\underline{a}^n := \underline{a}(t^n)$. With the notation $\boldsymbol{C} = \mathbf{I} + \Delta t\ \boldsymbol{B}$,

$$\boldsymbol{C}\underline{a}^{n+1} - \underline{a}^n = \underline{0} \tag{4}$$

Constrained POD with Karush-Kuhn-Tucker Conditions

Using Karush-Kuhn-Tucker condition, the non-negativity requirement is

$$\underline{\lambda}^T \mathbf{\Phi} \underline{a}^{n+1} \geq 0 \tag{1}$$

where $\mathbf{\Phi} = [\underline{\phi}_1 \ldots \underline{\phi}_m]$, $\mathbf{\Phi} \in \mathbb{R}^{N \times m}$ is the matrix of POD modes, N is the number of spatial points, and $\underline{\lambda}$ is the vector of KKT multipliers, $\underline{\lambda} \in \mathbb{R}^N$. Minimizing the functional

$$J = ||\boldsymbol{C}\underline{a}^{n+1} - \underline{a}^n||^2 + \underline{\lambda}^T \mathbf{\Phi} \underline{a}^{n+1} \tag{2}$$

which requires that

$$\begin{aligned} J_{\underline{a}^{n+1}} &= 2\boldsymbol{C}\underline{a}^{n+1} - 2\underline{a}^n + \mathbf{\Phi}^T \underline{\lambda} = 0 \\ J_{\underline{\lambda}} &= \mathbf{\Phi}\underline{a}^{n+1} = 0 \end{aligned} \tag{3}$$

yields the time coefficients and the KKT multipliers by solving

$$\begin{bmatrix} 2\boldsymbol{C} & \mathbf{\Phi}^T \\ \mathbf{\Phi} & \underline{0} \end{bmatrix} \begin{Bmatrix} \underline{a}^{n+1} \\ \underline{\lambda} \end{Bmatrix} = \begin{Bmatrix} 2\underline{a}^n \\ \underline{0} \end{Bmatrix} \tag{4}$$

Constrained POD with Karush-Kuhn-Tucker Conditions

Burgers equations

$$u_t + uu_x = 0, \quad x \in [0,1] \tag{1}$$

POD approximation and Galerkin projection yields

$$\dot{a}_k + a_i a_j G_{ijk} = 0, \quad i,j,k = 1,m \tag{2}$$

where $G_{ijk} = \int_0^1 \phi_j \phi_i' \phi_k dx$. For two POD modes, that is, m=2, the discretized form of (2) becomes:

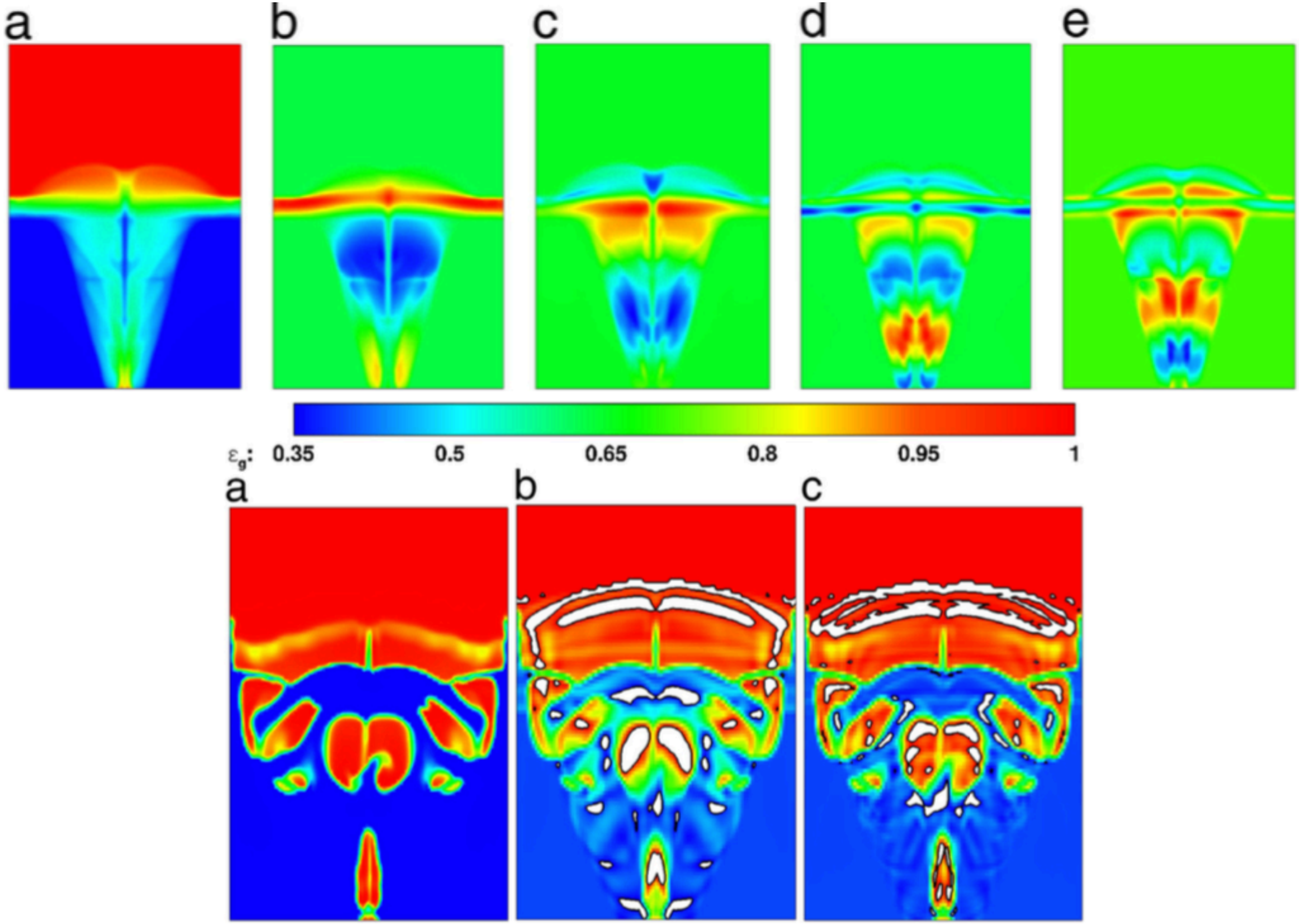
$$\begin{bmatrix} 1 + a_i^n G_{i11}\Delta t & a_i^n G_{i21}\Delta t \\ a_i^n G_{i12}\Delta t & 1 + a_i^n G_{i22}\Delta t \end{bmatrix} \begin{Bmatrix} a_1^{n+1} \\ a_2^{n+1} \end{Bmatrix} = \begin{Bmatrix} a_1^n \\ a_2^n \end{Bmatrix} \tag{3}$$

$$\boldsymbol{\mathcal{C}_{n\ell}}\, \underline{a}^{n+1} - \underline{a}^n = \underline{0} \tag{4}$$

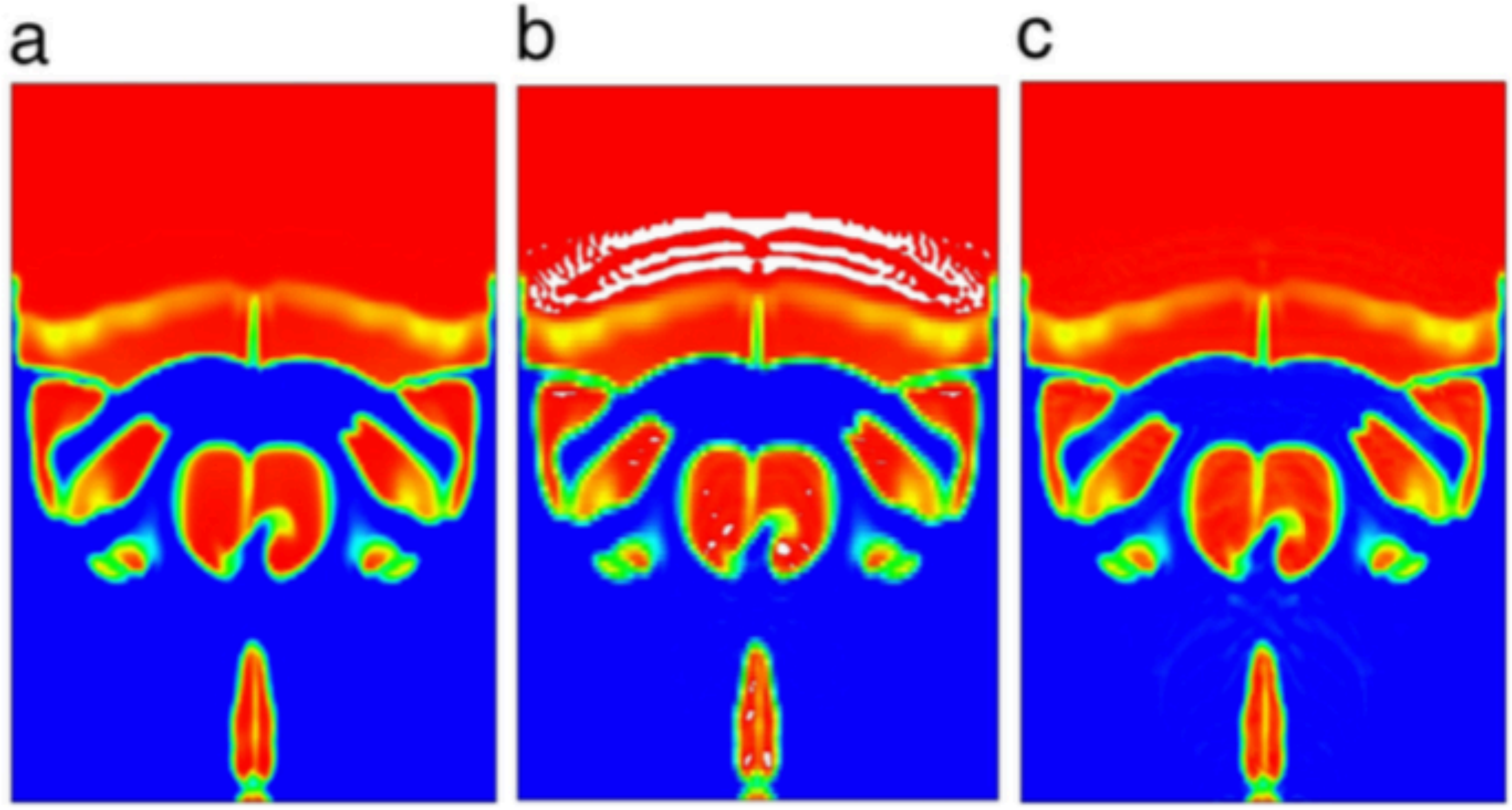
where

$$\boldsymbol{\mathcal{C}_{n\ell}} = \begin{bmatrix} 1 + a_i^n G_{i11}\Delta t & a_i^n G_{i21}\Delta t \\ a_i^n G_{i12}\Delta t & 1 + a_i^n G_{i22}\Delta t \end{bmatrix} \tag{5}$$

Constrained POD with Karush-Kuhn-Tucker Conditions



Constrained POD with Karush-Kuhn-Tucker Conditions



Mfix

Odex
86 modes

cOdex
86 modes

ζ-variable POD Method

Which State Vector to Use?

Options:

$$\text{conservative } q = \{\rho, \rho u, \rho v, \rho w, \rho E\}^T$$

$$\text{primitive } V = \{\rho, u, v, w, p\}^T$$

$$\text{zeta } Z = \{\zeta, u, v, w, p\}^T$$

Conservative Form Projection Difficulty

- Euler equations in conservative form

$$\rho_{,t} + (\rho u_i)_{,i} = 0, \quad i = 1,3$$
$$(\rho u_i)_{,t} + (\rho u_i u_j)_{,j} + p_{,i} = 0, \quad i,j = 1,3$$
$$(\rho E)_{,t} + (\rho u_i H)_{,i} = 0, \quad i = 1,3$$

- Consider, for example, the term

$$(\rho u_1^2)_{,1} \equiv \frac{\partial(\rho u^2)}{\partial x} = \frac{\partial}{\partial x}\left(\frac{(\rho u)^2}{\rho}\right) = \frac{\partial}{\partial x}\left(\frac{\alpha_i(t)\phi_i^{\rho u}(\boldsymbol{x})\alpha_j(t)\phi_j^{\rho u}(\boldsymbol{x})}{\alpha_k(t)\phi_k^{\rho}(\boldsymbol{x})}\right), \quad i,j = 0, M^{\rho u}, k = 0, M^{\rho}$$

- Galerkin projection

$$\left(\frac{\partial}{\partial x}\left(\frac{\alpha_i(t)\phi_i^{\rho u}(\boldsymbol{x})\alpha_j(t)\phi_j^{\rho u}(\boldsymbol{x})}{\alpha_k(t)\phi_k^{\rho}(\boldsymbol{x})}\right), \phi_\ell^{\rho u}(\boldsymbol{x})\right) \quad i,j,\ell = 0, M^{\rho u}, k = 0, M^{\rho}$$

- which becomes

$$\left(\frac{\alpha_i\alpha_j\alpha_k\left[\phi_{i_x}^{\rho u}\phi_j^{\rho u}\phi_k^{\rho} + \phi_i^{\rho u}\phi_{j_x}^{\rho u}\phi_k^{\rho}\right] + \alpha_i\alpha_j\alpha_k\phi_i^{\rho u}\phi_j^{\rho u}\phi_{k_x}^{\rho}}{(\alpha_k\phi_k^{\rho})^2}, \phi_\ell^{\rho u}\right) \quad i,j,\ell = 0, M^{\rho u}, k = 0, M^{\rho}$$

- Since the quotient contains time coefficients (*i.e.*, unknowns), cannot complete the summation step and data must be stored at all nodes

Navier-Stokes Equations using Primitive & ζ variables

Primitive variables

$$\frac{\partial \rho}{\partial t} + \rho \frac{\partial u_i}{\partial x_i} + u_i \frac{\partial \rho}{\partial x_i} = 0$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} + \frac{1}{\rho}\frac{\partial p}{\partial x_i} - \frac{1}{\rho \mathrm{Re}}\left(\frac{\partial^2 u_i}{\partial x_i^2} + \frac{1}{3}\frac{\partial^2 u_j}{\partial x_i \partial x_j}\right) = 0$$

$$\frac{\partial p}{\partial t} + u_i \frac{\partial p}{\partial x_i} + \gamma p \frac{\partial u_i}{\partial x_i} -$$

$$\frac{(\gamma - 1)\mu}{\mathrm{Re}} \frac{\partial u_i}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) = 0.$$

ζ variables

$$\frac{\partial \zeta}{\partial t} - \zeta \frac{\partial u_i}{\partial x_i} + u_i \frac{\partial \zeta}{\partial x_i} = 0$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} + \zeta \frac{\partial p}{\partial x_i} - \frac{\zeta}{\mathrm{Re}}\left(\frac{\partial^2 u_i}{\partial x_i^2} + \frac{1}{3}\frac{\partial^2 u_j}{\partial x_i \partial x_j}\right) = 0$$

$$\frac{\partial p}{\partial t} + u_i \frac{\partial p}{\partial x_i} + \gamma p \frac{\partial u_i}{\partial x_i} -$$

$$\frac{(\gamma - 1)\mu}{\mathrm{Re}} \frac{\partial u_i}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) = 0.$$

POD

Using reconstruction

$$\tilde{q}(x,t_i) = \sum_{k=1}^{M} a_k(t_i)\phi_k(x), \qquad i = 1,\ldots,M$$

into governing equations yields

$$\sum_{i=0}^{n^{Z_k}} \dot{a}_i^{Z_k}\phi_i^{Z_k}(\mathbf{x}) = \mathscr{I}_k(\mathbf{a},\boldsymbol{\phi}), \quad k \in [1, D+2]$$

Projecting along POD basis functions

$$\dot{a}_j^{Z_k} = \left(\mathscr{I}_k(\mathbf{a},\boldsymbol{\phi}), \phi_j^{Z_k}(\mathbf{x})\right), \quad j \in [1, n^{Z_k}],\ k \in [1, D+2]$$

System of ODEs

$$\dot{\mathbf{a}} = \mathscr{F}(\mathbf{a})$$

Stability of POD

- POD-based reduced-order models do not necessarily retain stability of full-order models
- Approaches used to obtain stable POD
  - Penalty method
  - Artificial dissipation
  - Modified number of modes

Penalty Method

- Definition
  - "on-reference conditions" = conditions identical to those for which FOM predicted snapshots
- For on-reference conditions, boundary conditions imposed on FOM constrain basis functions and ROM satisfies boundary conditions
- For off-reference conditions, ROM might not satisfy boundary conditions; boundary conditions become compatibility constraints on ROM

Penalty Method (cont'd)

- Penalty method used to alleviate compatibility constraints
- System of ODEs supplemented by the projection of difference ROM solution and prescribed boundary condition

$$\dot{\mathbf{a}} = \mathscr{F}(\mathbf{a}) - \boldsymbol{\tau}\mathbf{K}. \qquad \mathbf{K}_i = (\mathbf{Z}_{\mathrm{BC}} - \mathbf{F}, \phi_{\mathrm{BC}_i}),\ 1 \leq i \leq m$$

- Penalty parameter bounds

$$\tau \geq \frac{(\mathscr{F}(\mathbf{a}), \mathbf{a})}{(\mathbf{K}, \mathbf{a})}$$

Artificial Dissipation

$$\dot{\mathbf{a}} = \mathscr{F}(\mathbf{a}) + \tilde{\boldsymbol{\nu}}\mathbf{S}$$

- Diagonal matrix $\tilde{\boldsymbol{\nu}} \in \mathbb{R}^{m \times m}$
- Projection of basis function Laplacian $\mathbf{S} \in \mathbb{R}^{m}$
- Combined artificial dissipation and penalty method

$$\dot{\mathbf{a}} = \mathscr{F}(\mathbf{a}) + \tilde{\boldsymbol{\nu}}\mathbf{S} - \boldsymbol{\tau}\mathbf{K}.$$

- Dissipation range

$$0 \leq \tilde{\nu} \leq \frac{(\boldsymbol{\tau}\mathbf{K} - \mathscr{F}(\mathbf{a}), \mathbf{a})}{(\mathbf{S}, \mathbf{a})}.$$

Modified Number of Modes

- Size of system of ODEs is $m = \sum_{k=1}^{D+2} n^{Z_k}$
- Jacobian of reduced-order model

$$J_{ij}^{Z_k} = \frac{\partial \mathscr{F}_i^{Z_k}(\mathbf{a})}{\partial a_j}, \quad i, j \in [1, n^{Z_k}],\ k \in [1, D+2].$$

- Part of Jacobian that accounts for effect of specific volume on energy balance equation

$$\frac{\partial \boldsymbol{\mathscr{F}}^p}{\partial \mathbf{a}^\zeta} = \begin{bmatrix} \frac{\partial \mathscr{F}_1^p}{\partial a_1^\zeta} & \frac{\partial \mathscr{F}_1^p}{\partial a_2^\zeta} & \cdots & \frac{\partial \mathscr{F}_1^p}{\partial a_{n^\zeta}^\zeta} \\ \frac{\partial \mathscr{F}_2^p}{\partial a_1^\zeta} & \frac{\partial \mathscr{F}_2^p}{\partial a_2^\zeta} & \cdots & \frac{\partial \mathscr{F}_2^p}{\partial a_{n^\zeta}^\zeta} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathscr{F}_{n^p}^p}{\partial a_1^\zeta} & \frac{\partial \mathscr{F}_{n^p}^p}{\partial a_2^\zeta} & \cdots & \frac{\partial \mathscr{F}_{n^p}^p}{\partial a_{n^\zeta}^\zeta} \end{bmatrix}$$

Results

ζ-variable POD Method

Results

- Quasi-One-Dimensional Nozzle Flow
- NASA Rotor 67
- 10th Standard Configuration
- 11th Standard Configuration

Computational Time

- Comparison among three models
  - Two ROMs (zeta-POD, rho-POD)
  - FOM
- Comparison among three flow cases
  - Rotor 67
  - 10th Standard Configuration
  - 11th Standard Configuration

Computational Time (cont'd)

Breakdown of CPU runtime for FOM and ζ-POD

Case	Rotor 67	10 SC	11 SC
Grid nodes, N	299,844	33,068	78,260
Snapshots per Period	50	100	100
Total Snapshots	250	500	500
FOM Snapshots [s]	388,891	473,059	2,260,688
POD Basis Functions [s]	479	173	397
POD Basis/FOM Snapshots	0.12%	0.04%	0.02%

CPU runtime: FOM vs ROM

Case	N	M	FOM [s]	ROM [s]	Ratio
Rotor 67	299,844	250	4,860,247	51.30	94,749
10 SC	33,068	500	530,558	12.46	42,580
11 SC	78,260	500	2,260,688	24.35	92,841

Computational Time (cont'd)

Breakdown of CPU runtime of ζ-POD ROM and theoretical upper limit of speedup

Case	No. Periods	ROM Setup [s]	ROM Solve ODE [s]	Upper Limit	% Upper Limit
Rotor 67	5	19.45	31.84	152,623	62%
10 SC	5	5.97	6.49	81,773	52%
11 SC	5	11.93	12.42	182,020	51%

Computational Time (cont'd)

- CPU runtime with and without additional post processing for 11th Standard Configuration
- Without post-processing: ζ-POD outputs only time coefficients at each time step
- With post-processing: ζ-POD outputs also relative errors which require reconstruction at each time step

	Post-process	
	No	Yes
zeta-POD [s]	24.35	189.33
FOM [s]	2.26×10^6	2.26×10^6
FOM/zeta-POD	92,841	11,940
rho-POD [s]	3.77×10^5	3.77×10^5
rho-POD/zeta-POD	15,469	1989

Conclusions

- ζ-POD without post-processing is five orders of magnitude faster than FOM and four orders of magnitude faster than traditional POD
- ζ-POD with post-processing is four orders of magnitude faster than FOM and three orders of magnitude faster than traditional POD
- ζ-POD was not applied yet to two-phase flows but it should

Thank you!