Comparison of Model Reduction Techniques on High-Fidelity Linear and Nonlinear Electrical, Mechanical, and Biological Systems

Matthew J. Zahr
University of California, Berkeley

September 27, 2010

Abstract

This document presents the results of a testbed developed for the comparison of model reduction techniques on large-scale linear and nonlinear, static and dynamic systems. The current capabilities of the testbed allow the user to compare eight model reduction methods on three linear, dynamical systems and three reduction techniques on five nonlinear systems (four dynamic and one static).

1 Introduction

As engineers and scientists strive to simulate increasingly large and complex real-world systems, there comes a point when their ambition is restricted by the speed and memory capacity of modern computers. Computational restrictions are more pronounced when real-time applications are involved. Attempts to lessen these restrictions have led to methods to reduce the size of the model at hand. The model reduction is achieved by projecting the full order model onto a subspace in which solutions are sought.

The goal of model order reduction (MOR) is to solve high-fidelity models much faster, while maintaining the accuracy of the solution. An important application of reduced order models (ROMs) is an area called "deployed" analysis, wherein computer-based models are used in real-time, in-the-field applications. An example would be using real-time flight conditions to calculate the optimal speed and altitude to minimize the drag on an aircraft. Another application is in control, wherein computer models are used to control the behavior of some system, such as an unmanned aircraft. Nondestructive evaluation/parameter estimation is the final application of model reduction that will be mentioned in this document, but there are many others. In this application, damaged components of a complex body are identified unintrusively. This is generally done by applying a disturbance to both the model and the physical system; then an inverse problem is solved wherein problematic areas of the body are identified by matching the responses of the two systems. All of the applications above are very dependent on having the ablility to solve the governing equations very quickly, while only introducing marginal error.

Many different methods for choosing an appropriate search subspace exist for linear dynamical systems, which include Proper Orthogonal Decomposition (POD), Balanced POD, Balanced Truncation, and Krylov Methods. Since nonlinear systems are more difficult and less well-understood than linear systems, the only common model reduction methods are Galerkin and Petrov-Galerkin POD.

In practice, it is highly unlikely that one particular model reduction technique will be equally well-suited for all systems. It is more likely that certain techniques will be optimal for one class of problems, but not for others. The aim of this document is to investigate which techniques are best-suited for the example systems chosen.

To properly study the systems in this paper, the governing equations were spatially discretized using either finite differences or the finite element method. The problems were then discretized in time using finite differences and evolved over all time steps using the backward Euler integration scheme. Subsequent sections present the governing equations (continuous) of each problem and the discreted form of the equation or a reference to it.

Since linear and nonlinear systems are treated quite differently, this document analyzes them separately. Linear systems will be treated in Section 2 and nonlinear systems in Section 3 . The linear and nonlinear sections will introduce the model reduction techniques, the governing equations of the example problems, and the results of the analysis of the full order model. Section 4 will present the results from the model reduction comparison.

2 Linear Systems

2.1 Model Reduction Methods

A number of model reduction techniques have been developed for linear dynamical systems, but the scope of this paper is limited to the following reduction methods:

1. POD in the Time Domain
2. Weighted POD in the Time Domain
3. POD in the Frequency Domain
4. Weighted POD in the Frequency Domain
5. Balanced POD
6. Balanced Truncation
7. Moment Matching via Lanzcos Algorithm
8. Moment Matching via Arnoldi Algorithm

The reader is directed to [2], [1], and [7] for additional information on the above reduction methods.

2.2 Problem Formulations

All of the systems defined in this section are linear dynamical systems since they can be written in the following form:

$$\begin{cases} \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + Bu(t) \\ y(t) = C\mathbf{x}(t) + Du(t) \end{cases} \tag{1}$$

The scope of this paper is limited to single-input, single-output systems (SISO), therefore, $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^N$, $C^T \in \mathbb{R}^N$, $D \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^N$, and $y \in \mathbb{R}$. Furthermore, this paper assumes $D = 0$ for simplicity. Throughout this paper, $\mathbf{x}$(t) will be used to denote the state vector at time t; not to be confused with $\hat{\mathbf{x}}$, which will represent generalized coordinates of the problem (i.e. x in 1D, $[x, y]^T$ in 2D, and $[x, y, z]^T$ in 3D).

Multiple Mass-Spring-Damper System

The mass-spring-damper (MSD) configuration analyzed is illustrated in Figure 1. The configuration consists of N degrees of freedom (DOF) where each DOF has the same triplet of parameters (mass, damping, stiffness). The output of interest in this problem is the velocity of the last mass. The system has 250 masses, where all masses are initially at rest and have an initial displacement of +2 units. There are two reasons this system was choosen: 1) Distinct spikes in the Bode Plot make it a difficult problem to solve with a reduced model and 2) the mass-spring-damper system is well-understood and has become a canonical engineering problem. The governing equations for the multiple mass-spring-damper system is presented in (2).

$$M\ddot{\mathbf{x}} + C\dot{\mathbf{x}} + K\mathbf{x} = \mathbf{F} \tag{2}$$

$M \in \mathbb{R}^{N\times N}$ is the a diagonal matrix whose (i, i) entry contains the mass of block i. $C \in \mathbb{R}^{N\times N}$ is the damping matrix and $K \in \mathbb{R}^{N\times N}$ is the stiffness matrix of the system; both of which are symmetric. As previously mentioned, all blocks have the mass (m), stiffness coefficient (k), and damping coefficient (d). It is more intuitive to define the characteristics of each block in terms of the mass, damping ratio (ζ), and natural frequency (ω_0) using the following relationships.

$$\omega_0 = \sqrt{\tfrac{k}{m}}$$

$$\zeta = \frac{d}{2m\omega_0}$$

Therefore, the parameter triplet used for this system is $(m, \zeta, \omega_0) = (5, 0.2, 2)$. The time history and Bode plots for the full order model are included in Figures 2a and 2b.

Figure 1: Mass-Spring-Damper Configuration (N = 250)

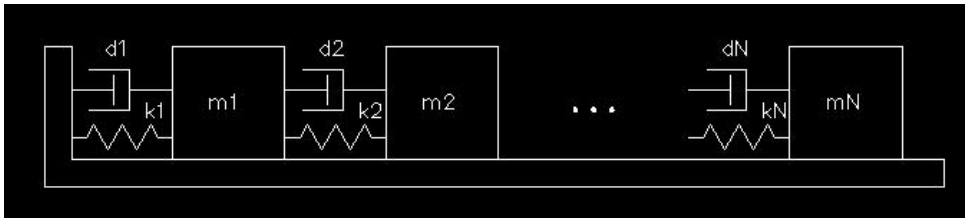
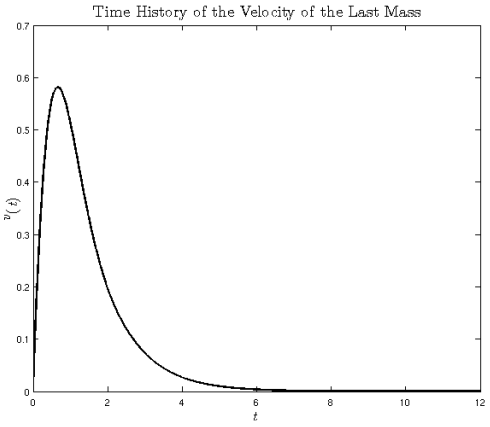
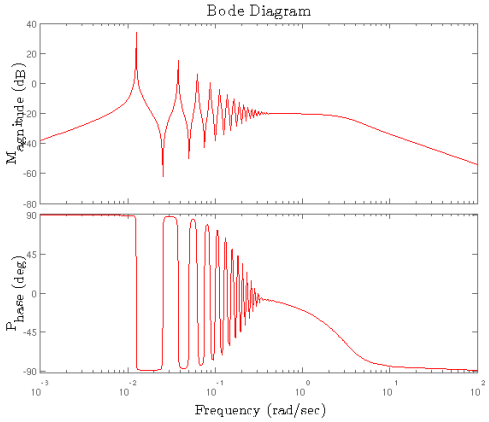


Figure 2: Full Order Analysis of Mass-Spring-Damper System

(a) Time History for Full Order MSD System



(b) Bode Plot for Full Order MSD System



Penzl Example

The so-called Penzl example is a numerical experiment investigated in [5] with supporting MATLAB code provided by David Amsallem. There are 1006 DOFs in the full order system. Unlike the other two linear dynamical systems , it is not directly linked to a physical phenomena. The initial condition is a vector of ones. For additional information, the reader is referred to [5].

Since this problem has no physical relevance, this paper will only use the Bode plot for the output of the system. The Bode plot for the full order model is in Figure 3.

Transient Heat Flow

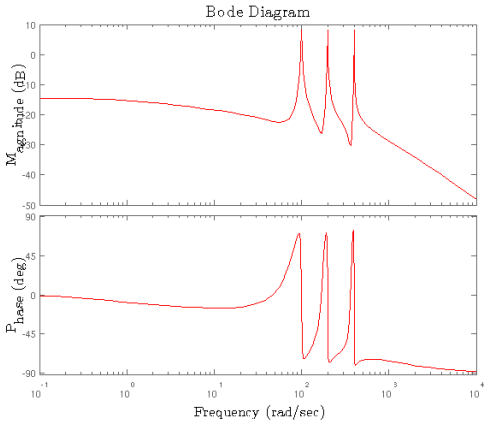
The final linear dynamical system investigated is 2D transient heat flow through a rectangular plate. The complete strong formulation of the heat transfer problem, including initial and boundary conditions, is provided in (3) - (8).

$$\rho(\hat{\mathbf{x}})c(\hat{\mathbf{x}})\frac{\partial T(\hat{\mathbf{x}};t)}{\partial t} = \nabla \cdot k(\hat{\mathbf{x}})\nabla T(\hat{\mathbf{x}};t) + f(\hat{\mathbf{x}};t) \tag{3}$$

$$T(\hat{\mathbf{x}};0) = T_0 \quad \forall \hat{\mathbf{x}} \in \Omega \tag{4}$$

$$T(\hat{\mathbf{x}},t) = \bar{T} \quad \forall \hat{\mathbf{x}} \in \Gamma_T \tag{5}$$

Figure 3: Bode Plot for Full Order Penzl Model



$$\mathbf{q}(\hat{\mathbf{x}};\mathbf{t}) \cdot \mathbf{n} = \bar{q} \quad \forall \hat{\mathbf{x}} \in \Gamma_Q \tag{6}$$

$$\mathbf{q}(\hat{\mathbf{x}};t) \cdot \mathbf{n} = \sigma\epsilon(\hat{\mathbf{x}};t)(T(\hat{\mathbf{x}};t)^4 - T_\infty^4) \quad \forall \hat{\mathbf{x}} \in \Gamma_R \tag{7}$$

$$\mathbf{q}(\hat{\mathbf{x}};t) \cdot \mathbf{n} = \beta(T(\hat{\mathbf{x}};t) - T_\infty) \quad \forall \hat{\mathbf{x}} \in \Gamma_H \tag{8}$$

where $\Omega \in [0,1]^2$ (meters) is the domain and $\partial\Omega = \Gamma_T \cup \Gamma_Q \cup \Gamma_R \cup \Gamma_H$ is the boundary of the domain.

Since the scope of this section is restricted to linear systems, the radiation boundary conditions in (7) will be ignored. The rectangular plate is a non-homgeneous plate consisting of aluminum with a hole of radius $0.25m$ cut out of the center and replaced with brass. Furthermore, there are two energy sources in the body of the plate at the midpoint of the upper left and lower right quadrants and one energy sink at the center of the plate. The energy sources add 10^6 J to the plate, while the energy sink extracts 10^6 J. The parameters of the problem are: $T_\infty = 300$ K, $\bar{q} = 0$, $\bar{T} = 500$ K, $T_0 = 500$ K, $\beta = 45$, $\rho_{Al} = 2700$ kg/m^3, $\rho_{Brass} = 8700$ kg/m^3, $c_{Al} = 910J/K$, and $c_{Brass} = 377J/K$. Finally, the boundary conditions are: $\Gamma_T = [0,y]^T$, $\Gamma_Q = [1,y]^T$, $\Gamma_H = [x,0]^T \cup [x,1]^T$.

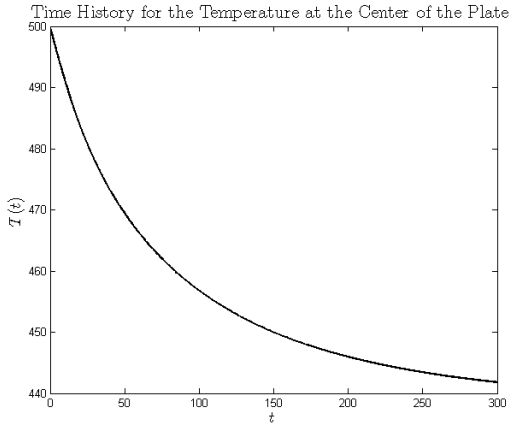
The complete weak formulation of the heat transfer equation is given in (9) and (10). It was discretized in space using the finite element method to yield a linear dynamical system in the form of (1). The output of this system is the temperature at the center of the plate. There were 30 4-node quadrilateral elements used in each dimension in the finite element mesh, so the full order model has 930 degrees of freedom after application of Dirichlet boundary conditions.

$$\int_\Omega (w(\hat{\mathbf{x}})\rho(\hat{\mathbf{x}})c(\hat{\mathbf{x}})\frac{\partial T(\hat{\mathbf{x}};t)}{\partial t} + \nabla w(\hat{\mathbf{x}}) \cdot k(\hat{\mathbf{x}})\nabla T(\hat{\mathbf{x}};t) - wf(\hat{\mathbf{x}};t))d\Omega = 0 \tag{9}$$

$$w(\hat{\mathbf{x}}) \in \mathcal{S} = \{w(\hat{\mathbf{x}}) \mid w(\hat{\mathbf{x}}) = 0 \quad \forall \hat{\mathbf{x}} \in \Gamma_T\} \tag{10}$$

The time history of the output for the full order model is presented in Figure 4.

Figure 4: Output for Full Order Heat Flow Model



3 Nonlinear Systems

3.1 Model Reduction Methods

As previously mentioned, nonlinear systems are more complicated and less well-understood than linear systems. Therefore, there are a limited number of model reduction techniques for nonlinear systems. The two reduction methods investigated in this documents are:

1. Galerkin POD
2. Petrov-Galerkin POD

Numerical solutions to nonlinear, unsteady partial differenetial equations involves performing Newton iterations at each time step. This requires computing the full residual and Jacobian of the nonlinear terms $n \times \bar{\tau}$ times, where n is the number of time steps and $\bar{\tau}$ is the average number of Newton iterations over all time steps. Evaluation of the full residual and Jacobian is very expensive for large systems, so the necessity for model reduction is much more pronounced for nonlinear problems.

Model reduction on nonlinear systems amounts to reducing the dimension of the nonlinear system and performing Newton iterations on the reduced system. However, computing the full residual and Jacobian and performing required matrix multiplication with the reduced basis bears cost that scales with the size of the FOM. Therefore, the resulting reduced model may offer only marginal cost saving and can even be more expensive. To actually improve performance, another level of approximation is necessary, which can be acheived using Trajectory Piece-Wise Linear (TPWL) approximation or a "Gappy POD" method.

TPWL attempts to approximate the solution to the full nonlinear system by linearizing the trajectory of the solution. At a high level, this method involves selecting points along the trajectory to linearize about and then solving the resulting linear systems using one of the techniques from Section 2. Additional information on this topic can be found in [6].

Gappy POD improves performance by only requiring the computation of certain rows of the residual and Jacobian. Gappy-like methods are an advanced topic that are beyond the scope of this paper, so the interested reader is referred to [3].

The remainder of this document will make use of the terms "training" input and "online" input. Training input refers to the input parameter that was used to generate the snapshots during the offline calculations. The online input is the input used during the online computations. A majority of the analysis in the document only considers the case where the training and online inputs are the same. Section 4 will test the robustness of nonlinear model reduction methods by using online inputs that differ from the training input.

3.2 Problem Formulations

This paper uses six nonlinear systems to demonstrate the efficiency and robustness of the model reduction techniques discussed above. The nonlinear systems in this section include: 1) 1D Burger's Equation, 2) Nonlinear Transmission Line, 3) FitzHugh Nagumo Equation, 4) Transient Heat Flow with Radiation BCs, 5) Micromachined Device, and 6) A Highly Nonlinear 2D Steady State Problem. All six nonlinear systems are formulated in this section and the results of the full order models are presented.

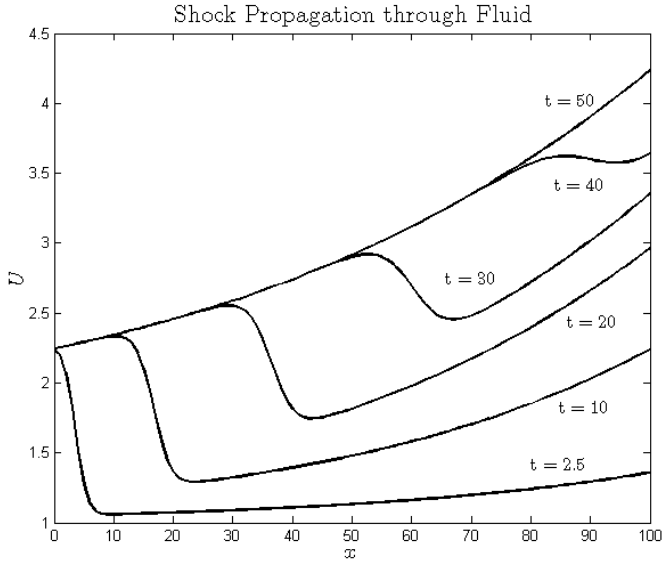
1D Shock Propagation through Fluid

In this system, Burger's Equation is used to model shock propagation through a fluid. In (11), U denotes the unknown conserved quantity (mass, density, heat, etc.) and the source term is chosen to be: $g(x) = 0.2e^{0.02x}$.

$$\frac{\partial U(x;t)}{\partial t} + \frac{\partial f(U(x;t))}{\partial x} = g(x) \tag{11}$$

With initial condition $U(x,0) = 1, \quad \forall x \in [0, L]$ and boundary condition $U(0,t) = u(t), \quad t > 0$. The nonlinearity of this problem arises from $f(U(x;t)) = 0.5U^2$. The incoming flow is taken as $u(t) = \sqrt{5}$ and the length of the domain is $L = 100$. The full order model has 101 unknowns. The output of this model is the conserved quantity U at $t = [2.5, 10, 20, 30, 40, 50]$. The results from the full order analysis of this problem are presented in Figure 5. For more information on the formulation of this problem or for the discretized form the governing equation, the reader is directed to [6].

Figure 5: Full Order Model of Burger's Equation



Nonlinear Transmission Line Circuit

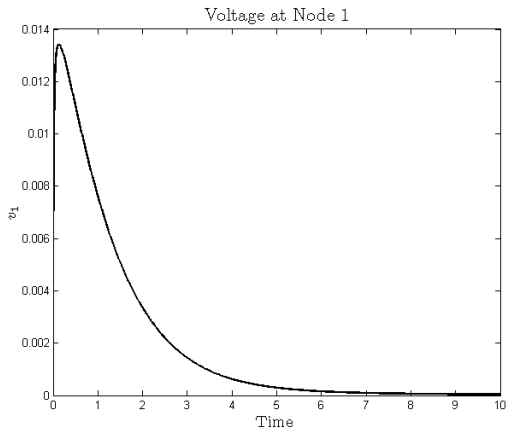
The nonlinear transmission line citcuit system was also considered in [6]. The circuit consists of linear resistors and capacitors and nonlinear diodes with a constitutive equation $i_d(v) = e^{40v} - 1$, where v is the voltage between the diode's terminals. All capacitors and resistors have unit values, current sources $u(t) = i(t) = \frac{1}{2}\left(\cos\frac{2\pi t}{10} + 1\right)$ and $i(t) = e^{-t}$ are considered, and the output is the voltage at node 1. The full order model consists of 100 nodes. This problem is similiar to the MSD problem in that it is naturally discretized, since the unknowns are the voltage at each node. The problem just described yields the following governing equations:

$$\begin{cases} \dfrac{dx}{dt} = f(x) + Bu \\ y = C^T x \end{cases} \tag{12}$$

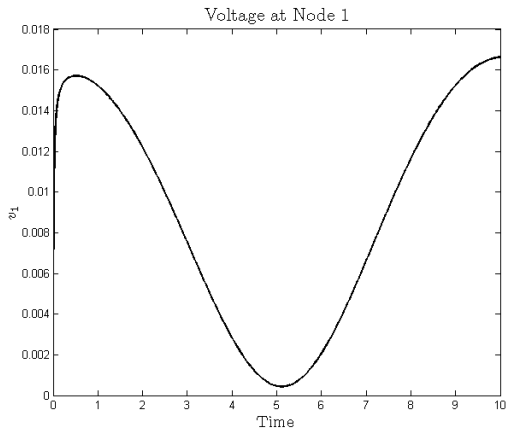
The results of the full order model are presented in Figure 6. For more information on the problem formulation, including the nonlinear function $f(v)$ and the discretized equation, see [6].

Figure 6: Full Order Model of Nonlinear Transmission Line System

(a) Input Current: $i(t) = e^{-t}$



(b) Input Current: $i(t) = \frac{1}{2}\left(\cos\frac{2\pi t}{10} + 1\right)$



1D Model for Neuron Activity in the Brain

The FitzHugh-Nagumo equations are used for the prediction of neuronal exitability and spike-generating mechanisms. It is a simplied version of the Hodgkin-Huxley model, which is considered one of the most realistic and physically sound models related to neuronal activity. This paper analyzes the FitzHugh-Nagumo equations for simplicity and convenience with regard to presentation of results. The governing equations for this system are presented in (13) and (14). The model consists of two coupled equations in two unknown variables, voltage (v) and voltage recovery (w). The full order model consists on 1024 unknowns (512 voltages and 512 voltage recoveries).

$$\epsilon\frac{\partial v(x,t)}{\partial t} = \epsilon^2\frac{\partial^2 v(x,t)}{\partial x^2} + v(x,t)(v(x,t)+\alpha)(\beta - v(x,t)) - w(x,t) + c \tag{13}$$

$$\frac{\partial w}{\partial t} = bv(x,t) - \gamma w(x,t) + c \tag{14}$$

With initial condition: $v(x,0) = w(x,0) = 0 \quad \forall x \in [0,L]$ and boudary conditions: $\frac{\partial v(0,t)}{\partial x} = -i_0(t)$, $\frac{\partial v(L,t)}{\partial x} = 0 \quad \forall t \geq 0$. The parameters in this problem take the following values: $L = 1, \epsilon = 0.015, b = 0.5, c = 0.05, \alpha = -0.1, \beta = 1, \gamma = 2,$ and $i_0(t) = 50000t^3e^{-15t}$.

The spatially discretized form of this problem was derived using finite differences and presented in equations (15) - (19).

$$\mathbf{v} = \begin{bmatrix} v_1 & v_2 & \dots & v_N \end{bmatrix}^T \tag{15}$$

$$\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \dots & w_N \end{bmatrix}^T \tag{16}$$

$$\mathbf{c} = \begin{bmatrix} c & c & \dots & c \end{bmatrix}^T \tag{17}$$

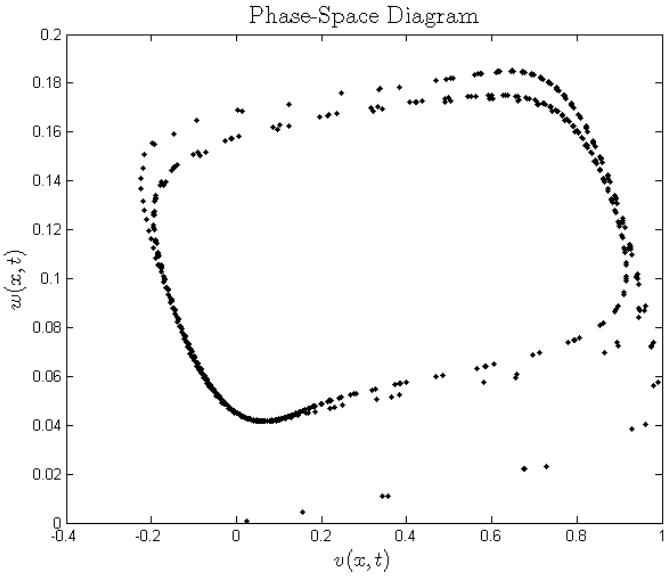
$$\begin{bmatrix} \mathbf{v}_{,t} \\ \mathbf{w}_{,t} \end{bmatrix} = \begin{bmatrix} \frac{\epsilon}{(\Delta x)^2}\mathbf{A} & -\frac{1}{\epsilon}\mathbf{I} \\ b\mathbf{I} & -\gamma\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} + \frac{\epsilon}{(\Delta x)^2}\begin{bmatrix} -(\Delta x)i_0(t) \\ \mathbf{0} \end{bmatrix} + \frac{1}{\epsilon}\begin{bmatrix} f(\mathbf{v}) \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \frac{1}{\epsilon}\mathbf{c} \\ \mathbf{c} \end{bmatrix} \tag{18}$$

$$\mathbf{A} = \begin{bmatrix} -1 & 0 & & & (\mathbf{0}) \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ (\mathbf{0}) & & & 0 & -1 \end{bmatrix} \tag{19}$$

where $\mathbf{c} \in \mathbb{R}^N$, $\mathbf{I}$ is the identity matrix $\in \mathbb{R}^{N \times N}$, and $\mathbf{A} \in \mathbb{R}^{N \times N}$.

Since this system has two unknown variables, which are both functions of position and time, it is conveninent to choose a phase-space diagram as the output. Other outputs that are considered are $v(x, t = 4s)$, $v(x = 0.5, t)$, $w(x, t = 4s)$, and $w(x = 0.5, t)$. The result of the full order model are in Figures 7 and 8. Additional information on this problem can be found in [4].

Figure 7: Phase Diagram for Full Order Model of FitzHugh-Nagumo Equations



Transient Heat Flow with Radiation Boundary Conditions

The nonlinear, transient heat flow problem has the same governing equations as the linear, transient heat flow from (3) - (10). The nonlinearity arises from radiation boundary conditions in (7) that will now be included. The example problem choosen will consist of a homogeneous aluminum plate with the same domain characteristics as the linear heat flow problem from Section 2. The plate is initially at a uniform temperature of $950K$, the boundary conditions are: $\Gamma_R = [x, 1]^T \cup [1, y]^T$ and $\Gamma_Q = [x, 0]^T \cup [0, y]^T$, and the body energy term is $f(\hat{\mathbf{x}}, t) = 10^{10} \sin\left(\frac{\pi}{10} t\right)\ J$ applied at the center of the lower left and upper right quadrants of the plate. This problem was also solved using a 30 x 30 mesh of 4-node quadrilateral elements; however, there are 961 degrees of freedom due to the lack of a Dirichlet boundary condition. The solution of the FOM for this problem is presented in Figure 9.

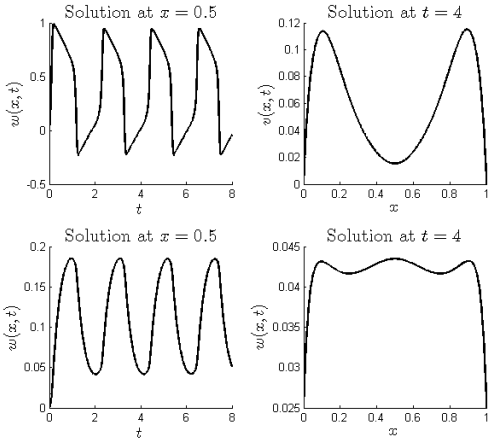
A Micromachined Device

Due to the complexity of this problem and the large number of parameters, the reader will be referred to [6] for background on this problem. Two important points to consider are: 1) the full order model has 880 degrees of freedom and 2) the input voltage for the problem is $v(t) = 9H(t)$, where $H(t)$ is the step function.

$$EI\frac{\partial^4 u}{\partial x^4} - S\frac{\partial^2 u}{\partial x^2} = F_{elec} + \int_0^w (p - p_a)dy - \rho\frac{\partial^2 u}{\partial t^2} \tag{20}$$

Figure 8: Other Outputs for FHN Full Order Model

(a) 1D Outputs for FHN Full Order Model



(b) 2D Outputs for FHN Full Order Model

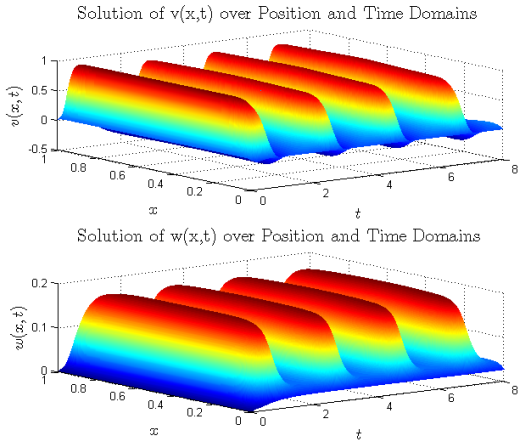
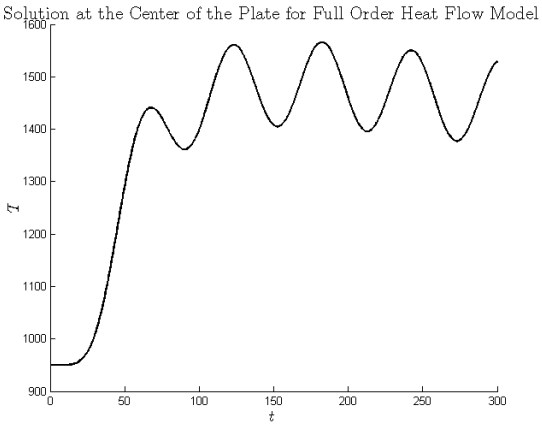


Figure 9: Output of Full Order Nonlinear Heat Transfer Problem



$$\nabla \cdot ((1 + 6K)u^3 p \nabla p) = 12\mu \frac{\partial (pu)}{\partial t} \tag{21}$$

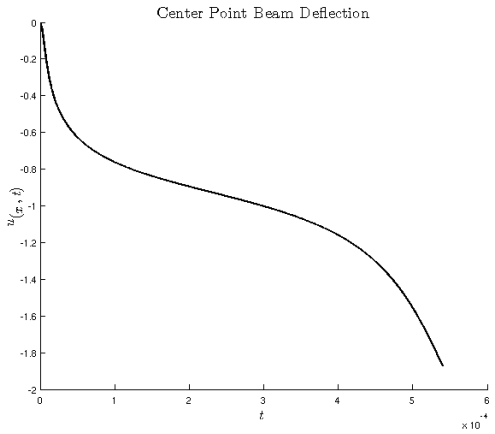
The output of the MEMS problem is the center point deflection of the beam. The output of the full order model is in Figure 10

Highly Nonlinear 2D Steady State Problem

Unlike most of the examples in this paper (with the exception of the Penzl example), this system does not directly correspond to a physical phenomena. This toy problem was choosen for two reasons: 1) a high degree of nonlinearity makes it difficult to solve with a ROM and 2) it is a static system, which necessitates a parametic study. Both of these topics will be discussed more is subsequent sections. The governing (continuous equation) is given in (22) and the discretized form in (23). The full order model consists of 50 grid points in each dimension, so there are 2304 unknowns after application of Dirichlet boundary conditions. For more information, the reader is referred to [4].

$$-\nabla^2 u(\hat{\mathbf{x}}) + \frac{\mu_1}{\mu_2}(e^{\mu_2 u(\hat{\mathbf{x}})} - 1) = 100 \sin(2\pi x) \sin(2\pi y) \tag{22}$$

Figure 10: Output of Full Order MEMS Problem



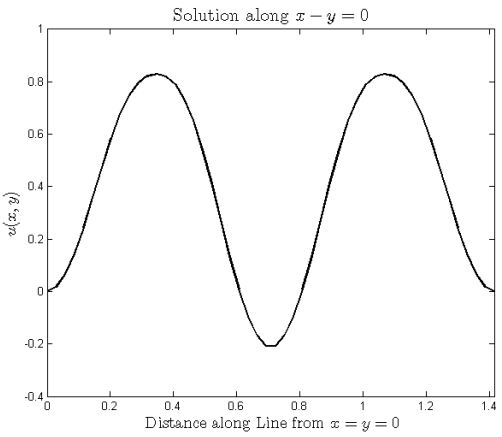
$$\hat{\mathbf{x}} = (x, y) \in \Omega = [0, 1]^2$$
$$\mu = (\mu_1, \mu_2) \in \mathcal{D} = [0.01, 10]^2 \subset \mathbb{R}^2$$
$$u(\hat{\mathbf{x}}) = 0 \quad \forall \hat{\mathbf{x}} \in \partial\Omega$$

$$-\frac{u_{p+1,q} + u_{p-1,q}}{(\Delta x)^2} - \frac{u_{p,q+1} + u_{p,q-1}}{(\Delta y)^2)} + 2u_{p,q}\left[\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2}\right] + \frac{\mu_1}{\mu_2}(e^{\mu_2 u_{p,q}} - 1) - 100\sin(2\pi x_p)\sin(2\pi y_q) \tag{23}$$
$$u_{p,q} = u(x_p, y_q)$$

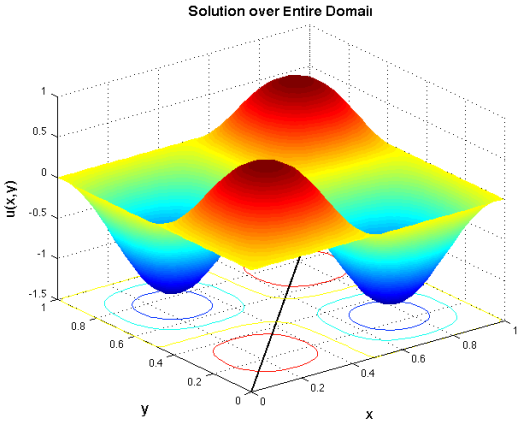
The output for this problem is $u(x, y)$ along the line $x = y$, instead of the solution over the entire domain. In the subsequent sections, it will be much easier to make comparisons on a single plot with a one-dimensional output. For completeness, the one- and two-dimensional output are given in Figures 11a and 11b for the full order model. The line $x = y$ is shown in Figure 11b with a bold, black line.

Figure 11: Highly Nonlinear 2D Steady State FOM

(a) One-Dimensional Output of HNL2dSS Problem



(b) Two-Dimensional Output of HNL2dSS Problem



4 Comparison Results

This section presents the results of the various model reduction methods on the examples introduced in the previous sections. Linear systems will be considered first in Section 4.1 and nonlinear systems in Section 4.2.

4.1 Linear Systems

The results of the model reduction comparison will be presented in a 3x3 subplot for each problem. The first row of subplots will correspond to the L_2 error between the FOM and the ROM as a function of the order of the reduced model. The second row will present the H_2 error vs. order of the ROM. The third row is the offline time required for each model reduction technique. The first two columns of these subplots simply compare the different reduction techniques, while the third column compares different snapshot collections for POD in the time domain. For space efficiency, the marker and color legend for the linear analysis are provided in Figure 12 and Table 1, respectively.

Figure 12: Legend for Linear Results

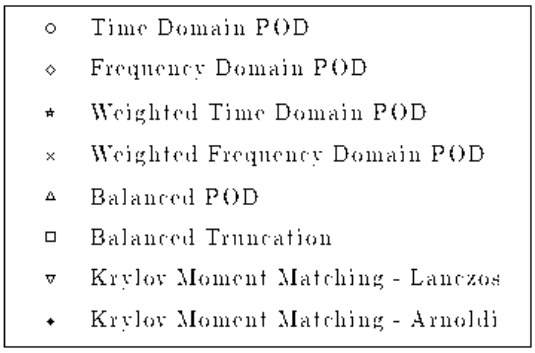


Table 1: Linear Results Color Legend

Color	**Meaning**
red	unstable system
black	stable system
blue	$u(t_i)$ Snapshots
green	$u(t_i) - u(t_0)$ Snapshots
magenta	$u(t_i) - u(t_{i-1})$ Snapshots

Multiple Mass-Spring-Damper

In addition to the plots discussed above, the mass-spring-damper problem includes plots of H_∞ -Error vs. order of the ROM.

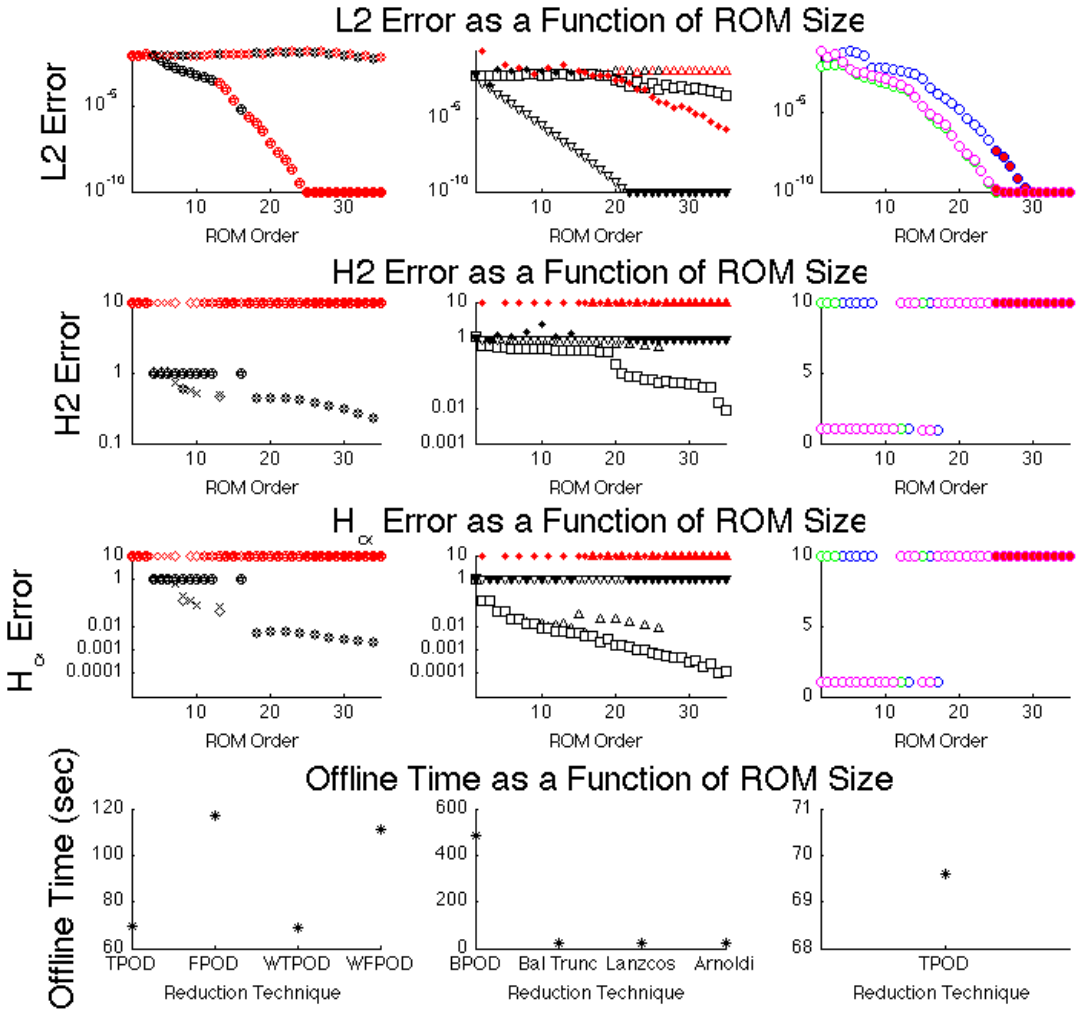
Figure 13 presents the results of the anaylsis of the mass-spring-damper problem. The POD techniques in the time domain have a large number of unstable points, but an unstable ROM does not necessarily correspond to large error. The Balanced POD (BPOD) and Balanced Truncation methods remained stable throughout all degrees of model reduction, which is a theoretical expectation. Furthermore, the POD methods in the time domain tend to have the lowestL_2 error across all ROM orders considered and the POD methods in the frequency domain have the lowest H_2 and H_∞ errors. Notice the large number of unstable systems that were generated by reducing the order of the mass-spring-damper system.

Due to the nature of linear systems, the online computational time for all MOR techniques are roughly equal, so the quantification of speedup will involve the offline time. The POD methods in the frequency

domain and BPOD were the most computationally expensive MOR techniques, while the Krylov methods are the least expensive.

The snapshot collection that references the initial condition minimized the L_2 error for most model reduction orders. Notice that the H_2 and H_∞ plots did not yield any useful information due to the large number of unstable points (H_2 and H_∞ errors of an unstable system are infinite).

Figure 13: Results of MOR Comparision for Mass-Spring-Damper System



Penzl Example

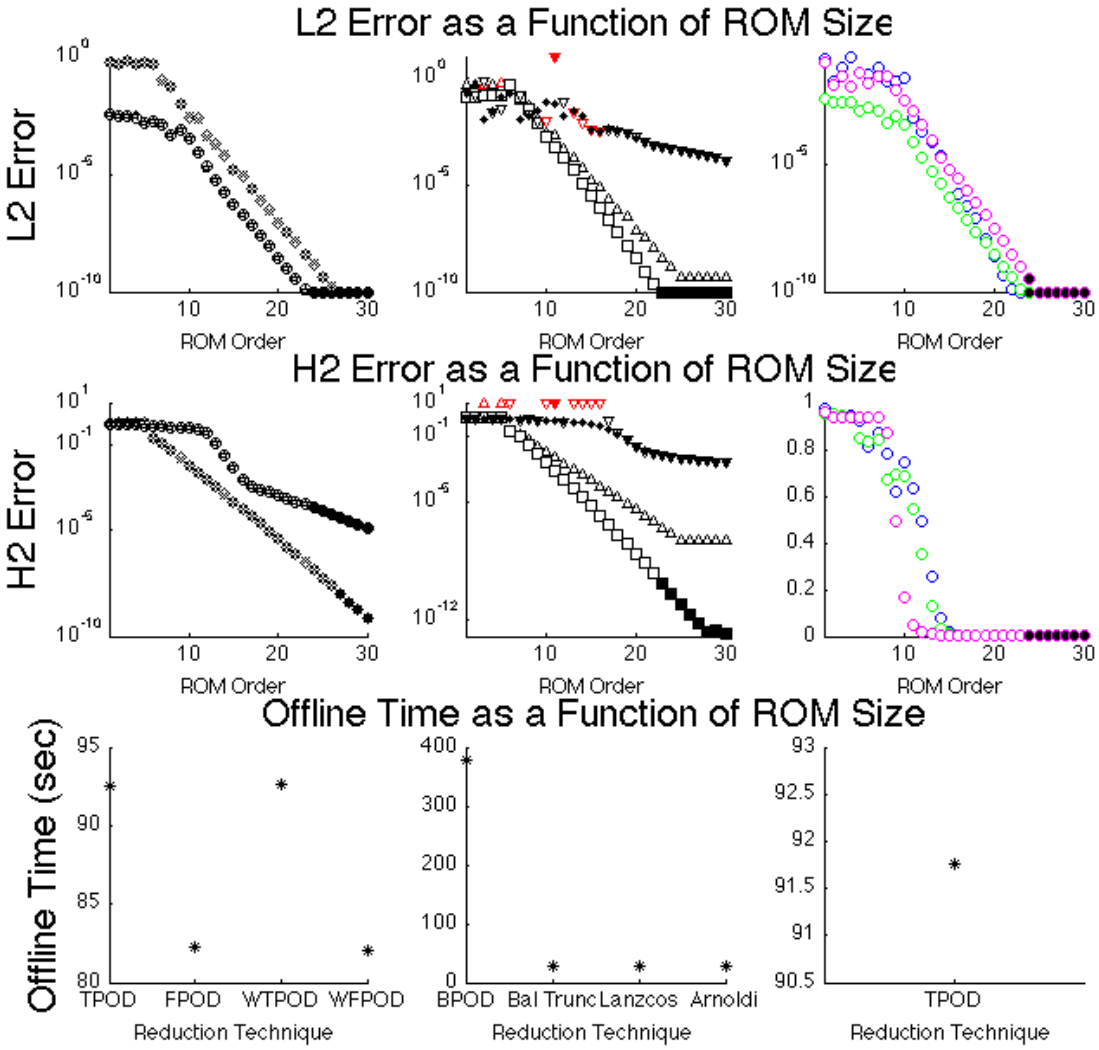
Similar to the mass-spring-damper system, the time POD methods have lower L_2 errors and higher H_2 errors than the frequency POD methods. For the ROM order spectrum investigated, Balanced Truncation yielded the most accurate ROMs since it generated the smallest L_2 and H_2 errors.

Unlike the mass-spring-damper system, the time domain POD methods were more expensive than the frequency POD methods. However, similar to the mass-spring-damper system, the Krylov methods were the most computationally inexpensive.

The snapshot comparison for this system was inconclusive because the optimal snapshot collection

was dependent on the order of the ROM. However, all snapshot collections produced ROMs with very small L_2 errors.

Figure 14: Results of MOR Comparision for Penzl Example



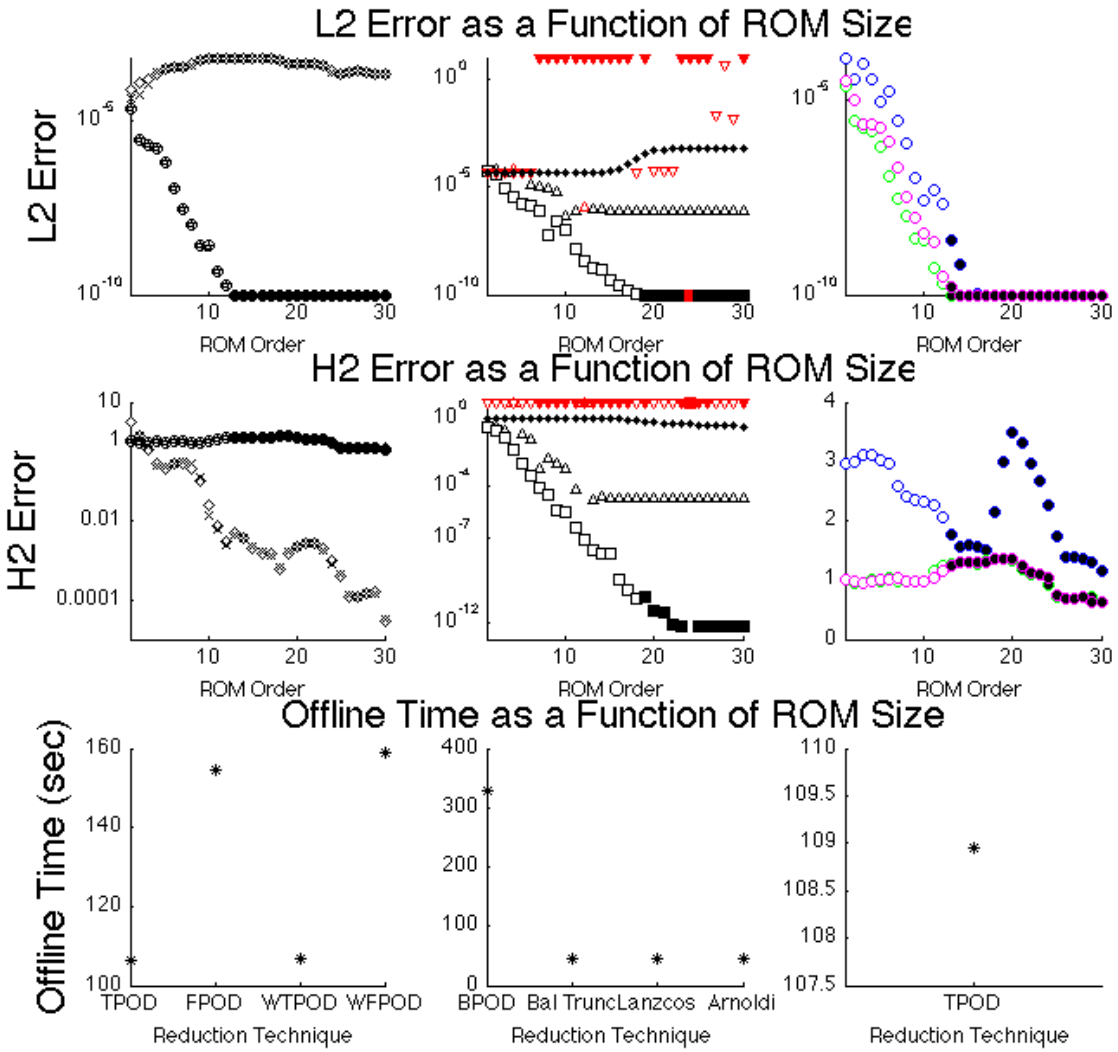
Linear Heat Flow

As was seen in the other linear systems, the time domain POD methods have lower L_2 errors and larger H_2 errors than the frequency domain POD methods. Similar to the Penzl Example, Balanced Truncation is the optimal MOR technique for the heat flow problem since it generates ROMs with the smallest L_2 and H_2 errors. The heat flow problem has more unstable ROM systems than the Penzl Example, but fewer than the mass-spring-damper system. Most of the unstable systems were generated using the Lanczos algorithm for Krylov moment-matching.

The snapshots that reference the initial condition are the optimal snapshot collection for this problem.

The trends in computational complexity are identical to those seen in the mass-spring-damper system.

Figure 15: Results of MOR Comparision for Linear Heat Flow System



4.2 Nonlinear Systems

This section presents the results from the model reduction comparison on nonlinear problems. As discussed previously, this section will compare the accuracy and performance of Galerkin POD, Petrov-Galerkin POD, Gappy POD, and TPWL on six nonlinear problems. However, each of these reduction techniques have variable parameters that will affect the accuracy and performance of the reduced model. For the remainder of this paper, it will be assumed that the training input and the online input are the same unless otherwise specified.

For Galerkin and Petrov-Galerkin POD, there are different snapshot collections available that will affect the quality of the reduced model. The three common snapshot collections are: 1) non-referenced, current state vector ($u(t)$), 2) current state vector referenced using the initial condition ($u(t_i) - u(t_0)$), and 3) current state vector referenced using the previous state vector ($u(t_i) - u(t_{i-1}$). The first step in the comparison of model reduction techniques will be to determine the best snapshot collection for each

problem.

The second step in the evaluation of model reduction techniques is to determine the optimal Gappy parameters for each problem. There are three adjustable parameters for Gappy POD that will have dramatic affects on the quality of the reduced model. These parameters are n_R, n_J, and n_I; however, the discussion of these parameters is beyond the scope of this paper. The notation was taken directly from [3], which is where an in-depth theoretical discussion of the Gappy method is provided.

The next step in the comparison of MOR techniques is to determine the optimal TPWL linearization point selection algorithm for each problem. The quality of the reduced model that results from TPWL is extremely dependent on the linearization points selected. Three algorithms for selecting linearization points were employed. The first two algorithms use distance between state vectors [6] and a residual-based distance [7], respectively, to select linearization points. The final algorithm uses the curvature of the solution trajectory to determine linearization points. This algorithm was developed by the author using the heuristic that it is advantageous to select more points in areas that are the most nonlinear. The algorithm also attempts to separate the linearization points enough so the entire trajectory space is covered. Equation 24 was used to calculate the curvature at each point along the trajectory of the solution.

$$\rho = \frac{\mid \gamma' \mid^3}{\sqrt{\mid \gamma' \mid^2 \mid \gamma'' \mid^2 - (\gamma' \cdot \gamma'')^2}} \tag{24}$$

The final step of the inter-MOR analysis procedure is to compare Galerkin POD, Petrov-Galerkin POD, Gappy POD, and TPWL using only the optimal parameters determined from the previous steps.

The final comparison made is an intra-MOR robustness analysis. The robustness study will use the optimal parameters determined from the previous analyses and compare each ROM trained with the "training" input to the corresponding ROM trained with the "online" input.

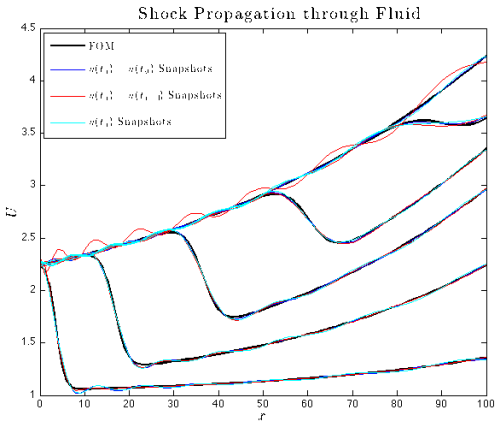
Burger's Equation

A ROM order of 15 was selected for the Burger's Equation model, which is a fairly modest reduction given that the order of the FOM was 101. Given the degree of reduction, one would expect all of the model reduction techniques to perform well on this problem, which is not necessarily the case. The scaled-up FOM and ROM for this problem have 4000 and 40 degrees of freedom (dofs), respectively.

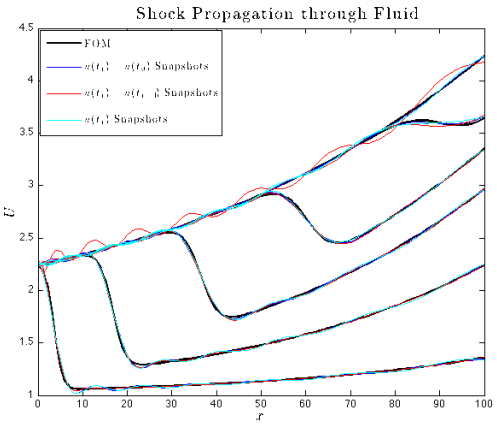
A snapshot collection comparison was made using Galerkin and Petrov-Galerkin POD projections. It is evident from Figure 16 and Tables 2 and 3 (at the end of the document) that the unreferenced snapshot collection and the snapshots referencing the initial vector outperform the snapshots referencing the previous time step.

Figure 16: Snapshot Collection Comparison for Burger's Equatin

(a) Galerkin POD



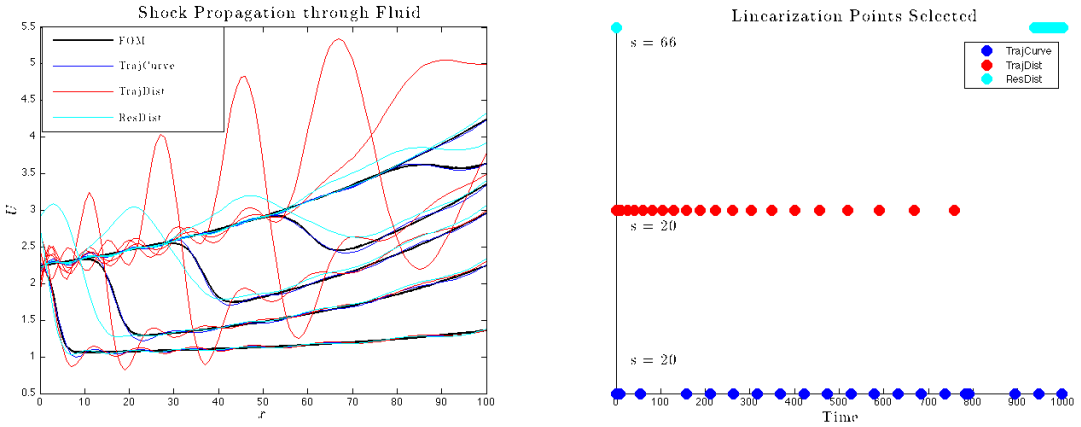
(b) Petrov-Galerkin POD



The comparison of the TPWL linearization point selection algorithms are provided in Figure 17 along

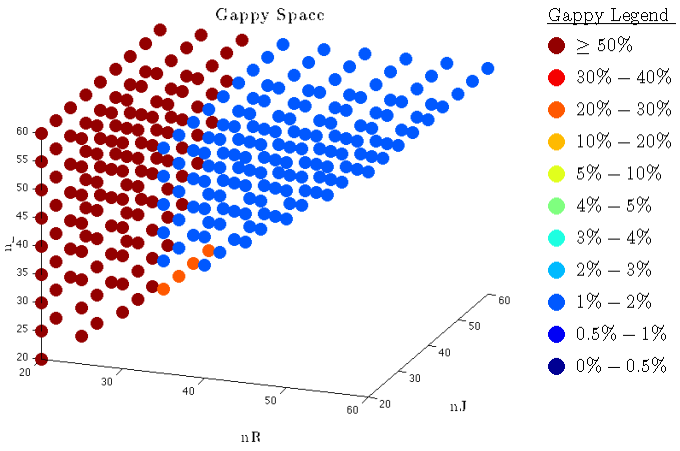
with the linearization points selected by each algorithm. The trajectory distance and residual distance algorithms are very dependent on a tolerance parameter that determines the minimum distance between snapshots or residuals, respectively. Furthermore, residual distance algorithm is designed such that the number of linearization points is an output rather than an input. This is the reason that residual distance algorithm usually doesn't have the same number of linearization points as the other two linearization point selection algorithms. Figure 17 shows that trajectory and residual distance algorithms fail to select appropriate linearization points and the resulting solution is not meaningful, while the trajectory curvature algorithm matches the full order solution very closely.

Figure 17: TPWL Linearization Point Selection Algorithm Comparison for Burger's Equation



The next step in the analysis is to determine the optimal Gappy parameters for the given problem. This will be done by utilizing what will be called "Gappy Space" ($\mathcal{G}$), where $\mathcal{G} \subset \mathbb{Z}^3$. $\mathcal{G}$ does not span all of $\mathbb{Z}^3$ due to the following parameter constraints described in [3]: $\min\{n_R, n_J\} \geq n_Y$ and $n_I \geq \max\{n_R, n_J\}$. From Figure 18, it can be concluded that the n_R is the most sensitive Gappy parameter. Therefore, a logical method to select Gappy parameters would be to first determine the n_R that corresponds to the maximum acceptable L_2 error, then select n_J and n_I that are as small as possible (to minimize computational cost) while still remaining in $\mathcal{G}$.

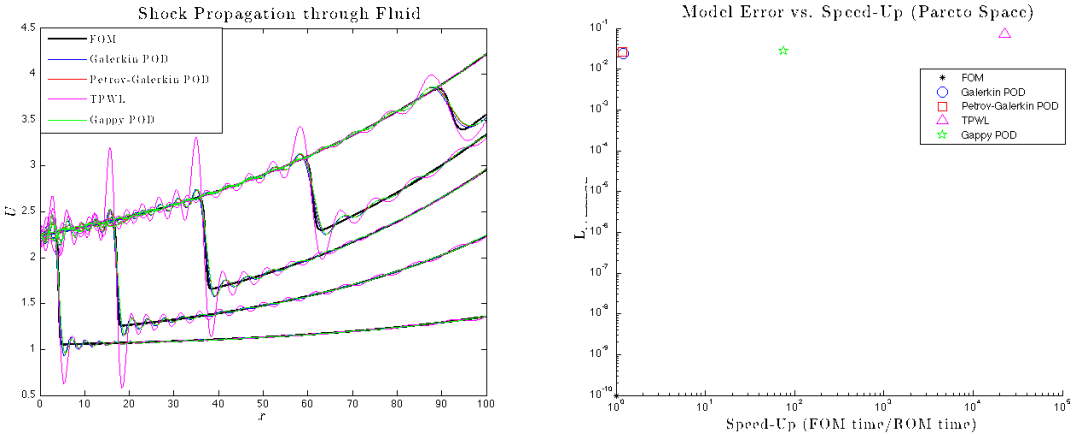
Figure 18: L_2 Error in "Gappy Space" for Burger's Equation



The next step of the analysis procedure is to use the best results from the preceeding analyses

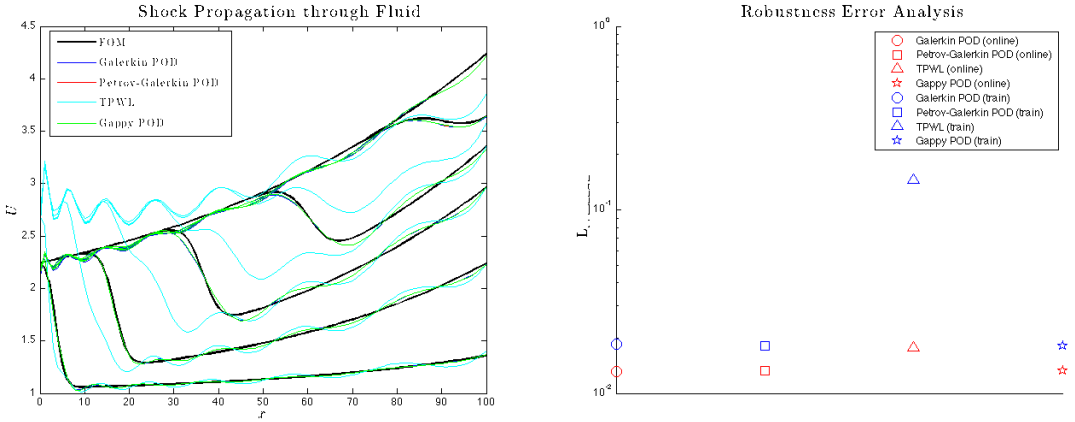
and compare all reduction techniques. Since this is a straight comparision of MOR techniques, not a robustness study, the "training" and "online" inputs are the same. For this component of the MOR analysis, the model is scaled up to highlight the speed-ups that result from each MOR technique. This comparison was made using snapshots that reference the initial vector for all of the POD techniques, trajectory curvature algorithm for TPWL, and the Gappy parameters $(n_R, n_J, n_i) = (50, 20, 50)$. Since the model was scaled up for this part of the analysis, it follows that the Gappy parameters will need to be scaled up as well, $(n_R, n_J, n_i) = (130, 40, 130)$.

Figure 19: Final Model Reduction Comparison for Burger's Equation'



The final MOR analysis performed on this problem is a robustness study. This part of the analysis uses the original "scaled-down" model, where the "training" input differs from the "online" input. For this problem, the "training" input is the step function $(H(t))$ with an amplitude of 1 and the "online" input is the step function with an amplitude of 5.

Figure 20: Model Reduction Robustness Analysis for Burger's Equation'



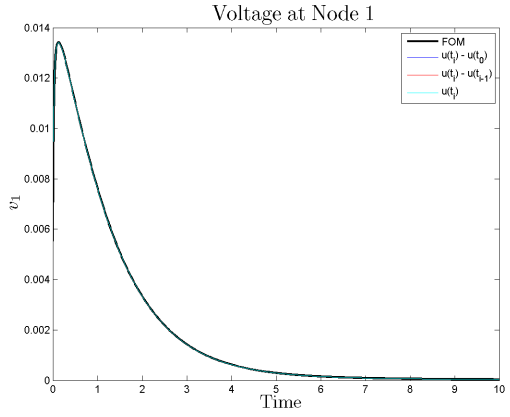
Nonlinear Transmission Line

Unlike the other problems in this paper, the nonlinear transmission line was analyzed for two different input currents, which are provided in Section 3. The FOM and ROMs have 100 and 10 dofs, respectively, for a majority of the analysis. The scaled-up model used in the final MOR comparison has 4000 and 30 dofs, respectively.

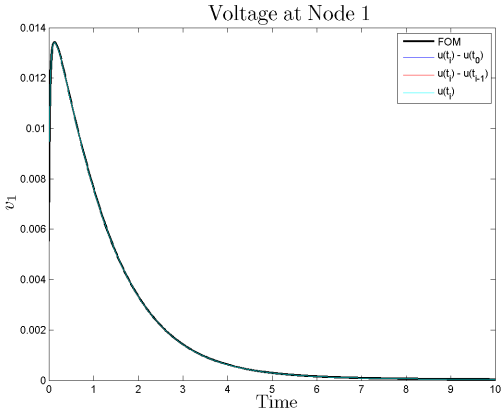
For this problem, all three snapshot collections perform quite well, but the snapshots that reference the previous state vector has a slight advantage. Recall that snapshots that reference the previous state vector performed the worst on Burger's Equation. Also, recall that the initial condition is the zero vector for this problem, so the nonreferenced snapshot collection and the collection referencing the initial condition are identical.

Figure 21: Snapshot Collection Comparison for the Nonlinear Transmission Line

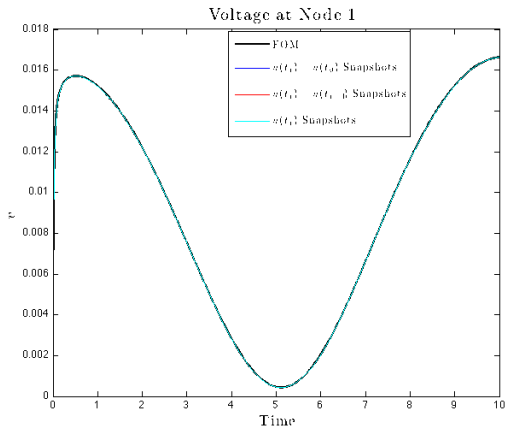
(a) Galerkin POD, $i(t) = e^{-t}$



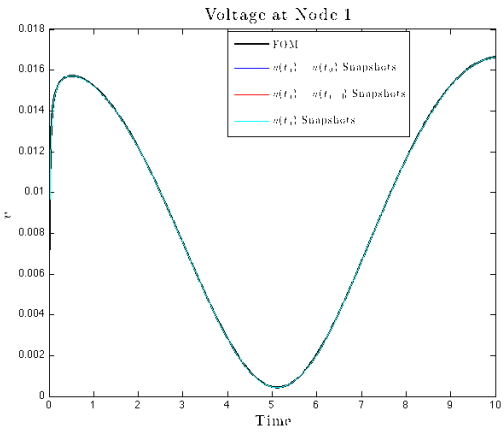
(b) Petrov-Galerkin POD, $i(t) = e^{-t}$



(c) Galerkin POD, $i(t) = \frac{1}{2}\left(\cos\frac{2\pi t}{10} + 1\right)$



(d) Petrov-Galerkin POD, $i(t) = \frac{1}{2}\left(\cos\frac{2\pi t}{10} + 1\right)$



The TPWL comparison for this problem yielded vastly different results than were seen in the previous problem. For both the cosinusoidal and exponential inputs, the residual distance algorithm selects the linearization points that minimizes the L_2 error. For the exponential input, the trajectory distance and curvature algorithms result in ROMs that initially follow the FOM solution very closely, but veer of after approximately 20% of the time steps. For the cosinusoidal input, the trajectory distance algorithm generates a ROM that mimics the shape of the FOM, but the magnitudes do not match. The ROM generated by the trajectory curvature method is quite inaccurate for a majority of the time domain.

For this problem, the trajectory curvature algorithm selects points that span nearly the whole time domain, while the other two algorithms select all of their points at the beginning of the domain. Notice that the residual distance algorithm outperform the other two, despite the fact that it uses less linearization points. The residual distance algorithm selects only two points and the other algorithms select five points.

Figure 24 presents the results from the Gappy analysis for the nonlinear transmission line problem.

Figure 22: TPWL Linearization Point Selection Algorithm Comparison for Transmission Line with $i(t) = e^{-t}$

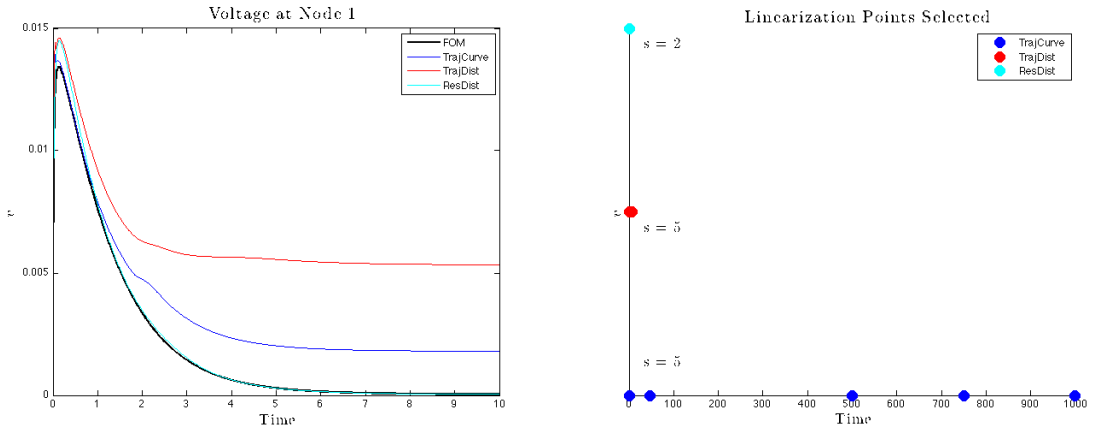
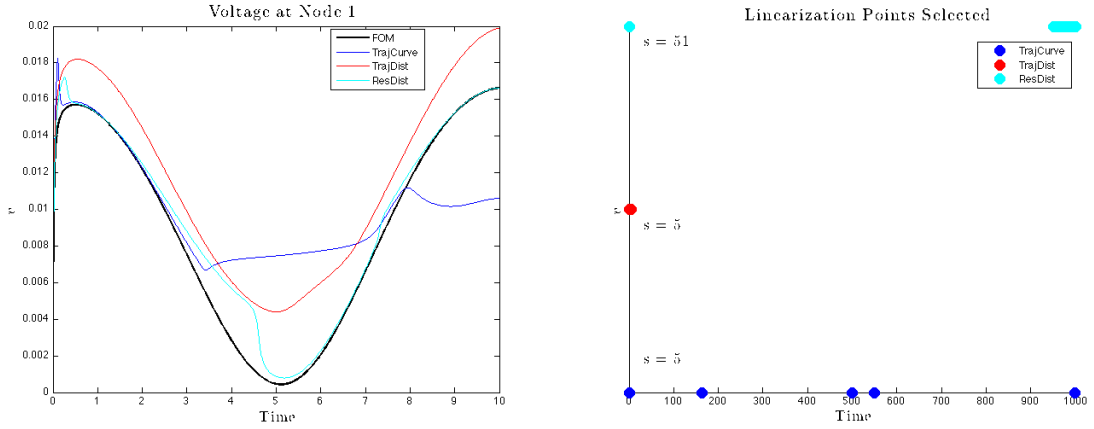


Figure 23: TPWL Linearization Point Selection Algorithm Comparison for Transmission Line with $i(t) = \frac{1}{2}\left(cos(2\pi t/10) + 1\right)$



Similar Gappy POD parameter trends were seen in this problem as in Burger's Equation. Namely, there is a strong dependence on n_R and a weak dependence on n_J and n_I. The optimal Gappy parameters for the exponential and cosinusoidal inputs are $(n_R, n_J, n_I) = (40,\ 10,\ 50)$. After scaling the model up for use in the next section, these parameters become $(n_R, n_J, n_I) = (50,\ 30,\ 60)$.

The comparison of all MOR techniques for this problem are presented in Figures 25 and 26. As with Burger's Equation, TPWL results in the most speedup and largest error. Gappy POD has the second largest error and speedup of the MOR techniques considered. Galerkin and Petrov-Galerkin POD methods are the most accurate MOR techniques; however, they do not significantly reduce computational cost.

Notice that in Figures 25 and 26 the Galerkin POD MOR method has a smaller L_2 error than the FOM. This is purely an artifact of the logarithmic ordinate. It was previously mentioned that the FOM is considered to be the "exact" solution and thus has zero error. Since a logarithmic axis cannot represent zero, a small number $\epsilon = 10^{-10}$ was assigned. For this problem, it turned out that the L_2 error of Galerkin POD model reduction is less than ϵ.

Figure 27 presents the robustness analysis for both inputs of the nonlinear transmission line problem.

Figure 24: L_2 Error in "Gappy Space" for the Transmission Line problem

(a) $i(t) = e^{-t}$ (b) $i(t) = \frac{1}{2}\left(\cos\frac{2\pi t}{10} + 1\right)$

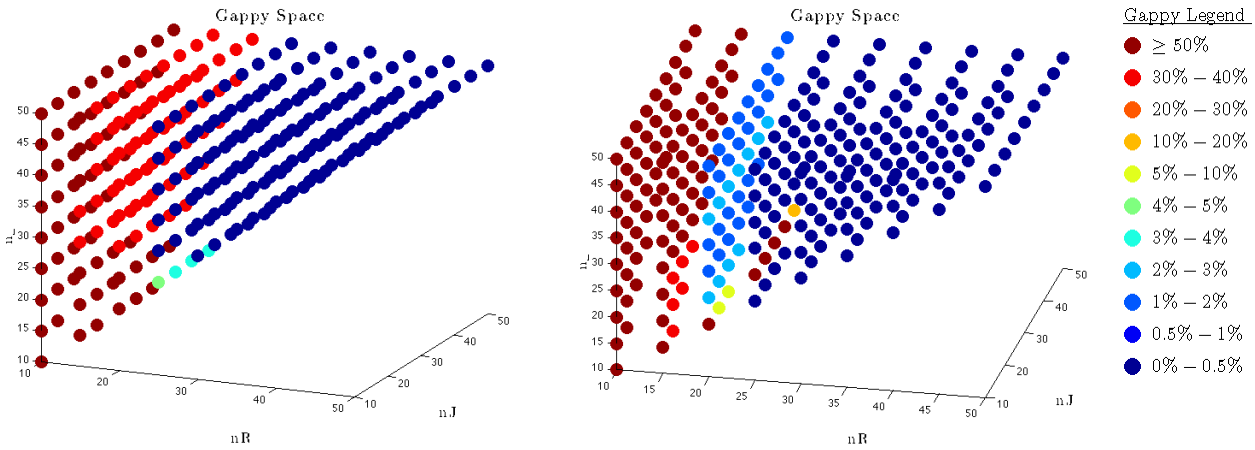
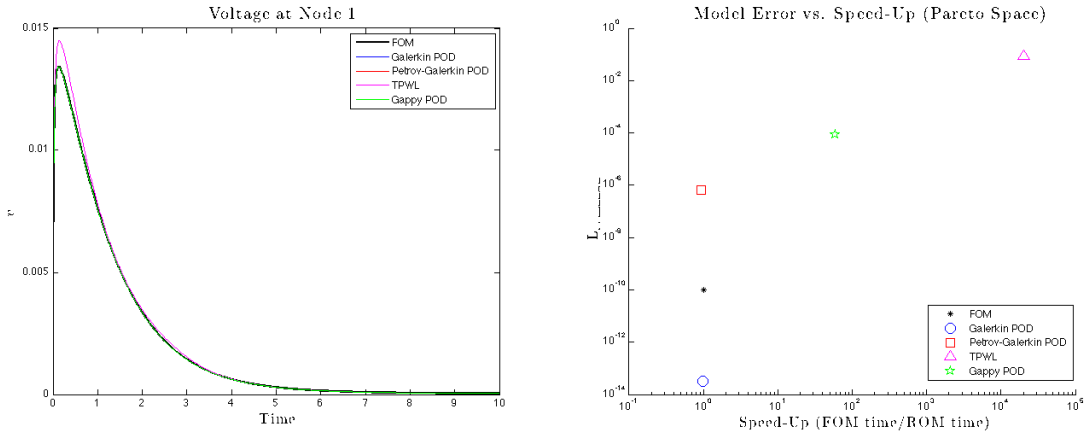


Figure 25: Final Model Reduction Comparison for the Transmission Line problem with $i(t) = e^{-t}$



All MOR techniques exhibit a great degree of robustness for this problem because the L_2 error is roughly the same regardless of whether or not the "training" and "online" inputs are equal. The "training" input is a step function ($H(t)$) of ampltitude 1 for the the exponential and cosinusoidal "online" inputs. Notice that of all the MOR techniques, TPWL is the least robust.

Figure 26: Final Model Reduction Comparison for the Transmission Line problem with $i(t) = \frac{1}{2}\left(\cos\frac{2\pi t}{10} + 1\right)$

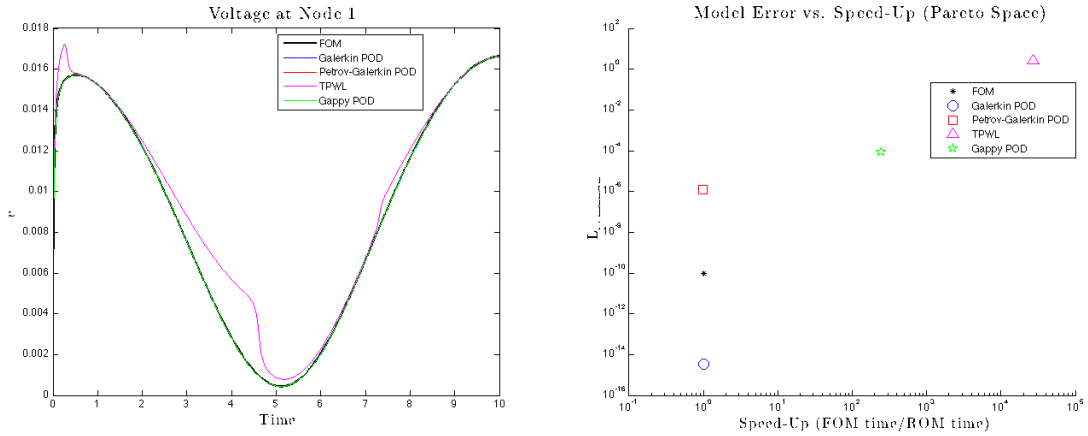
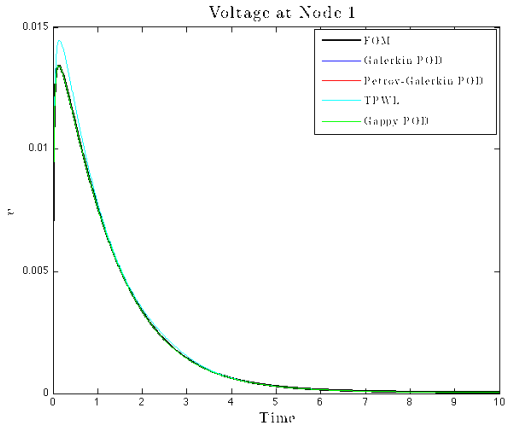
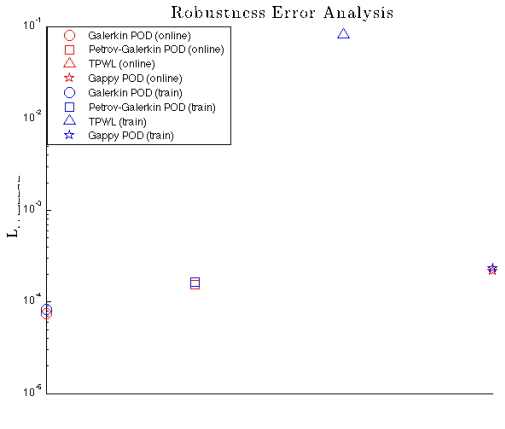


Figure 27: Model Reduction Robustness Analysis for the Transmission Line problem

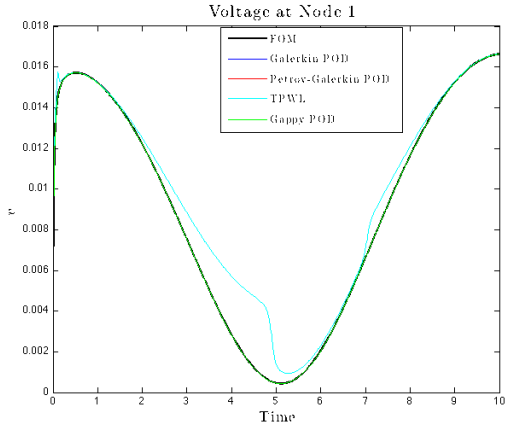
(a) $i(t) = e^{-t}$



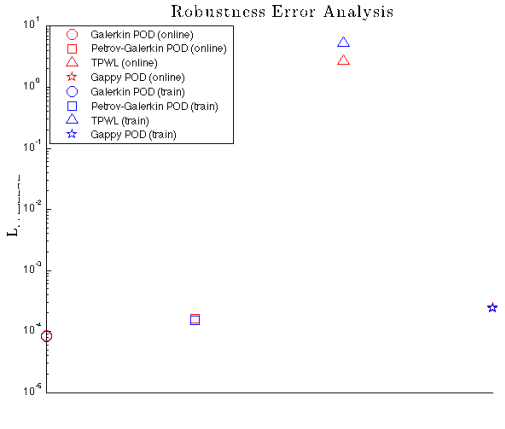
(b) $i(t) = e^{-t}$



(c) $i(t) = \frac{1}{2}\left(\cos\frac{2\pi t}{10} + 1\right)$



(d) $i(t) = \frac{1}{2}\left(\cos\frac{2\pi t}{10} + 1\right)$



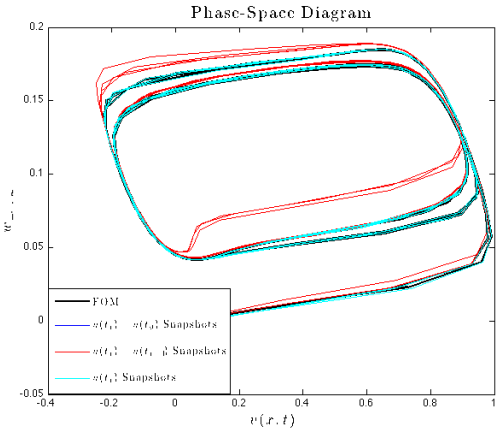
FitzHugh Nagumo System

Unlike the other systems in this document, FitzHugh Nagumo equations have multiple outputs of interest. The first output is the phase-space diagram of the voltage and voltage recovery of the system. The other outputs are the voltage and voltage recovery at specific times and positions that are indicated in the following figures. For most of the analysis, the FOM and ROM have 1024 and 20 dofs, respectively. The final, scaled-up MOR comparison uses a FOM and ROM with 2048 and 20 dofs, respectively.

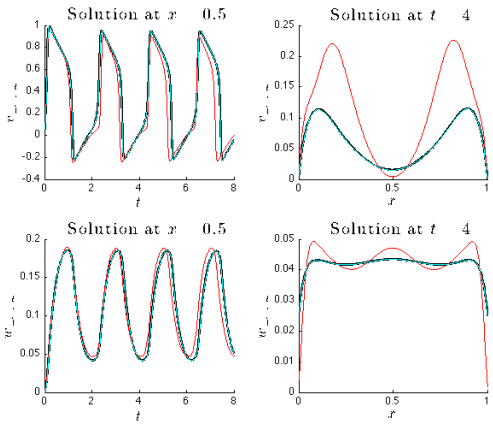
Figure 28 presents the results of the snapshot comparison for Galerkin and Petrov-Galerkin POD MOR techniques. Similar to the results seen in Burger's Equation, the snapshot that references the initial condition is the best snapshot collection with regard to L_2 error, while the snapshot that references the previous state vector is the worst.

Figure 28: Snapshot Collection Comparison for FitzHughh-Nagumo Equations

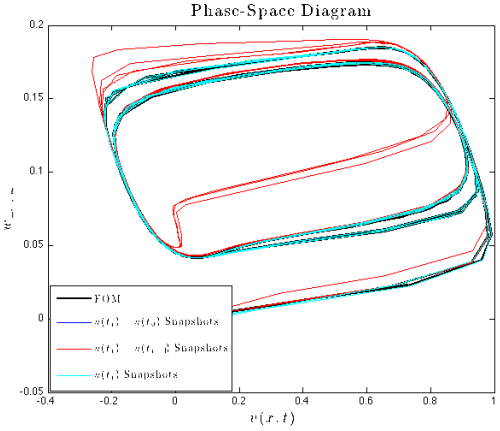
(a) Galerkin POD



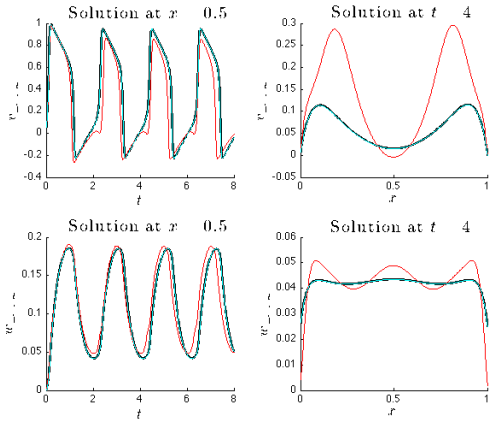
(b) Galerkin POD



(c) Petrov-Galerkin POD



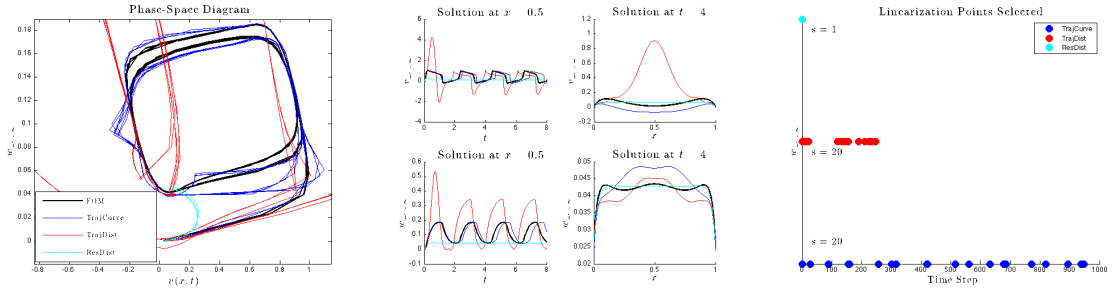
(d) Petrov-Galerkin POD



The TPWL linearization point selection algorithm comparison is presented in Figure 29. For this problem, none of the algorithms select points that generate an accurate ROM. The residual distance algorithm seems to be the superior algorithm with regard to the phase-space output, while the trajectory curvature algorithm outperforms the others in the time and position domains. The trajectory distance algorithm fails with regard to all outputs. The trajectory curvature algorithm is used for the final "scaled-up" comparison.

Figure 30 shows that the Gappy parameter trends for the FitzHugh-Nagumo eqautions are the same

Figure 29: TPWL Linearization Point Selection Algorithm Comparison for the FitzHugh-Nagumo Equations



as those from the previous problems. The optimal Gappy parameters for the "scaled-down" problem are $(n_R, n_J, n_I) = (90, 50, 110)$. The Gappy parameters used for the scaled-up model are $(n_R, n_J, n_I) = (90, 50, 110)$.

Figure 30: L_2 Error in "Gappy Space" for the FitzHugh-Nagumo problem

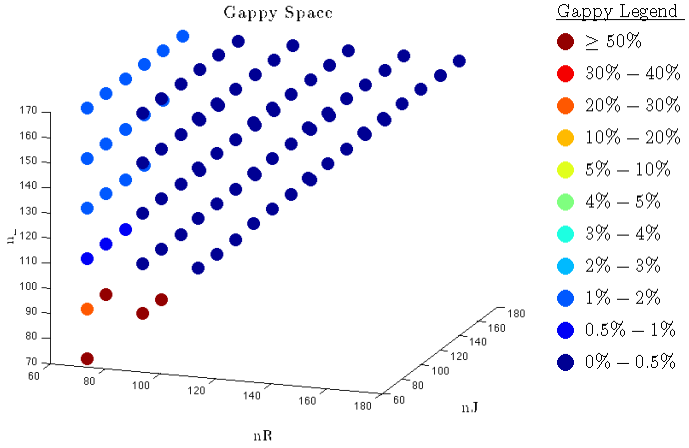


Figure 31 shows that the MOR techniques fall into the same heirarchy seen in the previous two problems. TPWL and Gappy POD are the fastest, but incur the greatest error. For this problem, the large speedup of the TPWL method is accompanied by an L_2 error that is unacceptably large ($\sim 10^1$), while Gappy POD is not as computationally efficient, but has a managable error ($< 10^{-2}$). Galerkin and Petrov-Galerkin POD have roughly the same speedup and L_2 error, with Petrov-Galerkin having a slight advantage.

Figure 32 presents the results of the robustness analysis for the FitzHugh Nagumo Equations. All of the methods exhibit a large degree of robustness with TPWL being the least robust. The "training" input is $i(t) = 10/(t+1)$ and the "online" input is $i(t) = 50000t^3e^{-15t}$.

Figure 31: Final Model Reduction Comparison for the FitzHugh-Nagumo Problem

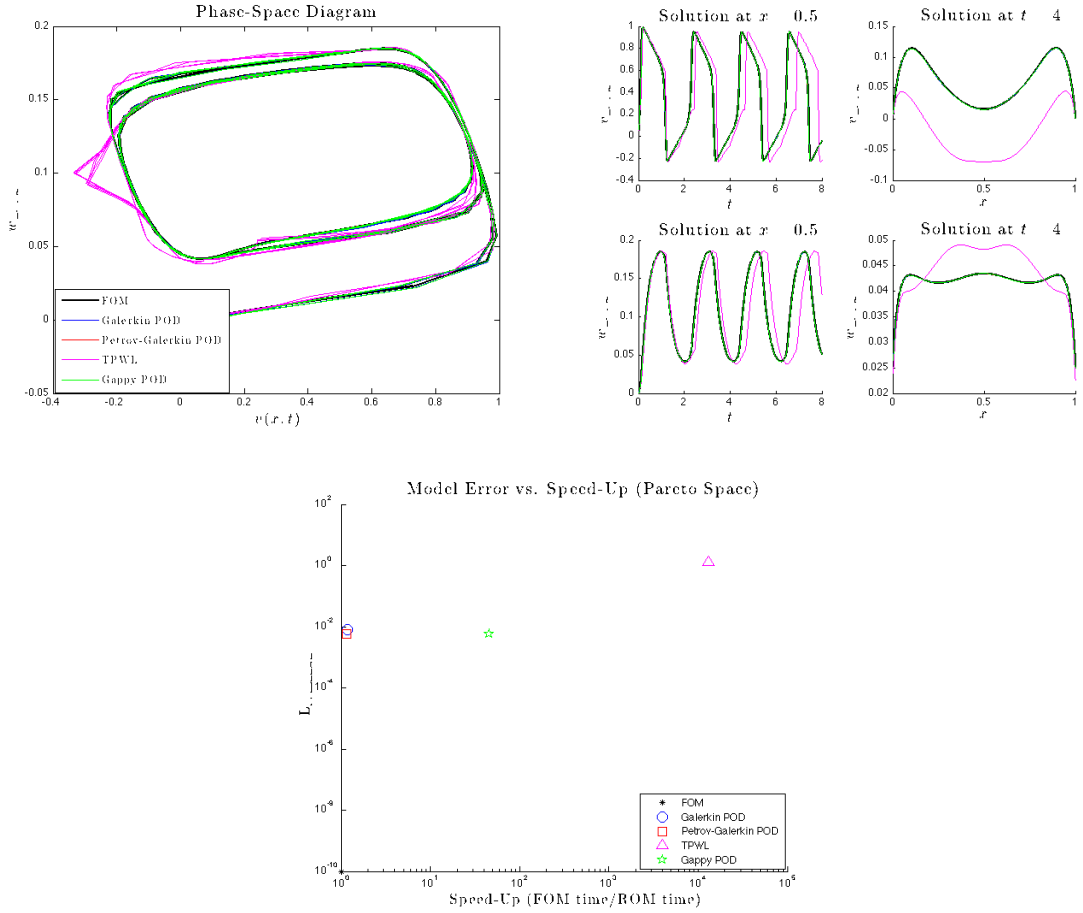
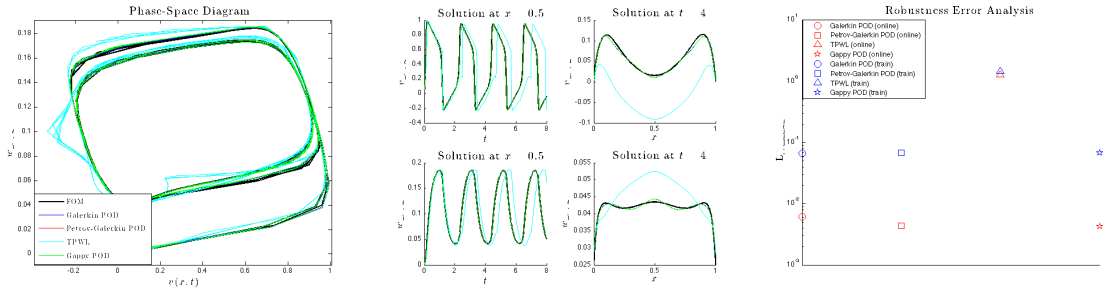


Figure 32: Model Reduction Robustness Analysis for the FitzHugh-Nagumo Equations'



MEMS Switch

The MEMS switch problem turned out to be the most difficult and computationally expensive problem of those considered. The FOM and ROM have 880 and 10 dofs, repsectively. Due to the computational expense of this problem, the model was not scaled up for the final MOR comparison.

Similar to the previous problems, the snapshot collection that references the initial condition generates

the most accurate ROM for the Galerkin and Petrov-Galerkin POD techniques. The snapshot collection that references the previous condition also does quite well for both POD techniques. The unreferenced snapshot collection completely fails on the Galerkin POD method only. For the Petrov-Galerkin method, the ROM solution veers off the FOM solution toward the end of the time interval.

Figure 33: Snapshot Collection Comparison for MEMS Switch

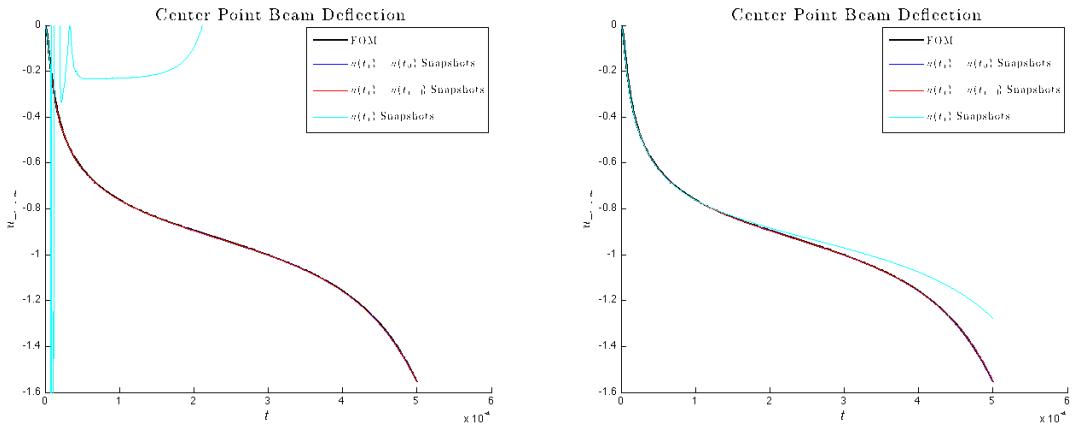


Figure 34 presents the result of the TPWL comparison for the MEMS switch. Due to the high degree of nonlinearity of this problem, it was difficult to approximate with a series of linear systems. To generate an acceptable ROM, it was necessary to use a ROM order of 500. Figure 34 shows that the trajectory curvature and distance algorithms generate the best approximations of the FOM, with the trajectory curvature algorithm having the advantage. The residual distance algorithm fails on the MEMS problem, despite the large order of the ROM.

Figure 34: TPWL Linearization Point Selection Algorithm Comparison for MEMS Switch

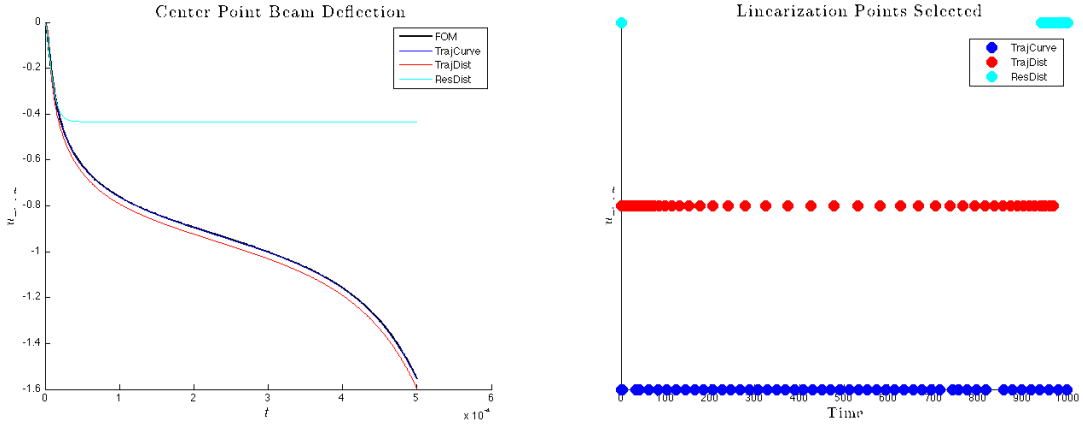


Figure 35 presents the results of the Gappy comparison for the MEMS switch. The trends are similar to those seen in the previous problems. The optimal Gappy parameters are $(n_R, n_J, n_I) = (32, 22, 42)$.

Figure 36 presents the results the final MOR comparion for the MEMS switch. The same heirarchy described in the above problems govern this problem.

The robustness analysis for the MEMS switch is summarized in Figure 37. The "training" input is a step function of amplitude 49 and the "online" input is a step function of amplitude 81. Due to the large

Figure 35: L_2 Error in "Gappy Space" for the MEMS Switch

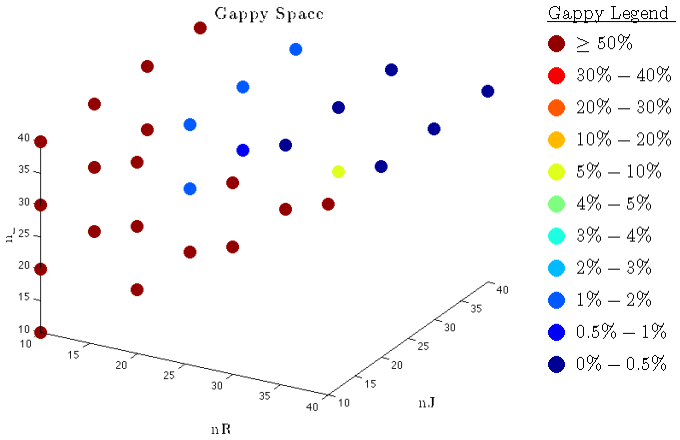
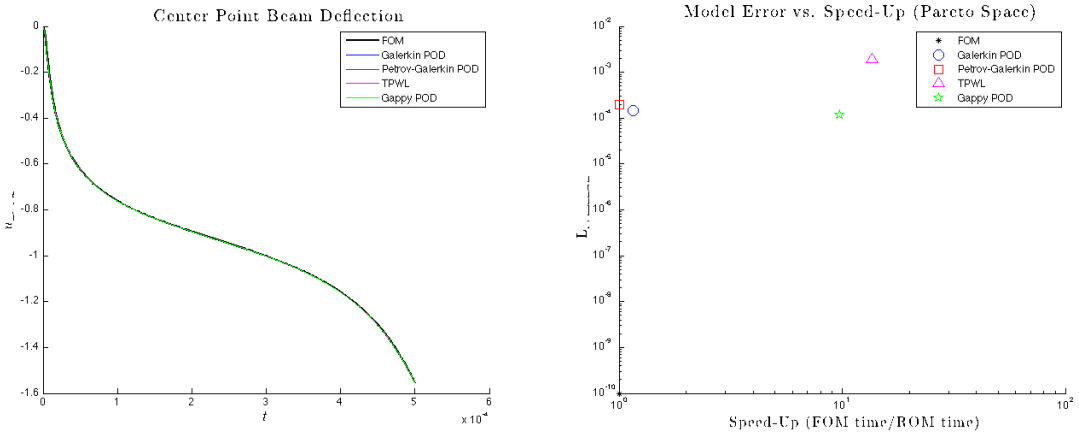
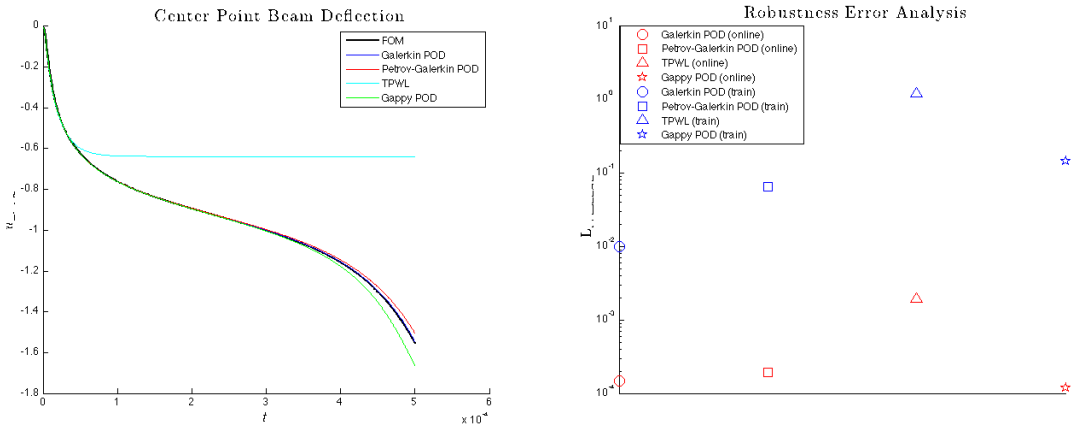


Figure 36: Final Model Reduction Comparison for the MEMS Switch



differences between the amplitudes of the "training" and "online" inputs, there are discrepancies in L_2 error between the ROMs trained with the "training" input and those trained with the "online" input. Overall, these discrepancies are small, so the MOR techniques are fairly robust on the MEMS problem.

Figure 37: Model Reduction Robustness Analysis for MEMS Switch



Highly Nonlinear 2D Steady State

The full order HNL2dSS problem was generated using a 50x50 grid of nodes, which translates to 2304 dofs after the adjustment for Dirichlet boundary conditions. The ROMs compared in the following analysis have 100 dofs. Since this is a static problem, it isn't meaningful to discuss snapshot collections, TPWL algorithms, or an input robustness analysis. Hence, the only results presented are the Gappy space comparisons and the final comparison between all three POD methods (Galerkin, Petrov-Galerkin, and Gappy).

Figure 38 presents the Gappy parameter comparison for the HNL2dSS problem. Unlike the other problems, the accuracy of the Gappy POD reduced order model is moderately dependent on n_J in addition to n_R. The optimal Gappy parameters for this problem are $(n_R, n_J, n_I) = (80, 60, 80)$.

Figure 38: L_2 Error in "Gappy Space" for the HNL2dSS Problem

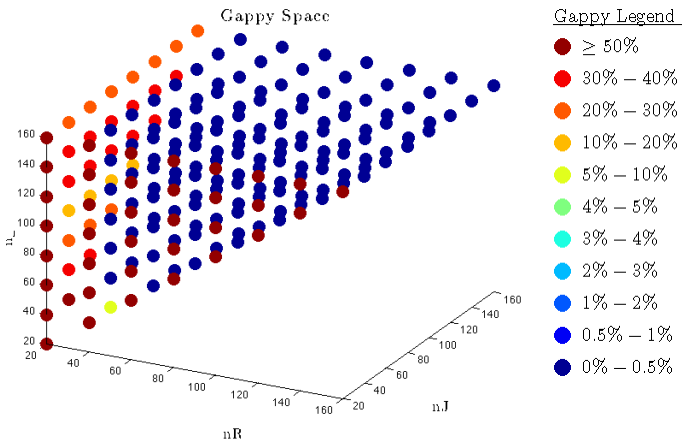
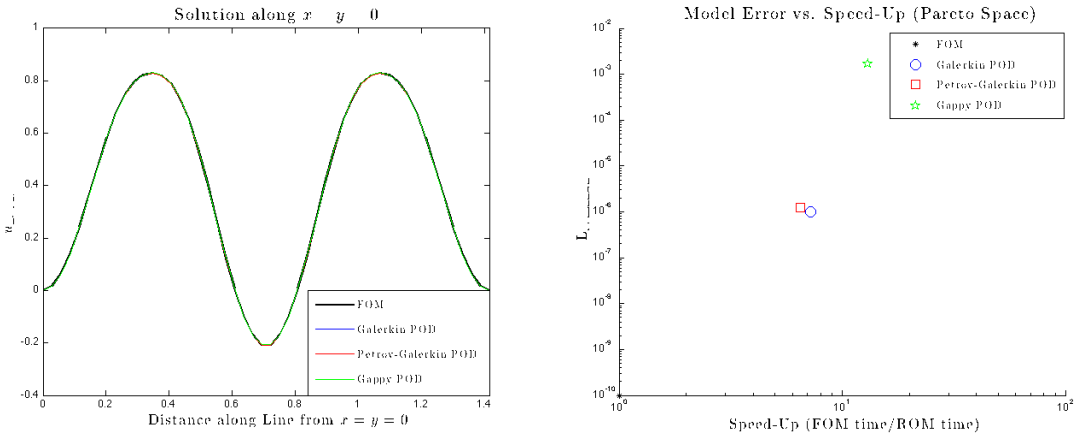


Figure 39 presents the final MOR comparison for the HNL2dSS problem.

Nonlinear 2D Heat Flow

Unlike the other nonlinear problems analyzed in this study, the nonlinear heat transfer system was solved using finite elements, not finite differences. This requires nontrivial generalizations in the testbed code, which are currently in the implementation phase. Therefore, this version of the document will only include the FOM for nonlinear heat flow system.

Figure 39: Final Model Reduction Comparison for the HNL2dSS Problem



4.3 Tabular Summary

Table 2: Summary of Snapshot Collection Comparison Results for Galerkin POD

L_2 *Error*	$u(t_i)$ Snapshots	$u(t_i) - u(t_0)$ Snapshots	$u(t_i) - u(t_{i-1})$ Snapshots
Burger's Eqn	1.47%	1.32%	1.89%
TransLine - cos†	8.60×10^{-3}%	8.60×10^{-3}%	5.04×10^{-3}%
TransLine - exp†	7.48×10^{-3}%	7.48×10^{-3}%	4.49×10^{-2}%
FHN†	0.608%	0.608%	∼100%
HNL2dSS	N/A	N/A	N/A
Heat Flow	-	-	-
MEMS	∼180%	0.0438%	0.213%

Table 3: Summary of Snapshot Collection Comparison Results for Petrov-Galerkin POD

L_2 *Error*	$u(t_i)$ Snapshots	$u(t_i) - u(t_0)$ Snapshots	$u(t_i) - u(t_{i-1})$ Snapshots
Burger's Eqn	1.46%	1.32%	1.84%
TransLine - cos†	1.60×10^{-2}%	1.60×10^{-2}%	6.52×10^{-3}%
TransLine - exp†	1.56×10^{-2}%	1.56×10^{-2}%	4.79×10^{-2}%
FHN†	0.432%	0.432%	∼150%
HNL2dSS	N/A	N/A	N/A
Heat Flow	-	-	-
MEMS	37.2%	0.0617%	0.767%

†These systems have an initial condition equal to the zero vector, so the snapshot collections $u(t_i)$ and $u(t_i) - u(t_0)$ are identical.

Table 4: Summary of TPWL Linearization Point Selection Algorithm Comparison

L_2 Error	Trajectory Curvature	Trajectory Distance	Residual Distance
Burger's Eqn	1.77%	26.2%	15.2%
TransLine - cos†	∼1600%	∼900%	40%
TransLine - exp†	∼4000%	∼1400%	8.42%
FHN†	∼130%	∼550%	90%
HNL2dSS	N/A	N/A	N/A
Heat Flow	-	-	-
MEMS	0.195%	5.88%	∼150%

Table 5: Summary of Robustness Analysis

L_2 Error	G-POD	PG-POD	TPWL	Gappy
Burger's Eqn	1.86%	1.81%	14.6%	1.81%
TransLine - cos†	8.25×10^{-3}%	1.53×10^{-2}%	39.1%	2.37×10^{-2}%
TransLine - exp†	8.36×10^{-3}%	1.66×10^{-2}%	8.31%	2.34×10^{-2}%
FHN†	6.58%	6.78%	∼150%	6.80%
HNL2dSS	N/A	N/A	N/A	N/A
Heat Flow	-	-	-	-
MEMS	1.00%	6.49%	∼122%	14.8%

Table 6: Optimal Parameters for All Nonlinear Problems

	FOM	ROM	G-Snapshots	PG-Snapshots	LinPt Algorithm	GappyParameters
Burger's Eqn	4001	40	$u(t_i) - u(t_0)$	$u(t_i) - u(t_0)$	Trajectory Curvature	(130,40,130)
TransLine - cos†	4000	30	$u(t_i) - u(t_{i-1})$	$u(t_i) - u(t_0)$	Residual Distance	(50,30,60)
TransLine - exp†	4000	30	$u(t_i) - u(t_0)$	$u(t_i) - u(t_0)$	Residual Distance	(50,30,60)
FHN†	2048	20	$u(t_i) - u(t_0)$	$u(t_i) - u(t_0)$	Residual Distance	(90,50,110)
HNL2dSS	2304	20	N/A	N/A	N/A	(80,60,80)
Heat Flow	961	25	-	-	-	-
MEMS	880	10	$u(t_i) - u(t_0)$	$u(t_i) - u(t_0)$	Trajectory Curvature	(32,22,42)

Table 7: Summary of Final Scaled-Up MOR Comparison

(L_2 Error; Speed-Up)	G-POD	PG-POD	TPWL	Gappy
Burger's Eqn	(2.40%; 1.20)	(2.66%; 1.19)	(7.24%; 19,445)	(2.82%; 59.7)
TransLine - cos†	($\sim 10^{-15}$%; 1.00)	(1.28×10^{-4}%; 1.00)	(39.4%; 20,238)	(8.96×10^{-3}%; 184)
TransLine - exp†	($\sim 10^{-14}$%; 0.98)	(6.51×10^{-5}%; 0.97)	(8.42%; 54,015)	(8.69×10^{-3}%; 157)
FHN†	(7.87×10^{-3}%; 1.17)	(5.75×10^{-3}%; 1.15)	(∼130%; 12,913)	(5.75×10^{-3}%; 45)
HNL2dSS	(50.9%; 7.50)	(∼180%; 6.64)	-	(85.5%; 12.8)
Heat Flow	-	-	-	-
MEMS	(0.0149%, 1.22)	(0.194%, 1.10)	(0.195%, 19)	(0.0120%, 15)

References

[1] David Amsallem. *Interpolation on Manifolds of CFD-Based Fluid and Finite Element-Based Structural Reduced-Order Models for On-Line Aeroelastic Predictions.* PhD thesis, Stanford University, 2010.

[2] Athanasios Antoulas. *Approximation of Large-Scale Dynamical Systems.* Society for Industrial and Applied Mathematics, 2005.

[3] K. Carlberg, C. Farhat, and C. Bou-Mosleh. Efficient Nonlinear Model Reduction via a Least-Squares Petrov-Galerkin Projection and Compressive Tensor Approximations. *International Journal for Numerical Methods in Engineering,* in press, May 2010.

[4] Saifon Chaturantabut and Danny Sorensen. Discrete empirical interpolation for a nonlinear model reduction. Technical report, Rice University, 2009.

[5] Thilo Penzl. Algorithms for model reduction of large dynamical systems. *Linear Algebra and Its Applications*, 2006.

[6] Michal Jerzy Rewienski. *A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems.* PhD thesis, Massachusetts Institute of Technology, 2003.

[7] T. Thanh and K. Willcox. Model reduction for large-scale cfd applications using the balanced proper orthogonal decomposition. Technical report, 17th AIAA Computational Fluid Dynamics Conference, Toronto, Ontario Canada, 2005.