REDUCED-ORDER MODELING FOR LAMINAR FLOW IN A PROTOTYPE AIRLINE TERMINAL CONFIGURATION

Contract number 233519
Contractor: Max Gunzburger
Additional participants: Janet Peterson and John Burkardt

November 2, 2004

1 Introduction

We consider the flow in the H-shaped region that is a prototype for an airline terminal. The inflow and outflow orifices are as described in the figures given to us. A sketch of the region and the outflow and inflow orifices are given in Figure 1 (which is not drawn to scale.) Note that there are four set of inflow orifices with each set containing four orifices; these are labeled 1 through 4 in the sketch. There are two sets of outflow orifices with each set containing three orifices; these are labeled 5 and 6 in the sketch. In addition, there is another orifice through which the flow is allowed to move in or out as needed in order to conserve mass.

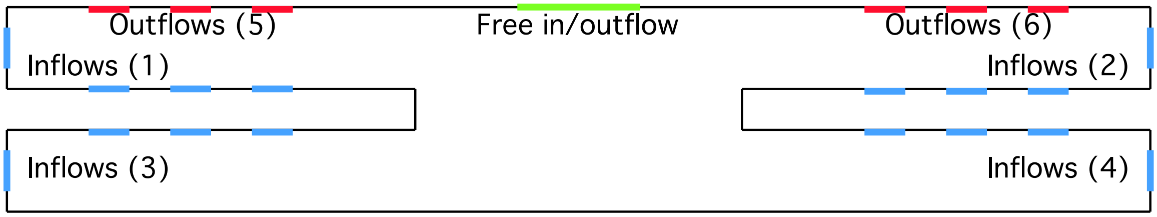


Figure 1: Prototype airline terminal with the configuration of inflow and outflow orifices.

We are allowed to specify the mass flow at each of the orifices (except, of course, the free orifice.) We do so as follows. At each relevant orifice, we assume the incoming or outgoing velocity is normal to the orifice and has a parabolic profile; thus, for example, at the orifices at the lower-left of the region, we specify

$$\begin{cases} u = \beta_3 V_{max,3} \phi_3(y) \quad \text{and} \quad v = 0 & \text{along the vertical orifice} \\ u = 0 \quad \text{and} \quad v = -\beta_3 V_{max,3} \psi_{3,k}(x) & \text{along the three horizontal orifices } (k = 1, 2, 3)\,, \end{cases} \tag{1}$$

where $\phi_3(y)$ and $\psi_{3,k}(x)$, $k = 1, 2, 3$, are specified parabolic functions with maximum values unity at the center of the corresponding orifices, $V_{max,3}$ is the maximum allowable speed, and β_3 is a parameter that sets the actual mass flow being specified. We have that $0 \leq \beta_3 \leq 1$. The other inflow and outflow orifices are treated in the same manner, except for the free orifice along which we give a free flow condition for which the horizontal velocity and the normal stress vanish. For the maximum speeds, we use the values given to us:

$$V_{max,1} = 0.12, \quad V_{max,2} = 0.125, \quad V_{max,3} = 0.11,$$
$$V_{max,4} = 0.088, \quad V_{max,5} = 0.128, \quad V_{max,6} = 0.224\,.$$

As a result, we have six parameters β_j, $j = 1, \ldots, 6$, at our disposal that specify the boundary data of the problem; allowable parameter values lie in or on the unit hypercube in six dimensions.[1] In addition, we have several other fixed parameters that help determine the problem specification. These include the time interval $[0, T]$ over which we wish to know the flow field and the Reynolds number. Once values for all these parameters have been chosen and an initial condition[2] has been specified, the flow in the region of interest is determined by solving a finite difference in time/finite element in space discretization of the Navier-Stokes equations for laminar, viscous, incompressible flow; the boundary conditions are chosen as described above. In our computations, we use the backward-Euler scheme and the Taylor-Hood element pair to effect the time and space discretizations, respectively.

1.1 Reduced-order modeling for flow problems

The reduced-order models we consider require that the following steps be performed. Step 1 represents the snapshot generation facet of reduced-order modeling; step 2 is the reduced-basis construction step; and step 3 is the use of the reduced-basis to determine a low-dimensional approximation of the flow.

1a. A set $\{\vec{\beta}_k\}_{k=1}^{K}$ of parameter values is determined.
   - In our context, each $\vec{\beta}_k$ is a six-dimensional vector and the set of parameters are determined by sampling K points in the six-dimensional unit hypercube.

1b. A set of time values $\{t_n\}_{n=1}^{N}$ in the interval $[0, T]$ are chosen; these values are usually chosen to coincide with some, but not all, of the time values used in the discretization of the flow equations.
   - In our context, we usually sample uniformly in time using a time step that is a multiple of the time step used for the backward-Euler discretization of the flow equations; sometimes, we use different multiples in different subintervals of $[0, T]$.

1c. The discretized flow equations are solved over the time interval $[0, T]$ for each of the parameter values determined in step 1a. Each discrete solution is evaluated at the time instants $\{t_n\}_{n=1}^{N}$. In this manner, one assembles the set $\{\vec{U}_\ell\}_{\ell=1}^{L}$ of snapshots, where, for $\ell = n + N(k-1)$, $\vec{U}_i$ is the vector of coefficients in the finite element expansion of the approximate solution corresponding to the parameter value β_k, $k = 1, \ldots, K$, evaluated at the time instant t_n, $n = 1, \ldots, N$. The cardinality of the snapshot set is $L = NK$.

1d. Depending on how reduced bases are used to solve the flow equations, the snapshot set determined in step 1c may be modified to satisfy homogeneous boundary conditions. This is accomplished by subtracting from each snapshot an appropriate particular solution.
   - In our context, we would have up to six boundary conditions to zero out. So, we would effect the replacement

$$\vec{U}_\ell \leftarrow \vec{U}_\ell - \sum_{i=1}^{6} \overline{\beta}_{i,\ell} \overline{U}_i,$$

[1] We will use the abbreviated vector notation $\vec{\beta}$ for a point in parameter space, i.e., $(\vec{\beta})_j = \beta_j$ for $j = 1, \ldots, 6$.

[2] We defer further discussion of initial conditions until specific simulations are discussed.

where $\overline{U}_i$, $i = 1, \dots, 6$, are six vectors of finite element coefficients corresponding to solutions of the Navier-Stokes equations and, for each ℓ, $\beta_{i,\ell}$ for $i = 1, \dots, 6$ are chosen so that the new $\vec{U}_\ell$ satisfies homogeneous boundary conditions. For example, the $\overline{U}_i$'s could be steady state solutions of the Navier-Stokes system corresponding to chosen values for the parameters or it could be a solution of that system at some instant of time, e.g., one of the snapshots.

2. Using the set of L (possibly modified) snapshots determined in step 1, we determine a reduced basis.
   - In our studies, we construct two types of reduced bases: POD (proper orthogonal decomposition) and CVT (centroidal Voronoi tessellation). In both cases, we end up with a set $\{\vec{V}_d\}_{d=1}^{D}$ of vectors, each of which is a linear combination of the (possibly modified) snapshot vectors.
3. The reduced-basis determined in step 2 is used to find a low-dimensional approximation of the flow field for specified initial and boundary data. If $u_d(\mathbf{x})$ denotes the finite element function corresponding to the basis vector $\vec{V}_d$, $d = 1, \dots, D$, then we seek an approximation $u_{ro}(t, \mathbf{x})$ of the flow equations of the form

$$u_{ro}(t, \mathbf{x}) = u_p(t, \mathbf{x}) + \sum_{d=1}^{D} c_d(t) u_d(\mathbf{x}).$$

At $t = 0$, $u_{ro}(0, \mathbf{x})$ is chosen to satisfy the specified initial data in the same manner as is done in standard finite element discretizations.

If the snapshot set has been modified so that the snapshot vectors satisfy zero boundary conditions, then $u_p(t, \mathbf{x})$ must be chosen so that $u_{ro}(t, \mathbf{x})$ satisfies the specified boundary conditions. For example, suppose that the boundary conditions are determined by the given functions $\beta_j(t)$, $j = 1, \dots, 6$. We then can define

$$u_p(t, \mathbf{x}) = \sum_{j=1}^{6} \alpha_j(t) \widehat{u}_j(\mathbf{x}),$$

where, for example, $\widehat{u}_j(\mathbf{x})$, $j = 1, \dots, 6$, could be six steady state solutions of the discretized Navier-Stokes system (corresponding to six different boundary conditions) or could be snapshots. For each t, the coefficients $\alpha_j(t)$ are solutions of the linear system

$$\beta_j = \sum_{i=1}^{6} \alpha_i \widehat{\beta}_{j,i} \qquad \text{for } j = 1, \dots, 6,$$

where $\widehat{\beta}_{j,i}$ denotes the i-th component of the boundary data vector satisfied by the j-th particular solution $\widehat{u}_j$. In the case of the snapshot set having been modified so that the snapshot vectors satisfy zero boundary conditions, the test functions in the Galerkin method are chosen to be the basis functions $u_d(\mathbf{x})$, $d = 1, \dots, D$.

A simpler means of determining a reduced-basis approximation to the flow is to use unmodified snapshots. In this case, both the snapshots and the reduced basis vectors will satisfy inhomogeneous boundary conditions. We also now simply choose $u_p(t, \mathbf{x}) = 0$ so that the reduced basis approximation is simply given by

$$u_{ro}(t, \mathbf{x}) = \sum_{d=1}^{D} c_d(t) u_d(\mathbf{x})$$

and we use the basis functions $u_d(\mathbf{x})$, $d = 1, \ldots, D$ as test functions. Note that neither the test or trial functions satisfy the correct boundary conditions so that these must be imposed separately, i.e., in addition to the equations resulting from the Galerkin discretization of the flow equations in the reduced basis space, we have the six equations

$$\sum_{d=1}^{D} c_d(t)\beta_{d,j} = \beta_j(t),$$

where $\beta_{d,j}$ denotes the j-th component of the boundary data vector satisfied by the basis function $u_d(\mathbf{x})$ and $\beta_j(t)$ denotes the j-th component of the given boundary data vector. We now have an overdetermined system, i.e., we have $D+6$ equations and D unknowns, so that we have to delete six of the equations coming from the Galerkin discretization of the flow equations. One possible choice is to delete the six equations corresponding to the test functions $u_d(\mathbf{x})$ for $d = D-5, \ldots, D$.

– The computational results presented here were determined by the first approach to satisfying boundary conditions, i.e., by using snapshots and reduced-basis vectors that satisfy homogeneous boundary conditions, and then using particular solutions to satisfy given boundary conditions.

2 Effects of hypercube sampling strategies on the potential accuracy of reduced-order models

We want to compare the effects of the use of different sampling algorithms on the accuracy of a type of reduced-order approximation. What is learned from this study will affect how we sample parameter space during the snapshot generation process. We performed the following study.

1. We sample seven points within the six-dimensional unit hypercube. We use several different sampling methods so that we produce several sets of points, with each set containing seven points in the unit hypercube. The sampling methods we use are:

   (a) Corner - the six vertices of the hypercube along the coordinate axes and the vertex opposite the origin

   (b) MC - uniform random or Monte Carlo sampling in the hypercube

   (c) Ham - Hammersley sampling in the hypercube

   (d) LHS - Latin hypercube sampling in the hypercube

   (e) CVT - centroidal Voronoi tessellation sampling in the hypercube

   (f) LCVT - Latinized centroidal Voronoi tessellation sampling in the hypercube.

2. We then solve the Navier-Stokes equations 42 times using each of the total of 42 points in the six sets determined in Step 1. Each point sampled has six coordinates which provide the values of $\{\beta_j\}_{j=1}^{6}$ that are used in boundary conditions such as (1). As a result, we obtained six reduced basis sets, one each corresponding to the six sets of sample points. Each reduced basis set contains seven vectors that are solutions of the Navier-Sokes equations for each of the seven points in parameter space which make up the corresponding sample set.

3. We select 10 more random points in the hypercube and solve the Navier-Stokes equations for the values of $\{\beta_j\}_{j=1}^6$ specified by each of the 10 points. We use these 10 solutions as representatives to see how well the reduced basis sets do in trying to approximate solutions of the Navier-Stokes equations for general data points in the hypercube.
4. We determine the best approximation (in a least-squares sense) out of the span of each of the six basis sets to each of the 10 random solutions determined in Step 3.

In Tables 1 and 2, we give for each of u and v and for each of the 10 random solutions found in Step 3, the relative L^2 errors in the approximate least-squares solutions determined from each of the six reduced bases. In the Table 3, we give the total (with respect to all 10 random solutions) relative L^2 errors in the approximate least-squares solutions determined from each of the six reduced bases.

From Tables 1–3, we see that the results for several of the point sampling methods are comparable, but that, in general, CVT point sampling seems to do a little better.

	Corner	MC	Ham	LHS	CVT	LCVT
Solution 1	0.212	0.121	0.289	0.079	0.077	0.113
Solution 2	0.115	0.278	0.025	0.093	0.019	0.035
Solution 3	0.181	0.051	0.092	0.053	0.109	0.092
Solution 4	0.153	0.094	0.112	0.089	0.052	0.043
Solution 5	0.107	0.073	0.095	0.100	0.044	0.069
Solution 6	0.148	0.103	0.054	0.082	0.040	0.054
Solution 7	0.286	0.287	0.319	0.310	0.280	0.284
Solution 8	0.236	0.166	0.295	0.205	0.077	0.055
Solution 9	0.143	0.072	0.071	0.087	0.024	0.044
Solution 10	0.122	0.162	0.228	0.099	0.077	0.070

Table 1: Relative L^2 errors in u of the approximate least-squares reduced basis approximations.

	Corner	MC	Ham	LHS	CVT	LCVT
Solution 1	0.304	0.169	0.460	0.092	0.098	0.165
Solution 2	0.149	0.194	0.035	0.134	0.024	0.050
Solution 3	0.278	0.083	0.160	0.073	0.175	0.140
Solution 4	0.291	0.156	0.272	0.209	0.105	0.069
Solution 5	0.174	0.083	0.167	0.110	0.061	0.113
Solution 6	0.258	0.158	0.137	0.144	0.067	0.090
Solution 7	0.395	0.305	0.406	0.266	0.279	0.310
Solution 8	0.352	0.193	0.524	0.215	0.104	0.092
Solution 9	0.192	0.059	0.112	0.123	0.035	0.055
Solution 10	0.263	0.238	0.527	0.224	0.231	0.242

Table 2: Relative L^2 errors in v of the approximate least-squares reduced basis approximations.

	Corner	MC	Ham	LHS	CVT	LCVT
u error	0.1618	0.1496	0.1591	0.1169	0.0841	0.0852
v error	0.2578	0.1642	0.2929	0.1599	0.1224	0.1321

Table 3: Total (with respect to all 10 solutions) relative L^2 errors in u and v of the approximate least-squares reduced basis approximations.

3 Reduced-order modeling of the flow in the airline terminal

We consider two preliminary test problems for the reduced order modeling (ROM) of the flow in the airline terminal in order to demonstrate the feasibility of the ROM approach.

For the first problem, snapshots are generated from simulations for which all the boundary condition parameters, β_i, $i = 1, \ldots, 6$ are identical; i.e., we only have one parameter determining the flow field. Reduced-order models determined from these snapshots are also used to approximate flows for which all the boundary condition parameters are the equal to each other.

For the second problem, snapshots are generated from simulations for which there are at most two parameters determining the flow field. Specifically, we have that $\beta_3 = \beta_4$ and $\beta_1 = \beta_2 = \beta_5 = \beta_6$. This choice of separation of the β_i's corresponds to the different regions of the terminal.

In what follows, note that two codes were used during the snapshot generation process: a steady-state code and a time-dependent code. In all calculations the viscosity was 0.01 and the number of finite element velocity degrees of freedom was 35,730.

In the tables and figures we provide, as functions of time, the relative L^2 or root-mean-square errors in the reduced-order solutions. More precisely, if (u^h, v^h) and (u_{rom}, v_{rom}) denotes the velocity components determined from the full-scale finite element code and a reduced-order model, respectively, we compute the relative errors

$$\frac{\|u^h - u_{rom}\|}{\|u^h\|} \qquad \text{and} \qquad \frac{\|v^h - v_{rom}\|}{\|v^h\|},$$

where

$$\|f\| = \left(\int_{flow\ region} f^2 \, dxdy \right)^{1/2}.$$

In the tables, we mostly report these relative errors as decimals, while in the figures we us percentages.

4 Problem 1. The one-parameter case

4.1 Snapshot generation

We generated a total of 439 snapshots from two runs as follows:

- Set $\beta_i = 1$, $i = 1, \ldots, 6$. Set initial condition to zero. The timestep for the backward-Euler method is $\Delta t = 0.1$ and the time-dependent code was run for 300 time steps. Snapshots were written off every two timesteps for a total of 150 snapshots. Normally for this run steady state is reached in approximately 283 timesteps with the usual steady state criteria. We then use the steady state (generated by the steady-state code) for all $\beta_i = 1$ for the particular solution used to modify the snapshots so that they satisfy homogeneous boundary conditions.

- Set $\beta_i = 0$, $i = 1, \ldots, 6$. Set initial conditions to the steady state for all $\beta_i = 1$. Set $\Delta t = 0.1$. We ran the code until it reached steady state and wrote off a total of 289 snapshots with a snapshot written off every timestep for the first 100 timesteps and then every other timestep for the next 200 timesteps and then every 5 timesteps after that. Of course, the particular solution in this case is zero.

4.2 POD reduced bases

Using POD analyses, we generated 4 and 8-dimensional POD bases from the snapshots determined as described in §4.1. The singular values corresponding the first 4 basis vectors were

$$\begin{pmatrix} 27.8963 \\ 6.4502 \\ 4.5800 \\ 2.2931 \end{pmatrix}$$

and those for the next four basis vectors were

$$\begin{pmatrix} 1.2649 \\ 0.7992 \\ 0.4416 \\ 0.2914 \end{pmatrix}$$

In each case the $m \times n$ snapshot matrix we used is of size $35,730 \times 439$. We note the (not too rapid) decay of the singular values of the snapshot matrix.

4.3 CVT reduced bases

CVT is an clustering method that is optimal in the sense that it reduces the total variance, or "energy," of the clustering. The means of the clusters are the CVT-reduced-basis vectors. In Tables 4 and 5, we provide some information about the 4 and 8-dimensional CVT reduced bases, respectively. In those tables, "Cluster" refers to the clusters of snapshots determined by the CVT clustering algorithms, "Population" refers to the cardinality of the corresponding cluster, the first "Percentage" refers to the percentage of snapshots found in the corresponding cluster, "Energy" refers to to the contribution made by the corresponding cluster to the CVT energy or variance, the second "Percentage" refers to the percentage contibution of the corresponding cluster to the total energy, and "Min" and "Max" respectively refer to the smallest and largest indices of the snapshots found in the corresponding cluster. We note that the contributions the total CVT energy is much better equilibrated for the case of the 8-dimensional basis.

Cluster	Population	Percentage	Energy	Percentage	Min	Max
1	48	10	3.80335	11	1	48
2	128	29	12.8132	37	151	278
3	161	36	13.6287	40	279	439
4	102	23	3.59558	10	49	150

Table 4: Information about the 4-dimensional CVT reduced basis.

4.4 Reduced-order simulations

We ran the reduced-order simulation code using both POD and CVT bases for two model problems.

Cluster	Population	Percentage	Energy	Percentage	Min	Max
1	38	8	1.08167	13	151	188
2	64	14	1.05191	13	189	252
3	71	16	0.980012	12	369	439
4	48	10	1.12282	13	27	74
5	52	11	0.658033	8	253	304
6	26	5	1.18009	14	1	26
7	76	17	1.15761	14	75	150
8	64	14	0.769459	9	305	368

Table 5: Information about the 8-dimensional CVT reduced basis.

4.4.1 Problem 1a

For this problem, we start with an initial condition equal to the steady state solution corresponding to all $\beta_i = 1$. The boundary condition for the problem we are solving is determined from

$$\beta_i = 0 \quad i = 1, \ldots, 6,$$

so that the particular solution is zero. The full Navier-Stokes code is used to generate the same solution with the same initial condition and the results are printed off for the first 20 timesteps ($\Delta t = 0.1$) and every 10 timesteps after that until steady state is reached.

Tables 6–9 provide a time history of the differences between the ROM solutions and the full finite element solutions for several POD and CVT bases. In those tables, (u^h, v^h) denotes the solution to the full Navier-Stokes equations at the indicated time. Also (u^h_{rb}, v^h_{rb}) denotes the solution to the ROM problem converted to a full vector and, if appropriate, adjusted to satisfy the boundary data. The norm used is the $L^2(\Omega)$-norm. From Tables 6–9, we see that both low-dimensional POD and CVT ROMs do quite well in approximating the flow field, with percentage errors for the 8-dimensional cases well under 5%.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.31377E-04	0.28909E-04	0.074964	0.11065
0.2	0.29125E-04	0.26972E-04	0.070042	0.10370
0.3	0.27043E-04	0.25257E-04	0.065456	0.097523
0.4	0.25121E-04	0.23731E-04	0.061188	0.092002
0.5	0.23349E-04	0.22367E-04	0.057224	0.087053
1.0	0.16490E-04	0.17401E-04	0.041608	0.068919
2.0	0.10626E-04	0.13024E-04	0.028224	0.053082
3.0	0.10287E-04	0.11807E-04	0.028575	0.049261
4.0	0.10677E-04	0.11409E-04	0.030870	0.048562
6.0	0.98962E-05	0.10763E-04	0.030698	0.047354
8.0	0.80137E-05	0.10004E-04	0.026436	0.045221
10.0	0.65741E-05	0.96023E-05	0.022929	0.044435
20.0	0.11436E-04	0.14489E-04	0.050229	0.073712
30.0	0.12380E-04	0.16567E-04	0.065610	0.091804
40.0	0.98224E-05	0.15136E-04	0.061240	0.091274
43.0	0.92208E-05	0.14730E-04	0.060133	0.091098

Table 6: Errors for Problem 1a using 4 POD basis functions.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.45346E-05	0.58908E-05	0.010834	0.022548
0.2	0.37104E-05	0.46534E-05	0.0089230	0.017891
0.3	0.30261E-05	0.36831E-05	0.0073243	0.014221
0.4	0.24753E-05	0.29373E-05	0.0060290	0.011388
0.5	0.20553E-05	0.23857E-05	0.0050371	0.0092852
1.0	0.15772E-05	0.16440E-05	0.0039797	0.0065113
2.0	0.18086E-05	0.17706E-05	0.0048040	0.0072162
3.0	0.11859E-05	0.12020E-05	0.0032943	0.0050149
4.0	0.71656E-06	0.74994E-06	0.0020719	0.0031920
6.0	0.12089E-05	0.11548E-05	0.0037501	0.0050808
8.0	0.11989E-05	0.11889E-05	0.0039551	0.0053743
10.0	0.78022E-06	0.80685E-06	0.0027212	0.0037338
20.0	0.13314E-05	0.13487E-05	0.0058475	0.0068613
30.0	0.61413E-06	0.56743E-06	0.0032546	0.0031444
33.0	0.99106E-06	0.92473E-06	0.0055271	0.0052562

Table 7: Errors for Problem 1a using 8 POD basis functions.

t	$\Vert u_{rb}^h - u^h\Vert_0$	$\Vert v_{rb}^h - v^h\Vert_0$	$\frac{\Vert u_{rb}^h-u^h\Vert_0}{\Vert u^h\Vert_0}$	$\frac{\Vert v_{rb}^h-v^h\Vert_0}{\Vert v^h\Vert_0}$
0.1	0.68600E-04	0.48035E-04	0.16389	0.18386
0.2	0.65436E-04	0.45724E-04	0.15737	0.17580
0.3	0.62420E-04	0.43591E-04	0.15108	0.16831
0.4	0.59542E-04	0.41609E-04	0.14503	0.16131
0.5	0.56793E-04	0.39758E-04	0.13919	0.15474
1.0	0.44742E-04	0.31996E-04	0.11289	0.12672
2.0	0.27102E-04	0.21170E-04	0.071991	0.086281
3.0	0.15625E-04	0.14072E-04	0.043403	0.058709
4.0	0.93748E-05	0.97441E-05	0.027106	0.041475
6.0	0.10964E-04	0.87039E-05	0.034010	0.038293
8.0	0.15643E-04	0.11691E-04	0.051604	0.052847
10.0	0.18431E-04	0.14189E-04	0.064281	0.065661
20.0	0.17479E-04	0.16302E-04	0.076770	0.082932
30.0	0.94155E-05	0.10624E-04	0.049898	0.058872
40.0	0.53630E-05	0.48304E-05	0.033437	0.029129
47.0	0.97141E-05	0.83381E-05	0.067107	0.053327

Table 8: Errors for Problem 1a using 4 CVT basis functions.

t	$\Vert u_{rb}^h - u^h\Vert_0$	$\Vert v_{rb}^h - v^h\Vert_0$	$\frac{\Vert u_{rb}^h-u^h\Vert_0}{\Vert u^h\Vert_0}$	$\frac{\Vert v_{rb}^h-v^h\Vert_0}{\Vert v^h\Vert_0}$
0.1	0.10130E-04	0.10712E-04	0.024202	0.041002
0.2	0.87229E-05	0.89942E-05	0.020978	0.034581
0.3	0.74572E-05	0.75256E-05	0.018050	0.029058
0.4	0.63191E-05	0.62616E-05	0.015392	0.024276
0.5	0.52968E-05	0.51683E-05	0.012981	0.020115
1.0	0.16444E-05	0.15973E-05	0.0041491	0.0063266
2.0	0.20001E-05	0.20371E-05	0.0053128	0.0083025
3.0	0.24795E-05	0.24007E-05	0.0068877	0.010016
4.0	0.19528E-05	0.18629E-05	0.0056463	0.0079292
6.0	0.46337E-06	0.41839E-06	0.0014374	0.0018407
8.0	0.65784E-06	0.62524E-06	0.0021702	0.0028264
10.0	0.86745E-06	0.81458E-06	0.0030254	0.0037695
20.0	0.55139E-06	0.53078E-06	0.0024218	0.0027002
30.0	0.22321E-06	0.23343E-06	0.0011829	0.0012935
40.0	0.11792E-05	0.11790E-05	0.0073523	0.0071097

Table 9: Errors for Problem 1a using 8 CVT basis functions.

4.4.2 Problem 1b

For this problem, we again start with an initial condition equal to the steady state solution corresponding to all $\beta_i = 1$. However, now the boundary condition for the problem we are solving is determined from

$$\beta_i = 0.5 \quad i = 1, \ldots, 6.$$

The particular solution used to force the ROM solutions to satisfy this boundary data is one-half the steady state solution corresponding to all $\beta_i = 1.0$. The full Navier-Stokes code is used to generate the same solution with the same initial condition and the results are printed off for the first 20 timesteps ($\Delta t = 0.1$) and every 10 timesteps after that until steady state is reached.

Tables 10–13 provide a time history of the differences between the ROM solutions and the full finite element solutions for several POD and CVT bases. We note that unlike Problem 1a, for this problem the target boundary condition is not one that was used for the generation of the snapshot set. Despite this fact, we see from Tables 10–13 that both low-dimensional POD and CVT ROMs still do quite well in approximating the flow field, with percentage errors for the 8-dimensional cases generally still under 5%.

Figures 2 and 3 provide the time history of the relative errors for the x and y-components of the velocity, respectively, for 4 and 8 dimensional POD and CVT reduced-order models. We see improvement in the accuracy as one increases the number or reduced-basis vectors. We also see that seemingly, POD-reduced-order modeling gives slightly better results than does CVT-reduced-order modeling. It should be noted that for both POD and CVT, very basic algorithms are being used and that improvements in both approaches may be possible by using more sophisticated methodologies.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.15492E-04	0.14116E-04	0.026283	0.036480
0.2	0.14183E-04	0.12852E-04	0.024106	0.033273
0.3	0.12975E-04	0.11736E-04	0.022091	0.030435
0.4	0.11862E-04	0.10745E-04	0.020230	0.027909
0.5	0.10840E-04	0.98610E-05	0.018518	0.025654
1.0	0.70951E-05	0.67396E-05	0.012216	0.017663
2.0	0.63136E-05	0.55998E-05	0.011018	0.014863
3.0	0.89299E-05	0.75923E-05	0.015763	0.020371
4.0	0.11112E-04	0.96503E-05	0.019811	0.026140
6.0	0.13745E-04	0.12441E-04	0.024919	0.034248
8.0	0.15178E-04	0.13857E-04	0.027897	0.038664
10.0	0.16166E-04	0.14530E-04	0.030060	0.041015
20.0	0.30051E-04	0.18295E-04	0.0568861	0.052581
22.0	0.32346E-04	0.19373E-04	0.061230	0.055679

Table 10: Error for Problem 1b using 4 POD basis functions

t	$\|u_{rb}^h - u^h\|_0$	$\|v_{rb}^h - v^h\|_0$	$\frac{\|u_{rb}^h - u^h\|_0}{\|u^h\|_0}$	$\frac{\|v_{rb}^h - v^h\|_0}{\|v^h\|_0}$
0.1	0.21864E-05	0.27427E-05	0.0037092	0.0070882
0.2	0.17988E-05	0.20222E-05	0.0030573	0.0052352
0.3	0.16266E-05	0.15819E-05	0.0027694	0.0041023
0.4	0.16672E-05	0.14410E-05	0.0028433	0.0037428
0.5	0.18613E-05	0.15424E-05	0.0031798	0.0040128
1.0	0.34009E-05	0.28585E-05	0.0058553	0.0074915
2.0	0.59180E-05	0.48912E-05	0.010328	0.012983
3.0	0.77137E-05	0.64404E-05	0.013617	0.017281
4.0	0.91450E-05	0.77603E-05	0.016304	0.021020
6.0	0.11401E-04	0.99160E-05	0.020669	0.027297
8.0	0.13155E-04	0.11573E-04	0.024177	0.032291
10.0	0.14602E-04	0.12874E-04	0.027152	0.036340
15.0	0.18156E-04	0.14448E-04	0.034370	0.041526

Table 11: Errors for Problem 1b using 8 POD basis functions

t	$\|u_{rb}^h - u^h\|_0$	$\|v_{rb}^h - v^h\|_0$	$\frac{\|u_{rb}^h - u^h\|_0}{\|u^h\|_0}$	$\frac{\|v_{rb}^h - v^h\|_0}{\|v^h\|_0}$
0.1	0.34128E-04	0.23717E-04	0.057900	0.061293
0.2	0.32387E-04	0.22306E-04	0.055045	0.057747
0.3	0.30733E-04	0.21022E-04	0.052325	0.054515
0.4	0.29160E-04	0.19846E-04	0.049731	0.051548
0.5	0.27663E-04	0.18761E-04	0.047257	0.048809
1.0	0.21200E-04	0.14393E-04	0.036500	0.037721
2.0	0.12591E-04	0.92841E-05	0.021974	0.024642
3.0	0.91903E-05	0.77553E-05	0.016223	0.020808
4.0	0.97655E-05	0.83805E-05	0.017411	0.022700
6.0	0.13524E-04	0.10882E-04	0.024518	0.029957
8.0	0.16280E-04	0.12754E-04	0.029922	0.035586
10.0	0.17962E-04	0.13930E-04	0.033401	0.039320
20.0	0.32598E-04	0.19235E-04	0.061706	0.055283

Table 12: Errors for Problem 1b using 4 CVT basis functions

t	$\Vert u_{rb}^h - u^h \Vert_0$	$\Vert v_{rb}^h - v^h \Vert_0$	$\frac{\Vert u_{rb}^h - u^h \Vert_0}{\Vert u^h \Vert_0}$	$\frac{\Vert v_{rb}^h - v^h \Vert_0}{\Vert v^h \Vert_0}$
0.1	0.50440E-05	0.52912E-05	0.8005574	0.013674
0.2	0.43640E-05	0.44385E-05	0.0074170	0.011491
0.3	0.38102E-05	0.37810E-05	0.0064871	0.0098051
0.4	0.33892E-05	0.32976E-05	0.0057802	0.0085655
0.5	0.31075E-05	0.29713E-05	0.0053086	0.0077302
1.0	0.34255E-05	0.29599E-05	0.0058978	0.0077573
2.0	0.60063E-05	0.48394E-05	0.010482	0.012845
3.0	0.79358E-05	0.65063E-05	0.014009	0.017457
4.0	0.93606E-05	0.78836E-05	0.016689	0.021354
6.0	0.11502E-04	0.10036E-04	0.020852	0.027627
8.0	0.13190E-04	0.11650E-04	0.024243	0.032505
10.0	0.14619E-04	0.12920E-04	0.027184	0.036469
20.0	0.29791E-04	0.18126E-04	0.056394	0.052094
30.0	0.52624E-04	0.30320E-04	0.099616	0.087143

Table 13: Errors for Problem 1b using 8 CVT basis functions

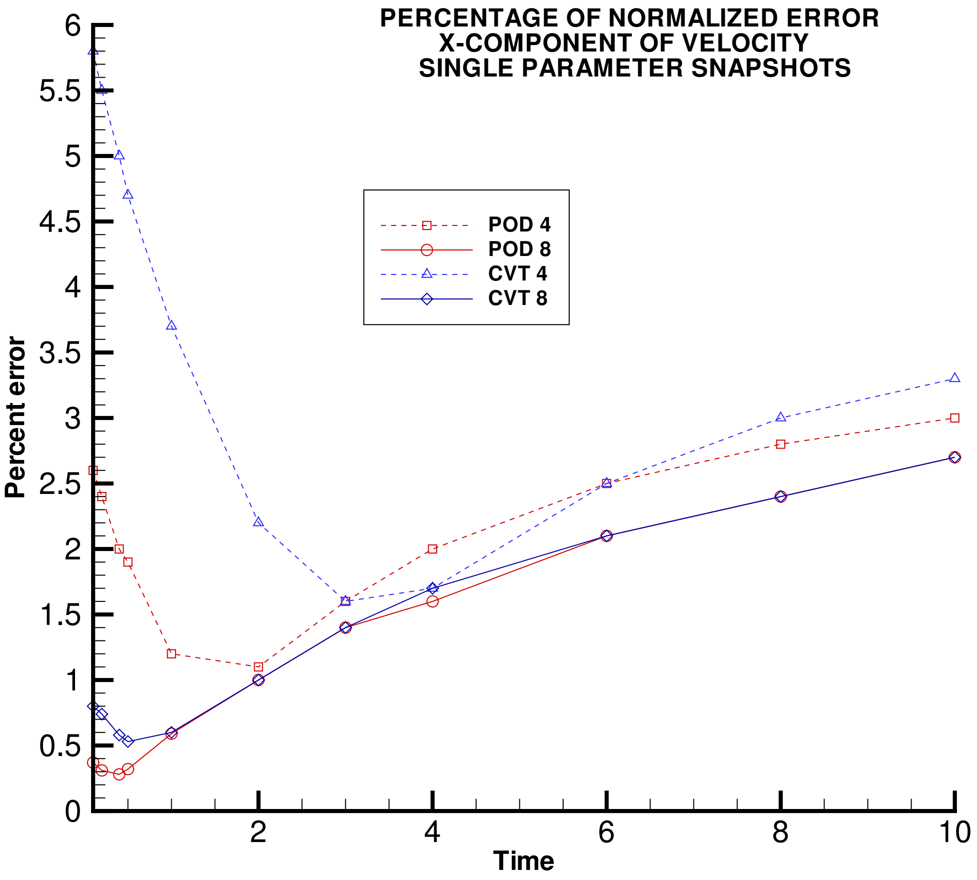


Figure 2: Percentage relative error in the x-component of the velocity for Problem 1b.

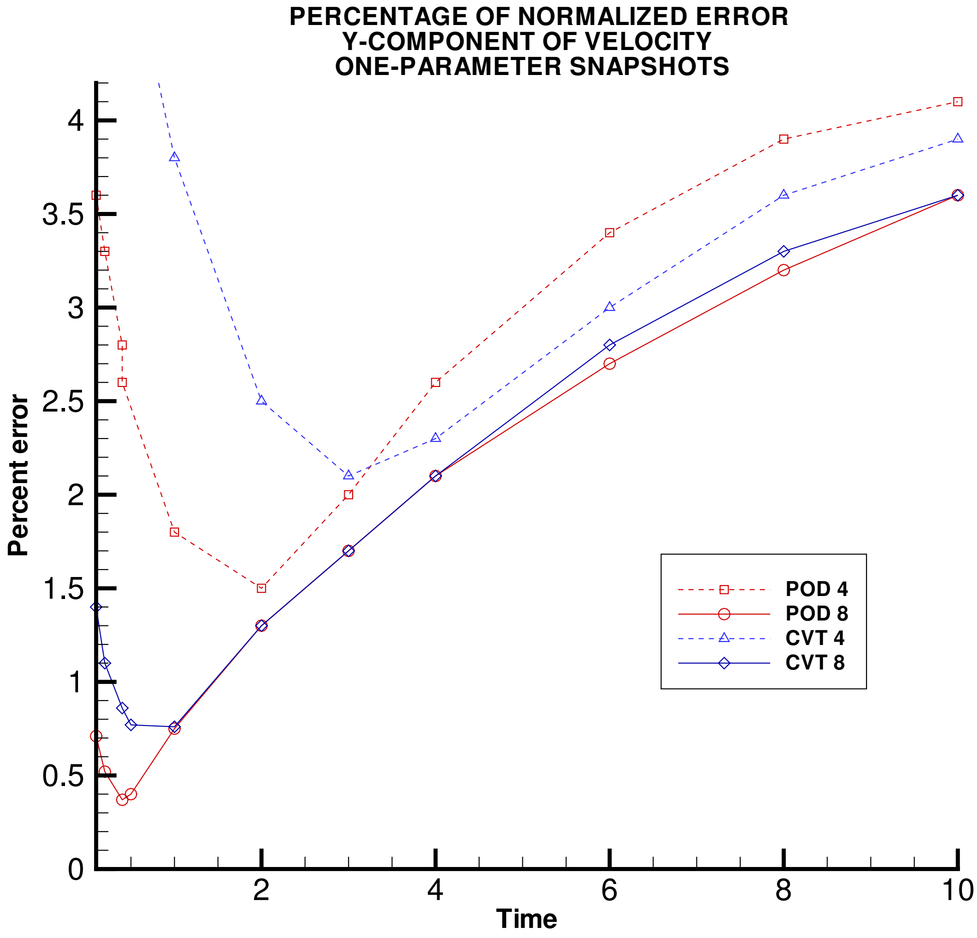


Figure 3: Percentage relative error in the y-component of the velocity for Problem 1b.

5 Problem 2. A two-parameter case

5.1 Snapshot generation

For this two-parameter problem, we generated 8 sets of snapshots. In each case, the time step used for the backward-Euler discretization was $\Delta t = 0.1$ and we ran 100 timesteps. The snapshots were generated as follows:

- Snapshots 1 - 100
  $$\text{boundary condition:} \quad \beta_i = 1 \quad i = 1\ldots,6$$
  $$\text{initial condition:} \quad 0$$
- Snapshots 101 - 200
  $$\text{boundary condition:} \quad \beta_i = 0 \quad i = 1\ldots,6$$
  $$\text{initial condition:} \quad \text{steady state solution corresponding to } \beta_i = 1,\ i = 1,\ldots,6$$
- Snapshots 201 - 300
  $$\text{boundary condition:} \quad \beta_1 = \beta_2 = \beta_5 = \beta_6 = 0 \qquad \beta_3 = \beta_4 = 1$$
  $$\text{initial condition:} \quad \text{steady state solution corresponding to } \beta_i = 1,\ i = 1,\ldots,6$$
- Snapshots 301 - 400
  $$\text{boundary condition:} \quad \beta_1 = \beta_2 = \beta_5 = \beta_6 = 1 \qquad \beta_3 = \beta_4 = 0$$
  $$\text{initial condition:} \quad \text{steady state solution corresponding to } \beta_i = 1,\ i = 1,\ldots,6$$
- Snapshots 401 - 500
  $$\text{boundary condition:} \quad \beta_1 = \beta_2 = \beta_5 = \beta_6 = 0.75 \qquad \beta_3 = \beta_4 = 0.25$$
  $$\text{initial condition:} \quad \text{steady state solution corresponding to } \beta_i = 1,\ i = 1,\ldots,6$$
- Snapshots 501 - 600
  $$\text{boundary condition:} \quad \beta_1 = \beta_2 = \beta_5 = \beta_6 = 0.25 \qquad \beta_3 = \beta_4 = 0.75$$
  $$\text{initial condition:} \quad \text{steady state solution corresponding to } \beta_i = 1,\ i = 1,\ldots,6$$
- Snapshots 601 - 700
  $$\text{boundary condition:} \quad \beta_1 = \beta_2 = \beta_5 = \beta_6 = 0.2 \qquad \beta_3 = \beta_4 = 0.4$$
  $$\text{initial condition:} \quad \text{steady state solution corresponding to } \beta_i = 1,\ i = 1,\ldots,6$$
- Snapshots 701 - 800
  $$\text{boundary condition:} \quad \beta_1 = \beta_2 = \beta_5 = \beta_6 = 0.8 \qquad \beta_3 = \beta_4 = 0.6$$
  $$\text{initial condition:} \quad \text{steady state solution corresponding to } \beta_i = 1,\ i = 1,\ldots,6$$

Each snapshot was modified to satisfy homogeneous boundary conditions by subtracting the steady state solution (generated by the steady-state code) for the same choice of β_i as was used to generate the snapshot.

5.2 POD reduced bases

Basis vectors were generated using 3 different sets of snapshots, each consisting of some or all the snapshots generated as described in §5.1.

- Case 1: snapshots 1-400 (excludes 4 sets generated by "random" β_i)
- Case 2: snapshots 1-200 and 401-800 (excludes snapshots generated at corners)
- Case 3: snapshots 1-800 (all snapshots)

The first 16 singular values of the snapshot matrix for each of the three cases are given in Table 14. For each case, sets of 8 and 16 POD basis vectors were computed.

Case 1	*Case 2*	*Case 3*
28.4317	31.2352	34.6394
10.8004	6.0001	22.8174
3.4575	3.7814	10.7466
3.0259	2.6014	4.2382
2.1111	1.7520	3.2045
1.4128	0.8029	2.2339
0.6675	0.7198	1.5430
0.5618	0.5974	1.4711
0.3007	0.5177	0.8029
0.1979	0.2925	0.5788
0.1697	0.2470	0.5105
0.1190	0.1988	0.3024
0.0629	0.1347	0.2966
0.0576	0.1140	0.2576
0.0393	0.0814	0.1955
0.0327	0.0544	0.1762

Table 14: Singular values for POD basis vectors generated using three sets of snapshots

5.3 Reduced order model

Reduced-order simulations were run for POD bases for five model problems. For all problems, the ROM code was run for 100 timesteps with $\Delta t = 0.1$. The ROM solution at specific timesteps was compared to the corresponding solution determined from the full finite element code. In each case, the $L^2(\Omega)$ was used for measuring the difference between ROM and full finite element results.

5.4 Problem 2a

For this problem, we set the initial condition to the steady state solution for all $\beta_i = 1$. We choose the boundary conditions to correspond to the choices

$$\beta_3 = \beta_4 = 1 \qquad \beta_1 = \beta_2 = \beta_5 = \beta_6 = 0.$$

We use this example to test three different ways of selecting a particular solution to use to have POD-based solutions satisfy the boundary conditions.

For Tables 15 and 16, the particular solution is chosen to be the steady state solution for the same values of β_i used to define the boundary condition; this is possible because that solution is available (it was already used in the generation of the snapshots.) From those tables, we see that we obtain excellent accuracy from the POD ROMs.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.95721E-05	0.78586E-05	0.010744	0.015942
0.2	0.84530E-05	0.68505E-05	0.0094917	0.013902
0.3	0.74459E-05	0.60291E-05	0.0083637	0.012239
0.4	0.65428E-05	0.53659E-05	0.0073518	0.010896
0.5	0.57384E-05	0.48385E-05	0.0064500	0.0098285
1.0	0.31471E-05	0.36356E-05	0.0035425	0.0073957
2.0	0.34254E-05	0.36916E-05	0.0038646	0.0075273
3.0	0.41501E-05	0.35292E-05	0.0046904	0.0072095
4.0	0.38752E-05	0.28887E-05	0.0043859	0.0059102
6.0	0.19502E-05	0.16363E-05	0.0022121	0.0033564
8.0	0.29726E-05	0.32075E-05	0.0033774	0.0065935
10.0	0.60240E-05	0.53568E-05	0.0068502	0.011024

Table 15: Errors for Problem 2a using 8 POD basis functions generated from the 400 snapshots of Case 1. The particular solution is chosen to be the steady state for the boundary condition.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.59461E-06	0.91350E-06	0.66743E-03	0.18531E-02
0.2	0.38775E-06	0.55505E-06	0.43540E-03	0.11264E-02
0.3	0.23528E-06	0.30204E-06	0.26428E-03	0.61315E-03
0.4	0.13933E-06	0.15577E-06	0.15656E-03	0.31631E-03
0.5	0.11572E-06	0.15108E-06	0.13006E-03	0.30689E-03
1.0	0.23055E-06	0.31548E-06	0.25952E-03	0.64177E-03
2.0	0.14408E-06	0.15662E-06	0.16255E-03	0.31935E-03
3.0	0.14408E-06	0.13883E-06	0.16283E-03	0.28360E-03
4.0	0.13434E-06	0.13766E-06	0.15204E-03	0.28165E-03
6.0	0.13609E-06	0.13269E-06	0.15436E-03	0.27218E-03
8.0	0.10491E-06	0.87615E-07	0.11920E-03	0.18010E-03
10.0	0.86787E-07	0.10267E-06	0.98677E-04	0.21125E-03

Table 16: Errors for Problem 2a using 16 POD basis functions generated from the 400 snapshots of Case 1. The particular solution is chosen to be the steady state for the boundary condition.

For Table 17, the particular solution is chosen to be the sum of the steady state solution for $\beta_3 = 1$ and all other $\beta_i = 0$ and the steady state solution for $\beta_4 = 1$ and all other $\beta_i = 0$. The results were substantially worse than those of Tables 15 and 16, with some percentage errors as high as 22%.

Next, we tried setting the particular solution used to satisfy the boundary conditions to be all zero except at the appropriate locations on the boundary. This resulted in very large error, e.g., percentage errors were around 80%.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.73449E-04	0.10538E-03	0.82444E-01	0.21377E+00
10.0	0.72971E-04	0.10567E-03	0.82980E-01	0.21747E+00

Table 17: Errors for Problem 2a using 8 POD basis functions generated from the 400 snapshots of Case 1. The solution satisfying the boundary data is chosen to be $u_3 + u_4$ where u_i is the steady state solution for $\beta_i = 1$, other $\beta_i = 0$.

5.5 Problem 2b

For this problem, we set the initial condition to the steady state solution for all $\beta_i = 1$. The boundary conditions correspond to

$$\beta_3 = \beta_4 = 0 \qquad \beta_1 = \beta_2 = \beta_5 = \beta_6 = 1.$$

The particular solution used to satisfy the boundary conditions is chosen to be the steady state solution for these values of β_i since this solution was available (it was used in the generation of the snapshots.) This setup is very similar to that that let to Tables 15 and 16 and, as expected, Tables 18 and 19 show that the results are also similar.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.87766E-05	0.98358E-05	0.16621E-01	0.23117E-01
0.2	0.78511E-05	0.85852E-05	0.14914E-01	0.20197E-01
0.3	0.70343E-05	0.76085E-05	0.13403E-01	0.17915E-01
0.4	0.63150E-05	0.68365E-05	0.12067E-01	0.16110E-01
0.5	0.56834E-05	0.62167E-05	0.10891E-01	0.14661E-01
1.0	0.35923E-05	0.43589E-05	0.69746E-02	0.10315E-01
2.0	0.25832E-05	0.28165E-05	0.51270E-02	0.67013E-02
3.0	0.26532E-05	0.22514E-05	0.53621E-02	0.53803E-02
4.0	0.26510E-05	0.21543E-05	0.54404E-02	0.51673E-02
6.0	0.21034E-05	0.21342E-05	0.44266E-02	0.51508E-02
8.0	0.20870E-05	0.22339E-05	0.44823E-02	0.54195E-02
10.0	0.46180E-05	0.39122E-05	0.10091E-01	0.95341E-02

Table 18: Errors for Problem # 2b using 8 POD basis functions generated from the 400 snapshots of Case 1. The particular solution is chosen to be the steady state for the boundary condition.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.48085E-06	0.74732E-06	0.91065E-03	0.17564E-02
0.2	0.32971E-06	0.49631E-06	0.62634E-03	0.11676E-02
0.3	0.23986E-06	0.36680E-06	0.45701E-03	0.86364E-03
0.4	0.19506E-06	0.30011E-06	0.37275E-03	0.70722E-03
0.5	0.17584E-06	0.25650E-06	0.33696E-03	0.60490E-03
1.0	0.14164E-06	0.18318E-06	0.27500E-03	0.43347E-03
2.0	0.19873E-06	0.26924E-06	0.39444E-03	0.64061E-03
3.0	0.12930E-06	0.13077E-06	0.26130E-03	0.31250E-03
4.0	0.84121E-07	0.10571E-06	0.17263E-03	0.25357E-03
6.0	0.17345E-06	0.21816E-06	0.36502E-03	0.52653E-03
8.0	0.10541E-06	0.99767E-07	0.22639E-03	0.24204E-03
10.0	0.37432E-06	0.14930E-06	0.41791E-03	0.60949E-03

Table 19: Errors for Problem 2b using 16 POD basis functions generated from the 400 snapshots of Case 1. The particular solution is chosen to be the steady state for the boundary condition.

5.6 Problem 2c

For this problem, the initial condition is set to the steady state solution corresponding to all $\beta_i = 1$. The boundary conditions correspond to the choices

$$\beta_3 = \beta_4 = 0.5 \qquad \beta_1 = \beta_2 = \beta_5 = \beta_6 = 0.0.$$

Again, for these choices, no steady state solution is available, so the particular solution used to satisfy the boundary conditions was chosen to be one-half the steady state solution corresponding to $\beta_3 = \beta_4 = 1.0$ and all other $\beta_i = 0$.

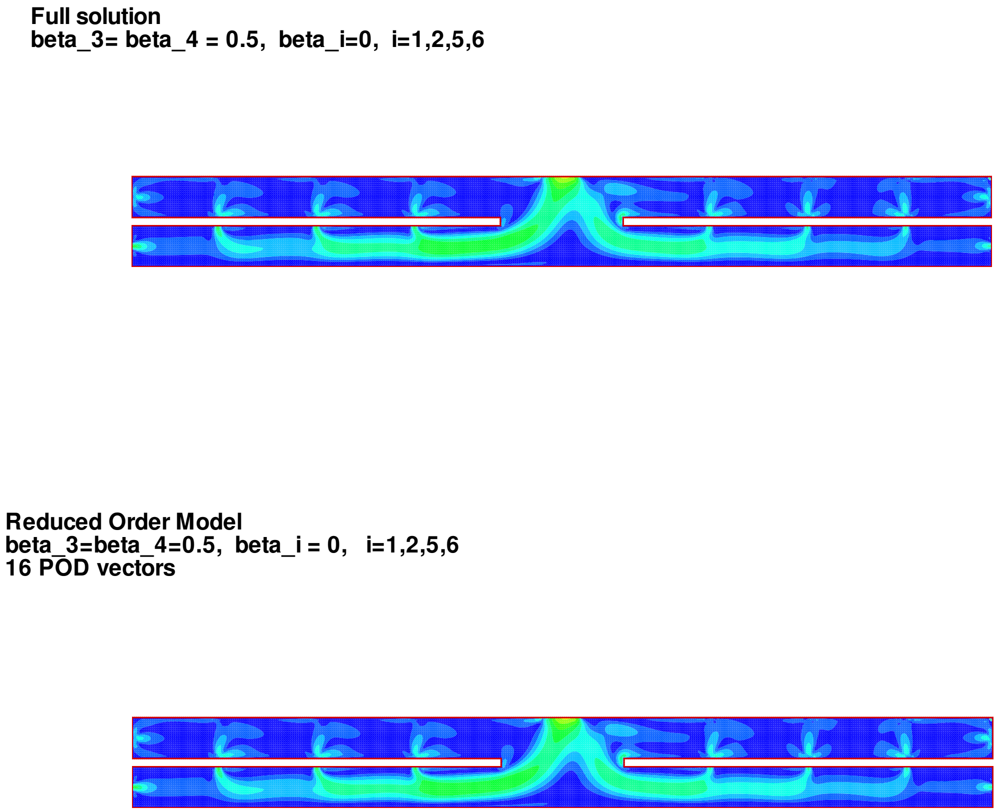
As is borne out by Tables 20 and 21, the results for this problem are generally very good. Also, we see from Figure 4 that, visually speaking, the full- and reduced-order solutions are very similar.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.61355E-05	0.45514E-05	0.010704	0.013327
0.2	0.52829E-05	0.39442E-05	0.0092386	0.011577
0.3	0.45460E-05	0.34553E-05	0.0079679	0.010166
0.4	0.39246E-05	0.30691E-05	0.0068939	0.0090499
0.5	0.34216E-05	0.27756E-05	0.0060233	0.0082023
1.0	0.26870E-05	0.24243E-05	0.0047778	0.0072367
2.0	0.43795E-05	0.37434E-05	0.0079219	0.011367
3.0	0.55791E-05	0.48866E-05	0.010238	0.015054
4.0	0.64171E-05	0.56981E-05	0.011922	0.017776
6.0	0.78196E-05	0.69956E-05	0.014832	0.022292
8.0	0.91790E-05	0.83173E-05	0.017708	0.026976
10.0	0.10732E-04	0.97833E-05	0.021005	0.032217

Table 20: Errors for Problem 2d using 8 POD basis functions generated from the 400 snapshots of Case 1.

t	$\|u_{rb}^h - u^h\|_0$	$\|v_{rb}^h - v^h\|_0$	$\frac{\|u_{rb}^h - u^h\|_0}{\|u^h\|_0}$	$\frac{\|v_{rb}^h - v^h\|_0}{\|v^h\|_0}$
0.1	0.31329E-06	0.45000E-06	0.54659E-03	0.13176E-02
0.2	0.44551E-06	0.51434E-06	0.77909E-03	0.15097E-02
0.3	0.62304E-06	0.62185E-06	0.10920E-02	0.18295E-02
0.4	0.80432E-06	0.74147E-06	0.14129E-02	0.21864E-02
0.5	0.98074E-06	0.86304E-06	0.17265E-02	0.25504E-02
1.0	0.17838E-05	0.14655E-05	0.31718E-02	0.43746E-02
2.0	0.32045E-05	0.27226E-05	0.57963E-02	0.82676E-02
3.0	0.44888E-05	0.39601E-05	0.82370E-02	0.12200E-01
4.0	0.56379E-05	0.50736E-05	0.10475E-01	0.15828E-01
6.0	0.75689E-05	0.68943E-05	0.14356E-01	0.21969E-01
8.0	0.91046E-05	0.82890E-05	0.17565E-01	0.26884E-01
10.0	0.10338E-04	0.94073E-05	0.20234E-01	0.30978E-01

Table 21: Errors for Problem 2d using 16 POD basis functions generated from the 400 snapshots of Case 1.



Full solution
beta_3= beta_4 = 0.5, beta_i=0, i=1,2,5,6

Reduced Order Model
beta_3=beta_4=0.5, beta_i = 0,   i=1,2,5,6
16 POD vectors

Figure 4: Comparison of full- and reduced-order flow solutions for Problem 2c at $t = 1$.

5.7 Problem 2d

For this problem, the initial condition is set to the steady state solution corresponding to all $\beta_i = 1$. The boundary conditions correspond to the choices

$$\beta_i = .5 \quad i = 1, \ldots, 6.$$

For this case, a steady state solution corresponding to this boundary condition setup is not available. So, the particular solution used to satisfy the boundary conditions is chosen to be one half the steady state solution corresponding to all $\beta_i = 1$. Four and eight-dimensional POD bases are used.

We use this example to test the relative performance of the snapshot sets determined according to Cases 1, 2, and 3 in §5.2. The results are given in Tables 22 to 29. We see that the results are generally very good, with percentage errors generally less than 4%. However, a key observation can be deduced from Tables 28 and 29: enlarging a snapshot set does not necessarily make the ROM (with a fixed number of basis vectors) better. From those tables, we see that generally the results for Case 2 (600 snapshots) are better than those for Case 1 (400 snapshots), but that the results for Case 3 (800 snapshots) using a 16-dimensional POD basis are also worse than those for Case 2. Looking at the results for Case 1 (400 snapshots) we also see that increasing the dimension of the POD basis does not necessarily result in appreciable improvements in accuracy.

The conclusions just drawn from the tables can also be clearly deduced from Figures 5 and 6. Also, we again see from Figure 7 that, visually speaking, the full- and reduced-order solutions are very similar.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.26539E-05	0.30206E-05	0.0045024	0.0078062
0.2	0.22515E-05	0.24997E-05	0.0038266	0.0064715
0.3	0.20324E-05	0.22216E-05	0.0034602	0.0057612
0.4	0.19962E-05	0.21404E-05	0.0034044	0.0055597
0.5	0.21092E-05	0.21946E-05	0.0036032	0.0057095
1.0	0.34656E-05	0.30711E-05	0.0059667	0.0080489
2.0	0.59394E-05	0.48187E-05	0.010365	0.012790
3.0	0.77161E-05	0.63028E-05	0.0103621	0.016911
4.0	0.91265E-05	0.75590E-05	0.016271	0.020475
6.0	0.11299E-04	0.95666E-05	0.020484	0.026335
8.0	0.12929E-04	0.11182E-04	0.023762	0.031199
10.0	0.14329E-04	0.12632E-04	0.026645	0.035657

Table 22: Errors for Problem 2c using 8 POD basis functions generated from the 400 snapshots of Case 1.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.41637E-06	0.40472E-06	0.00070638	0.0010459
0.2	0.74738E-06	0.64689E-06	0.0012702	0.0016747
0.3	0.10945E-05	0.93468E-06	0.0018634	0.0024239
0.4	0.14337E-05	0.12197E-05	0.0024452	0.0031681
0.5	0.17618E-05	0.14949E-05	0.0030097	0.0038891
1.0	0.32462E-05	0.27310E-05	0.0055891	0.0071573
2.0	0.56279E-05	0.47311E-05	0.0098216	0.012558
3.0	0.74615E-05	0.63264E-05	0.013171	0.016975
4.0	0.89207E-05	0.76339E-05	0.015904	0.020678
6.0	0.11136E-04	0.96475E-05	0.020188	0.026558
8.0	0.12801E-04	0.11138E-04	0.023527	0.031076
10.0	0.14431E-04	0.12760E-04	0.026122	0.035089

Table 23: Errors for Problem 2c using 16 POD basis functions generated from the 400 snapshots of Case 1.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.28171E-05	0.32734E-05	0.0047793	0.0084597
0.2	0.27292E-05	0.28686E-05	0.0046385	0.0074264
0.3	0.27872E-05	0.27120E-05	0.0047454	0.0070330
0.4	0.29453E-05	0.27298E-05	0.0050231	0.0070905
0.5	0.31628E-05	0.28497E-05	0.0054031	0.0074138
1.0	0.44294E-05	0.37641E-05	0.0076262	0.0098651
2.0	0.63058E-05	0.51509E-05	0.011004	0.013672
3.0	0.74804E-05	0.61909E-05	0.013205	0.016611
4.0	0.83273E-05	0.70679E-05	0.014847	0.019145
6.0	0.95497E-05	0.84694E-05	0.017312	0.023315
8.0	0.10510E-04	0.95599E-05	0.019317	0.026673
10.0	0.11546E-04	0.10551E-04	0.021470	0.029782

Table 24: Errors for Problem 2c using 8 POD basis functions generated from the 600 snapshots of Case 2.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.64363E-06	0.70238E-06	0.0010919	0.0018152
0.2	0.64925E-06	0.60029E-06	0.0011035	0.0015541
0.3	0.71790E-06	0.65675E-06	0.0012223	0.0017031
0.4	0.81128E-06	0.76569E-06	0.0013836	0.0019889
0.5	0.90914E-06	0.87701E-06	0.0015531	0.0022816
1.0	0.13092E-05	0.12751E-05	0.0022541	0.0033419
2.0	0.17389E-05	0.16470E-05	0.0030347	0.0043716
3.0	0.19172E-05	0.17751E-05	0.0033843	0.0047628
4.0	0.19793E-05	0.18503E-05	0.0035289	0.0050119
6.0	0.21164E-05	0.21160E-05	0.0038368	0.0058250
8.0	0.24248E-05	0.25230E-05	0.0044567	0.0070395
10.0	0.29232E-05	0.30219E-05	0.0054358	0.0085300

Table 25: Errors for Problem 2c using 16 POD basis functions generated from the 600 snapshots of Case 2.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.31088E-05	0.39181E-05	0.0052742	0.010126
0.2	0.29479E-05	0.35961E-05	0.0050102	0.0093098
0.3	0.29275E-05	0.34560E-05	0.0049843	0.0089623
0.4	0.30142E-05	0.34363E-05	0.0051406	0.0089256
0.5	0.31732E-05	0.34887E-05	0.0054209	0.0090762
1.0	0.43105E-05	0.40573E-05	0.0074214	0.010633
2.0	0.62012E-05	0.51750E-05	0.010822	0.013736
3.0	0.74366E-05	0.61600E-05	0.013127	0.016528
4.0	0.83319E-05	0.70143E-05	0.014855	0.019000
6.0	0.96045E-05	0.83708E-05	0.017412	0.023043
8.0	0.10579E-04	0.94326E-05	0.019444	0.026318
10.0	0.11619E-04	0.10419E-04	0.021606	0.029411

Table 26: Errors for Problem 2c using 8 POD basis functions generated from the 800 snapshots of Case 3.

t	$\|u^h_{rb} - u^h\|_0$	$\|v^h_{rb} - v^h\|_0$	$\frac{\|u^h_{rb}-u^h\|_0}{\|u^h\|_0}$	$\frac{\|v^h_{rb}-v^h\|_0}{\|v^h\|_0}$
0.1	0.10241E-05	0.12578E-05	0.0017374	0.0032507
0.2	0.11096E-05	0.11536E-05	0.0018859	0.0029864
0.3	0.12600E-05	0.11995E-05	0.0021452	0.0031107
0.4	0.14357E-05	0.13125E-05	0.0024486	0.0034093
0.5	0.16166E-05	0.14453E-05	0.0027617	0.0037602
1.0	0.24145E-05	0.20488E-05	0.0041571	0.0053695
2.0	0.34288E-05	0.29076E-05	0.0059838	0.0077174
3.0	0.39742E-05	0.34672E-05	0.0070154	0.0093029
4.0	0.42545E-05	0.37952E-05	0.0075853	0.010280
6.0	0.43757E-05	0.39410E-05	0.0079326	0.010849
8.0	0.42244E-05	0.37115E-05	0.0077641	0.010356
10.0	0.41780E-05	0.34796E-05	0.0077691	0.0098220

Table 27: Errors for Problem 2c using 16 POD basis functions generated from the 800 snapshots of Case 3.

t	*POD using 400 snapshots*		*POD using 600 snapshots*		*POD using 800 snapshots*	
	8	16	8	16	8	16
0.1	.45%	.071%	.48%	.11%	.53%	.17%
0.5	.36%	.30%	.54%	.16%	.54%	.28%
1.0	.60%	.56%	.76%	.23%	.74%	.42%
2.0	1.0%	.98%	1.1%	.30%	1.1%	.60%
4.0	1.6%	1.6%	1.5%	.35%	1.5%	.76%
6.0	2.0%	2.0%	1.7%	.38%	1.7%	.79%
8.0	2.4%	2.4%	1.9%	.45%	1.9%	.78%
10.0	2.7%	2.6%	2.1%	.54%	2.2%	.78%

Table 28: Comparison of percentage errors in the x-component of velocity for Problem 2c using different snapshot sets.

t	*POD using 400 snapshots*		*POD using 600 snapshots*		*POD using 800 snapshots*	
	8	16	8	16	8	16
0.1	.78%	.10%	.84%	.18%	1.0%	.33%
0.5	.57%	.39%	.74%	.23%	.91%	.38%
1.0	.80%	.72%	.98%	.33%	1.1%	.54%
2.0	1.3%	1.3%	1.4%	.44%	1.4%	.77%
4.0	2.0%	2.1%	1.9%	.50%	1.9%	1.0%
6.0	2.6%	2.7%	2.3%	.59%	2.3%	1.1%
8.0	3.1%	3.1%	2.7%	.70%	2.6%	1.0%
10.0	3.6%	3.5%	3.0%	.85%	2.9%	.98%

Table 29: Comparison of percentage errors in the y-component of velocity for Problem 2c using different snapshot sets.

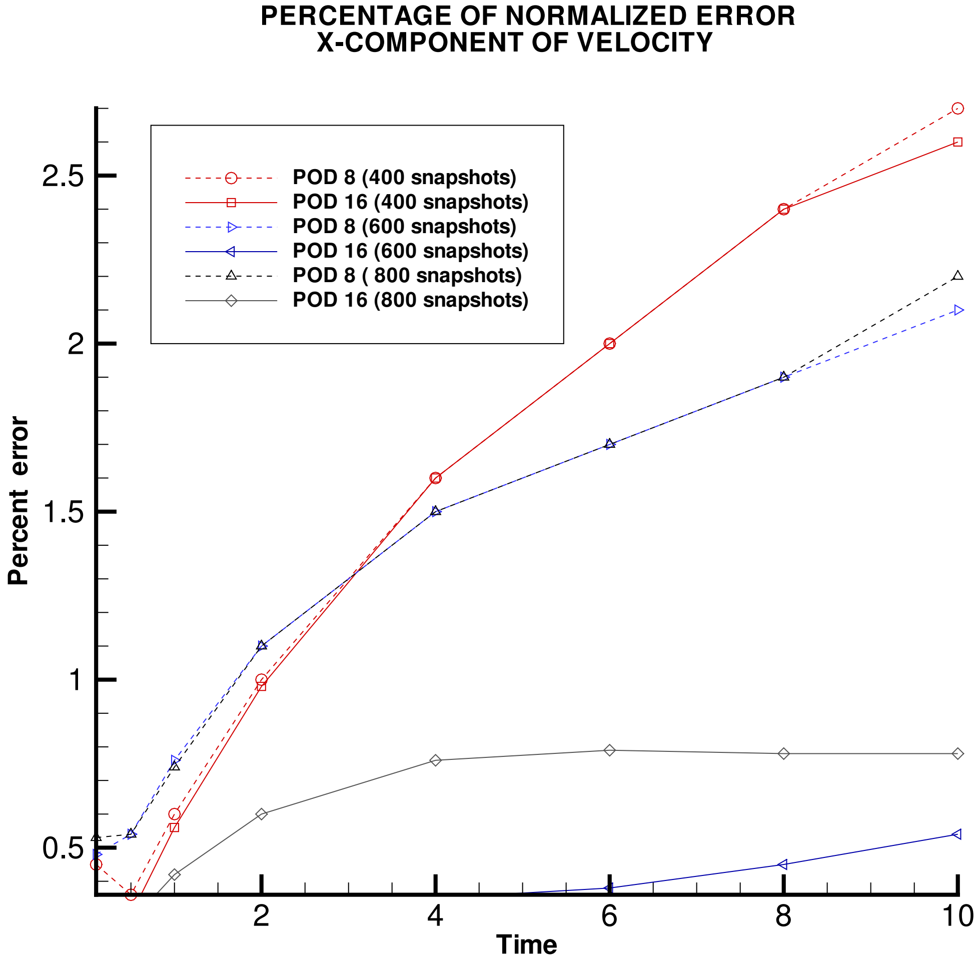


Figure 5: Percentage relative error in the x-component of the velocity for Problem 2d corresponding to different snapshot sets and different POD-basis dimension.

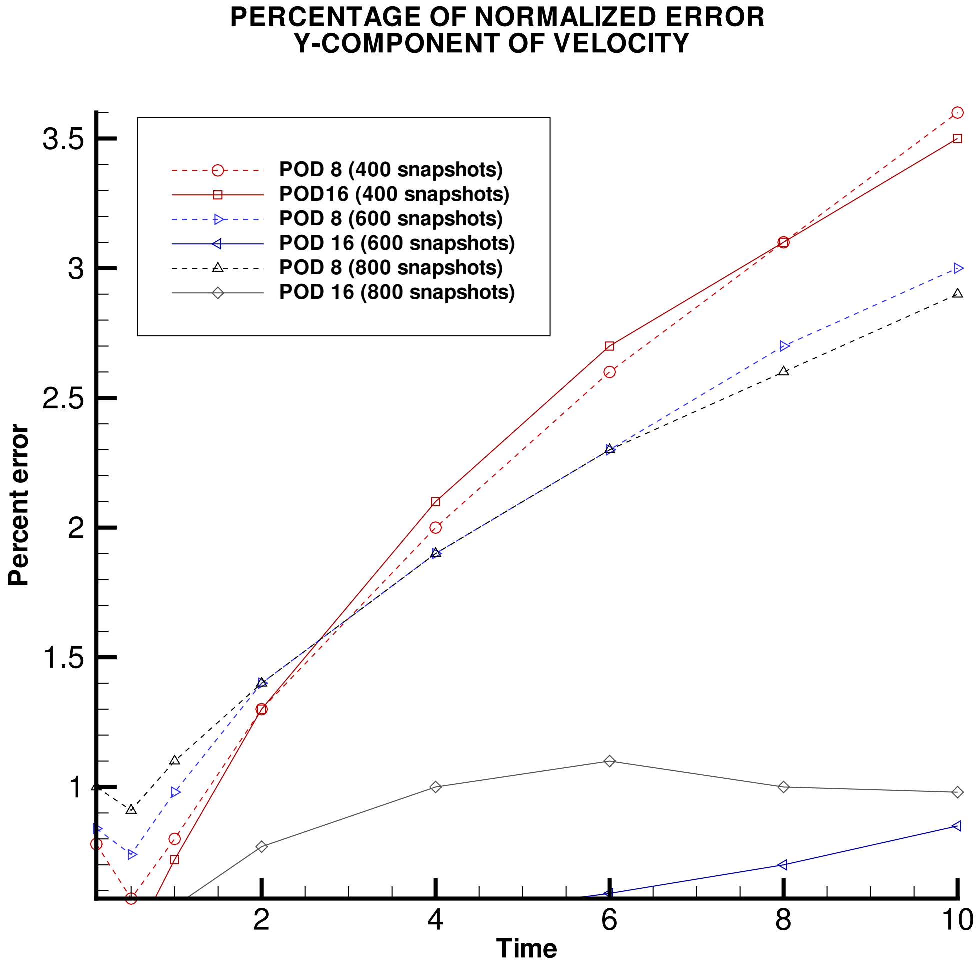
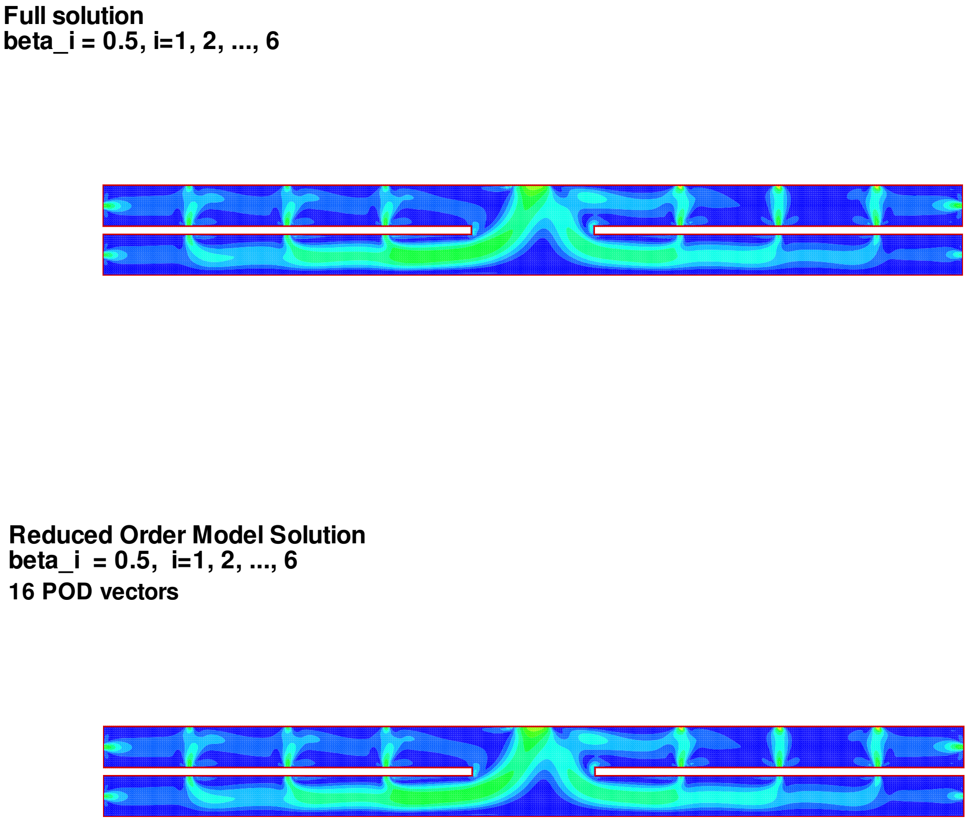


Figure 6: Percentage relative error in the y-component of the velocity for Problem 2d corresponding to different snapshot sets and different POD-basis dimension.



Full solution
beta_i = 0.5, i=1, 2, ..., 6

Reduced Order Model Solution
beta_i = 0.5, i=1, 2, ..., 6
16 POD vectors

Figure 7: Comparison of full- and reduced-order flow solutions for problem 2d at $t = 1$.

5.8 Problem 2e

For this problem, the initial condition is set to the steady state solution corresponding to all $\beta_i = 1$. The boundary conditions correspond to the "random" choices

$$\beta_3 = \beta_4 = 0.65 \qquad \beta_1 = \beta_2 = \beta_5 = \beta_6 = 0.15.$$

We use this example to further investigate the effects on the ROM simulation resulting from different choices for the particular solution used to satisfy the boundary conditions. The different choices are determined as follows.

- Method 1: the particular solution is the steady state for $\beta_3 = \beta_4 = 0.65$, $\beta_1 = \beta_2 = \beta_5 = \beta_6 = 0.15$.
- Method 2: the particular solution is 0.65$*$(steady state for $\beta_3 = \beta_4 = 1$, $\beta_1 = \beta_2 = \beta_5 = \beta_6 = 0$) + 0.15$*$(steady state for $\beta_3 = \beta_4 = 0$, $\beta_1 = \beta_2 = \beta_5 = \beta_6 = 1$).
- Method 3: the particular solution is 0.65$*$(steady state solution for $\beta_3 = 1$, $\beta_i = 0$) + 0.65$*$(steady state solution for $\beta_4 = 1$, $\beta_i = 0$) + 0.15$*$(steady state solution for $\beta_3 = \beta_4 = 0$, $\beta_1 = \beta_2 = \beta_5 = \beta_6 = 1$).

Note that the steady state solution used in Method 1 is not really available, while those used in Method 2 are available from the snapshot generation process. Those for Method 3 could be available since the involve corner values of the β_i's.

Table 30 and Figure 8 provide results for the three methods of choosing the particular solution; 8-dimensional POD bases were used for these results. If 16 POD vectors are used, the errors using Method 1 decrease as before. Using Method 2, the errors reduce somewhat and using Method 3, the errors remain about the same. Also, we again see from Figure 9 that, visually speaking, the full- and reduced-order solutions are very similar.

t	*Method 1*		*Method 2*		*Method 3*	
	u	v	u	v	u	v
0.1	1.1%	1.6%	2.5%	3.9%	7.0%	14.3%
0.5	.81%	1.3%	2.5%	3.9%	7.0%	14.3%
1.0	.75%	1.2%	2.6%	3.9%	6.9%	14.4%
2.0	.77%	1.1%	2.7%	4.0%	7.0%	14.7%
4.0	.78%	1.1%	2.7%	4.2%	7.1%	15.3%
6.0	.75%	1.2%	2.7%	4.3%	7.2%	15.8%
8.0	.79%	1.4%	2.7%	4.5%	7.2%	16.3%
10.0	.97%	1.7%	2.8%	4.8%	7.3%	16.7%

Table 30: Percent errors for Problem 2e using 8 POD basis functions generated from 600 snapshots using all but four random β_i. The particular solution is chosen in three different ways.

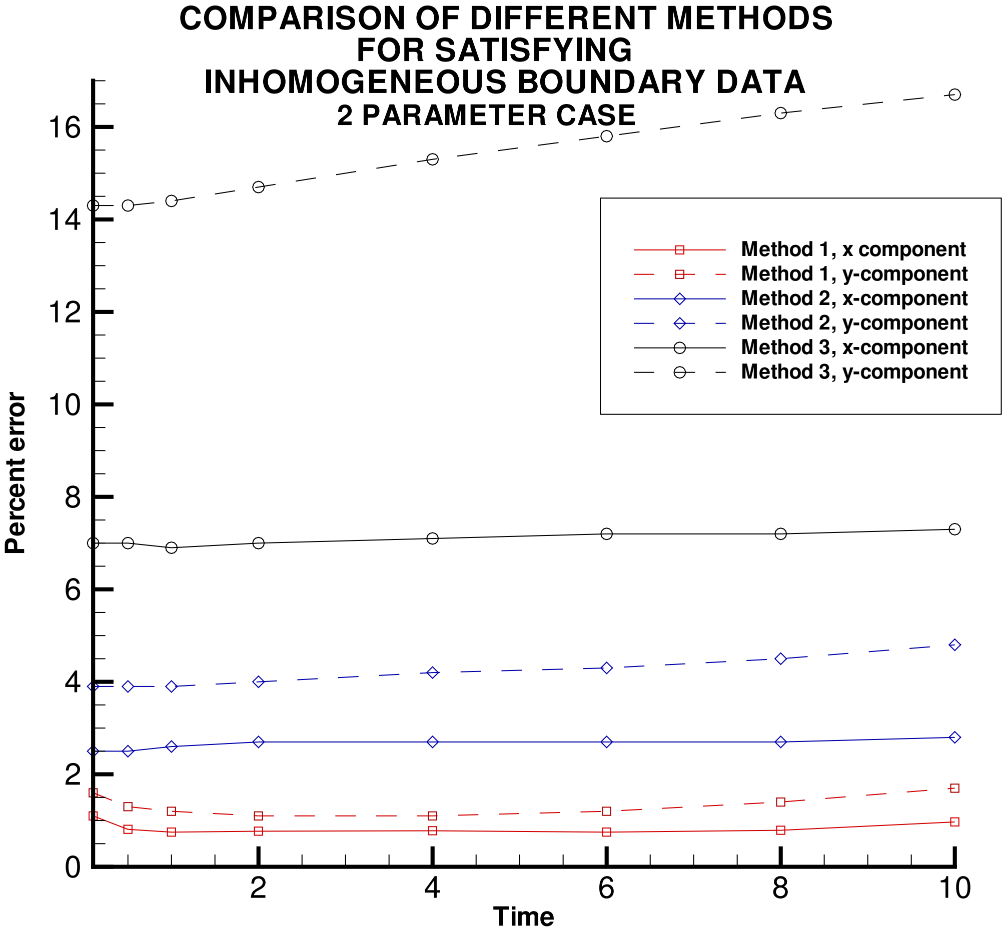


Figure 8: Comparison of the use of different particular solutions for satisfying inhomogeneous boundary conditions for Problem 2e.

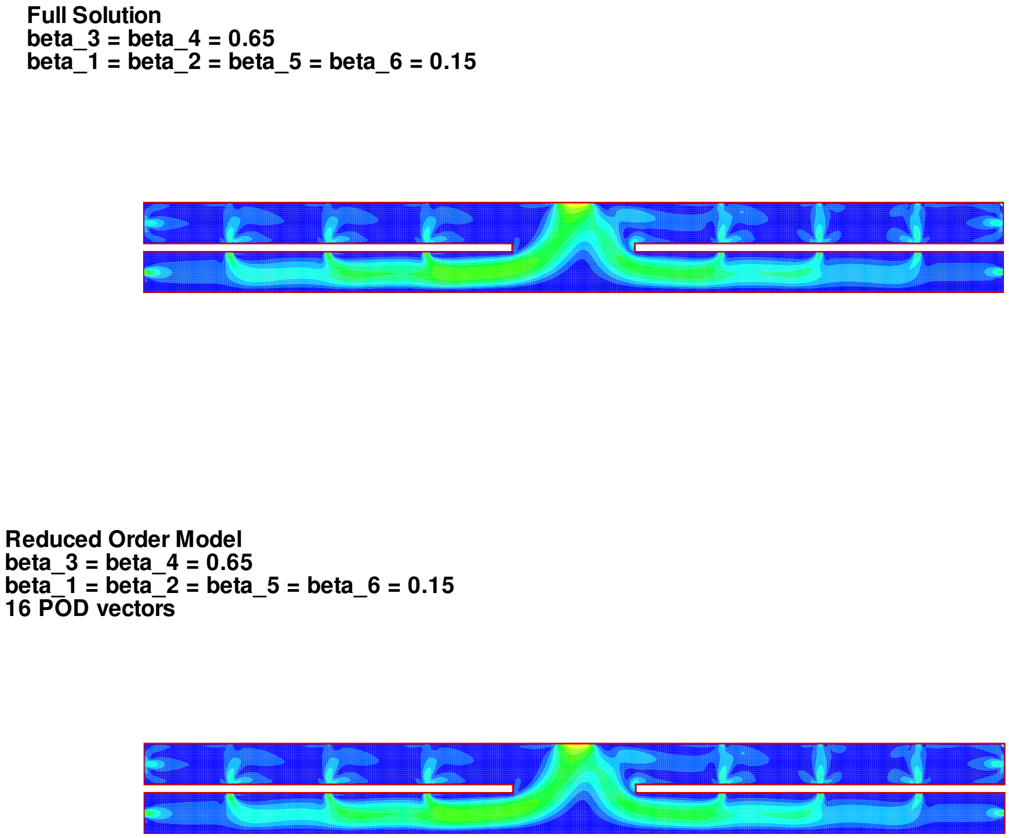


Figure 9: Comparison of full- and reduced-order flow solutions for Problem 2e at $t = 1$.

6 Concluding remark

A general conclusion from the five test problems is that the more the steady state solutions used to define a particular solution has to do with the boundary conditions being used in the simulations, the better the results. Of course, since such steady states are not known for general choices of boundary conditions, the best possible case, e.g., Method 1 in Table 30 and Figure 8, is not always realizable. However, it seems possible that acceptable results can still be obtained as in, e.g., Method 2 in Table 30 and Figure 8. Certainly further explorations are needed, as is the testing of the second way of imposing boundary conditions that does not require the definition of particular solutions; see page 3.