REDUCED ORDER MODELING

FOR HIGH SPEED FLOWS WITH MOVING SHOCKS

**DISSERTATION**

David J. Lucia
Major, USAF

AFIT/DS/ENY/01-03

**DEPARTMENT OF THE AIR FORCE**

**AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

**Wright-Patterson Air Force Base, Ohio**

Approved for public release; distribution unlimited

# Report Documentation Page

| Report Date | Report Type | Dates Covered (from... to) |
|---|---|---|
| 03 Dec 2001 | Final | Sept 2000 - Nov 2001 |

| | |
|---|---|
| **Title and Subtitle**<br>Reduced Order Modeling For High Speed Flows with Moving Shocks | **Contract Number** |
| | **Grant Number** |
| | **Program Element Number** |
| **Author(s)**<br>Major David J. Lucia, USAF | **Project Number** |
| | **Task Number** |
| | **Work Unit Number** |
| **Performing Organization Name(s) and Address(es)**<br>Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Bldg 640 WPAFB, OH 45433-7765 | **Performing Organization Report Number**<br>AFIT/DS/ENY/01-03 |
| **Sponsoring/Monitoring Agency Name(s) and Address(es)**<br>AFRL/VASD ATTN: Philip S. Beran 2210 Eighth Street, Bldg. 146 WPAFB, OH 45433-7531 | **Sponsor/Monitor's Acronym(s)** |
| | **Sponsor/Monitor's Report Number(s)** |

**Distribution/Availability Statement**
Approved for public release, distribution unlimited

**Supplementary Notes**

**Abstract**
The use of Proper Orthogonal Decomposition (POD) for reduced order modeling (ROM)of fluid problems is extended to high-speed compressible fluid flows. The challenge in using POD for high-speed flows is presented by the presence of moving discontinuities in the flow field. To ovecome these difficulties, a domain decomposition approach is developed that isolates the region containing the moving shock wave for special treatment. The domain decomposition implementtion produces internal boundaries between the various domain sections. The domains are linked using optimization-based solvers which employ constraints to ensure smoothness in overlapping portions of the internal boundary. This approach is applied to three problems with increasing difficulty. The accuracy and order reduction of the domain decomposition POD/ROM approach is quantified for each application. ROMs with as large as three orders of magnitude reduction in degrees of freedom (DOFs) produce flow fields with maximum errors below 5%. One order of magnitude in computational savings for the non-Galerkin solver implementations accompanies this reduction in DOFs. Finally, the robustness of the reduced order models across a wide parameter space is demonstrated.

| | |
|---|---|
| **Subject Terms**<br>Proper orthogonal decomposition, reduced order modeling, moving shock waves, high-speed flow, aeroelastic analysis, domain decomposition, optimization-based solver | |
| **Report Classification**<br>unclassified | **Classification of this page**<br>unclassified |
| **Classification of Abstract**<br>unclassified | **Limitation of Abstract**<br>UU |
| **Number of Pages**<br>181 | |

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

AFIT/DS/ENY/01-03

# REDUCED ORDER MODELING
# FOR HIGH SPEED FLOWS WITH MOVING SHOCKS

**DISSERTATION**

Presented to the Faculty of the Graduate School of Engineering and Management

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

David J. Lucia, B.S., M.S.

Major, USAF

November 2001

Approved for public release; distribution unlimited

AFIT/DS/ENY/01-03

# REDUCED ORDER MODELING FOR HIGH SPEED FLOWS WITH MOVING SHOCKS

**DISSERTATION**

David J. Lucia, B.S., M.S.

Major, USAF

Approved:

| | |
|---|---|
| Paul I. King<br>Committee Chairman | 16 Nov 01<br>Date |
| Mark E. Oxley<br>Committee Member | 16 Nov 01<br>Date |
| Meir Pachter<br>Committee Member | 19 Nov 01<br>Date |
| Philip S. Beran<br>Committee Member | 19 Nov 01<br>Date |
| Gary B. Lamont<br>Dean's Representative | 19 Nov 01<br>Date |

Robert A. Calico, Jr
Dean

## *Preface*

This research project benefitted from the guidance and inspiration of many colleagues and fellow students. Specifically, I wish to thank Dr. Paul King, Dr. Phil Beran, Dr. Mark Oxley and Dr. Meir Pachter for their guidance at various critical junctures of the research. The interaction and example of my fellow students provided stimulation and support throughout. Specifically, I am grateful for the help of Major John Anttonen for proof reading my documents, and helping with a variety of research related issues. I thank Major Ray Maple for allowing me to use his finite volume flow solver as my research code. My gratitude also extends to Capt Jeff Hebert for teaching me to use Latex. Finally, I must mention the support of my wife, and the hand of Providence which were essential to the prosperity I have enjoyed in this endeavor.

David J. Lucia

*Table of Contents*


Page

Preface . . . . . iii

List of Figures . . . . . ix

List of Tables . . . . . xii

Abstract . . . . . xiii

Abbreviations . . . . . xiv

Nomenclature . . . . . xv

I. Introduction . . . . . 1-1

1.1 Overview of Proper Orthogonal Decomposition . . . . . 1-2

1.2 Previous Work in Domain Decomposition . . . . . 1-6

1.3 Scope . . . . . 1-7

1.4 Research Approach . . . . . 1-8

1.4.1 1-D Analysis . . . . . 1-9

1.4.2 2-D Quasi Steady Analysis . . . . . 1-9

1.4.3 2-D Unsteady Analysis . . . . . 1-9

1.5 Research Questions . . . . . 1-9

1.6 Document Organization . . . . . 1-10

II. Methodology . . . . . 2-1

2.1 Fluid Modeling . . . . . 2-1

2.2 Domain Decomposition . . . . . 2-4

2.3 Proper Orthogonal Decomposition . . . . . 2-5

2.4 Reduced Order Modeling . . . . . 2-7

Page


2.4.1 Galerkin vs. Subspace Projection Methods . . . . . 2-7

2.4.2 Steady Formulation . . . . . 2-8

2.4.3 Time Accurate Formulation . . . . . 2-10

2.5 Domain Coupling with Optimization . . . . . 2-11

2.6 Summary . . . . . 2-14

III. Analysis of a One-Dimensional Nozzle Flow . . . . . 3-1

3.1 Introduction . . . . . 3-1

3.2 Quasi 1-D Nozzle . . . . . 3-1

3.3 Shocks with POD . . . . . 3-3

3.4 Robustness of POD/ROM . . . . . 3-7

3.5 Domain Decomposition . . . . . 3-9

3.5.1 Explicit Formulation and Results . . . . . 3-10

3.5.2 Quasi-Steady Formulation with Full Order Shock Region . . . . . 3-11

3.5.3 Quasi-Steady Formulation with Reduced Order Shock Region . . . . . 3-12

3.6 Quasi-Steady Results . . . . . 3-14

3.7 Conclusions . . . . . 3-16

IV. Analysis of a 2-D Blunt Body Problem . . . . . 4-1

4.1 Introduction . . . . . 4-1

4.2 2-D Blunt Body Problem . . . . . 4-1

4.3 Domain Decomposition . . . . . 4-1

4.4 Steady Formulation . . . . . 4-3

4.4.1 Domain Decomposition Solver . . . . . 4-5

4.5 Results . . . . . 4-6

4.5.1 Full Order Shock Region Results . . . . . 4-6

4.5.2 Reduced Order Shock Region Results . . . . . 4-10

4.6 Conclusions . . . . . 4-16

Page


V. Analysis of a Transonic Aeroelastic Problem . . . . . 5-1

5.1 Introduction . . . . . 5-1

5.2 Formulation . . . . . 5-1

5.2.1 Structural Dynamics Equations . . . . . 5-1

5.2.2 Fluid Equations . . . . . 5-3

5.2.3 Grid Generation . . . . . 5-3

5.2.4 Fluid-Structure Coupling . . . . . 5-4

5.2.5 Domain Decomposition . . . . . 5-6

5.2.6 Reduced Order Modeling . . . . . 5-6

5.2.7 Implicit Formulation with Overlapping Domains . . 5-7

5.3 Results . . . . . 5-10

5.3.1 Full-System Transonic LCO . . . . . 5-11

5.3.2 Solver Implementation . . . . . 5-14

5.3.3 Accuracy, Order Reduction and Compute Time . . . 5-16

5.3.4 Robustness . . . . . 5-19

5.4 Conclusions . . . . . 5-21

VI. Summary . . . . . 6-1

6.1 Conclusions . . . . . 6-1

6.1.1 Domain Decomposition . . . . . 6-2

6.1.2 Applications . . . . . 6-3

6.2 Significant Advances . . . . . 6-4

6.2.1 Shock Capturing with POD . . . . . 6-5

6.2.2 Domain Decomposition . . . . . 6-5

6.2.3 Optimization-Based Solvers . . . . . 6-5

6.2.4 Transonic Aeroelastic Panel Response . . . . . 6-5

6.3 Summary and Future Work . . . . . 6-6

Page

Bibliography . . . . . BIB-1

Appendix A. Numerical Analysis . . . . . A-1

A.1 Mathematical Foundations . . . . . A-1

A.2 Foundations for Reduced Order Modeling . . . . . A-5

A.2.1 Overview . . . . . A-5

A.2.2 POD/ROM Applied to a Fluid Problem . . . . . A-6

A.3 Summary of POD/ROM . . . . . A-17

Appendix B. Additional Results for Quasi 1-D Nozzle . . . . . B-1

B.1 Replicating a 100 Second High Dynamic Flow Evolution . . . B-1

B.2 Replicating a 30 Second Low Dynamic Flow Evolution . . . . B-5

B.3 Summary of Key Insights . . . . . B-7

Appendix C. Rocket Nozzle Flow Control Using a Reduced Order Fluid Dynamics Model . . . . . C-1

C.1 Introduction . . . . . C-3

C.2 Problem Statement . . . . . C-4

C.3 Controller Synthesis . . . . . C-9

C.4 Fluid Model . . . . . C-11

C.5 Reduced Ordered Modeling . . . . . C-13

C.6 Results . . . . . C-16

C.7 Conclusions . . . . . C-19

Appendix D. Validation of Two-Dimensional Euler Code . . . . . D-1

D.1 Subsonic and Transonic Validation Cases . . . . . D-1

D.2 Supersonic Validation Case . . . . . D-4

D.3 Transpiration Boundary Condition Validation . . . . . D-5

D.4 Aeroelastic Panel Response . . . . . D-6

Page

Appendix E. Data and Software Archives . . . . . E-1

E.1 Description . . . . . E-1

E.1.1 Quasi 1-D Nozzle Archives . . . . . E-1

E.1.2 2-D Blunt Body Archives . . . . . E-1

E.1.3 Aeroelastic Panel Archives . . . . . E-2

E.2 Example Run . . . . . E-4

Appendix F. Analysis of Domain Size on Panel Response . . . . . F-1

Vita . . . . . VITA-1

*List of Figures*


| Figure | | Page |
|---|---|---|
| 3.1. | Quasi 1-D Nozzle | 3-2 |
| 3.2. | Density Function $t = 0 \rightarrow 7s$, $t = 7 \rightarrow 30s$, and $t = 30 \rightarrow 100s$ | 3-4 |
| 3.3. | Density Modes $100s$ Case $1 \rightarrow 10$ | 3-5 |
| 3.4. | Density Modes $30s$ Case $1 \rightarrow 10$ | 3-6 |
| 3.5. | Shock Motion with Varying $\gamma$ | 3-8 |
| 3.6. | Domain Decomposition for the Quasi 1-D Nozzle | 3-10 |
| 3.7. | POD/ROM/DD Density Comparison with 250 Grid Points | 3-11 |
| 3.8. | Shock Motion with Varying $\gamma$ | 3-14 |
| 3.9. | POD/ROM/DD Shock Location in the Nozzle | 3-15 |
| 3.10. | POD/ROM/DD Density Error | 3-16 |
| 4.1. | Blunt Body Geometry (non-dimensional) | 4-2 |
| 4.2. | Steady State for Mach 5 (Left) and Mach 2.5 (Right) | 4-3 |
| 4.3. | Domain Decomposition | 4-4 |
| 4.4. | $C_P$ Percent Error on Blunt Body | 4-8 |
| 4.5. | $C_L$ from Steady Analysis | 4-9 |
| 4.6. | Error in $C_L$ for Steady Analysis | 4-10 |
| 4.7. | Density Modes for Shock Region | 4-14 |
| 4.8. | $C_P$ Error for POD/ROM/DD | 4-15 |
| 5.1. | Grid for Flow over Panel | 5-4 |
| 5.2. | Domain Decomposition | 5-5 |
| 5.3. | Panel Response with Changing Dynamic Pressure | 5-11 |
| 5.4. | Moving Transonic Shock | 5-12 |
| 5.5. | LCO Onset | 5-13 |
| 5.6. | $C^1$ Constraints | 5-15 |

Figure Page


5.7. Modeled Panel Response . . . . . 5-17

5.8. LCO for $\lambda = 2500$ . . . . . 5-20

5.9. LCO for $\lambda = 2000$ (trained at $\lambda = 2500$) . . . . . 5-21

A.1. Notional Trajectory within the Operator Manifold . . . . . A-4

A.2. Obtaining a Reduced Order Space . . . . . A-8

A.3. Linear Operations between $E_Q$ and $E_N$ . . . . . A-11

A.4. Operation on the Reduced Order Space . . . . . A-17

B.1. $100s$ Density Function Flow Evolution . . . . . B-2

B.2. $100s$ Density Function Flow Evolution . . . . . B-5

B.3. Density Function $t = 0 \rightarrow 30s$ . . . . . B-6

C.1. Thrust From Unmodified Rocket . . . . . C-5

C.2. Rocket Engine Schematic . . . . . C-6

C.3. State Vector of Flow Variables . . . . . C-8

C.4. Open Loop Control Diagram . . . . . C-9

C.5. Solution Process . . . . . C-10

C.6. Variable Throat Nozzle Geometry . . . . . C-12

C.7. Optimal Controller Thrust Performance . . . . . C-16

C.8. Optimal Controller Thrust Error . . . . . C-17

C.9. Optimal Controller Actuator Commands . . . . . C-18

C.10. Sub-Optimal Thrust Performance . . . . . C-19

C.11. Sub-Optimal Thrust Error . . . . . C-20

D.1. Grid for Subsonic and Transonic Validation . . . . . D-1

D.2. Subsonic Surface Pressure . . . . . D-2

D.3. Transonic Surface Pressure . . . . . D-3

D.4. COBALT Surface Pressure Comparison for Mach 0.789 . . . . . D-4

D.5. Supersonic Validation at Mach 2 . . . . . D-4

Figure Page

D.6. Transpiration Boundary Condition Comparison . . . . . D-5

D.7. Unsteady Validation Data . . . . . D-6

F.1. Panel Problem Domain Decomposition . . . . . F-1

F.2. Unsteady Validation Data . . . . . F-2

*List of Tables*


| Table | | Page |
|---|---|---|
| 4.1. | PC Specifications | 4-7 |
| 5.1. | Computational Costs | 5-18 |
| 5.2. | Work Station Specifications | 5-18 |
| B.1. | Maximum Error Norm for 100$s$ Trajectory | B-8 |
| B.2. | $L^2$ Error Norm for 100$s$ Trajectory | B-9 |
| B.3. | Time Integrated $L^2$ Error Norm for 100$s$ Trajectory | B-10 |
| B.4. | Maximum Error Norm for 30$s$ Trajectory | B-11 |
| B.5. | $L^2$ Error Norm for 30$s$ Trajectory | B-11 |
| B.6. | Time Integrated $L^2$ Error Norm for 30$s$ Trajectory | B-12 |
| D.1. | Flow Values Behind Shock at Mach 2: Exact Vs. Euler Code | D-5 |

AFIT/DS/ENY/01-03


*Abstract*

The use of Proper Orthogonal Decomposition (POD) for reduced order modeling (ROM) of fluid problems is extended to high-speed compressible fluid flows. The challenge in using POD for high-speed flows is presented by the presence of moving discontinuities in the flow field. To overcome these difficulties, a domain decomposition approach is developed that isolates the region containing the moving shock wave for special treatment. The domain decomposition implementation produces internal boundaries between the various domain sections. The domains are linked using optimization-based solvers which employ constraints to ensure smoothness in overlapping portions of the internal boundary.

This approach is applied to three problems with increasing difficulty. The ability of POD/ROM to model a high-speed quasi 1-D nozzle flow is analyzed first. The quasi 1-D nozzle contains a standing shock which is moved in a quasi-steady manner by varying the ratio of specific heats ($\gamma$). Next, reduced order modeling for quasi-steady shock motion in a two-dimensional high-speed flow is studied. A 2-D blunt body flow containing a strong bow shock is the subject of this analysis. The bow shock is moved by varying boundary parameters. Lastly, the approach is applied to a transonic aeroelastic problem that involves large motions of a normal shock. 2-D inviscid flow over an elastic panel produces transonic limit cycle oscillations under certain panel parameters and free stream conditions. Panel flutter in the transonic regime results in the large motion of a transonic normal shock across the panel surface. Previously, no reduced order modeling method has been successfully applied to this problem.

The accuracy and order reduction of the domain decomposition POD/ROM approach is quantified for each application. ROMs with as large as three orders of magnitude reduction in degrees of freedom (DOFs) produce flow fields with maximum errors below 5%. One order of magnitude in computational savings for the non-Galerkin solver implementations accompanies this reduction in DOFs. Finally, the robustness of the reduced order models across a wide parameter space is demonstrated.

*Abbreviations*

| | |
|---|---|
| AoA | Angle of attack |
| CFD | Computational fluid dynamics |
| CFL | Courant, Friedrichs, and Lewy stability criteria |
| DD | Domain decomposition |
| DOF | Degree of freedom |
| LCO | Limit cycle oscillation |
| ODE | Ordinary differential equations |
| MPFV | Modes per fluid variable |
| POD | Proper orthogonal decomposition |
| ROM | Reduced order model |
| POD/ROM | Any of a number of reduced order implementations using POD to approximate fluid dynamics on a bounded domain |
| POD/ROM/DD | Use of domain decomposition to couple several POD/ROMs into a a single reduced order model (sometimes one of the domain sections is modeled at full order) |
| SVD | Singular value decomposition |

## *Nomenclature*

*Aerodynamic Symbols*

| | |
|---|---|
| $E$, $F$ | Vector of X-axis and Y-axis fluxes |
| $E_T$ | Total energy, $E_{Td}/\rho_\infty u_\infty^2$ |
| $L$ | Reference length, dimensional |
| $M$ | Mach number |
| $P$ | Pressure $P_d/\rho_\infty u_\infty^2$ |
| $t$ | Time, $t_d u_\infty/L$ |
| $u$, $v$ | Fluid velocity components, $u_d/u_\infty$ $v_d/v_\infty$ |
| $\underline{U}$, $\underline{\widehat{U}}$ | Full and reduced order vector of conserved flow variables |
| $\underline{w}$, $\underline{\widehat{w}}$ | Full or reduced order vector for a single fluid variable |
| $x$, $y$ | Spatial coordinates, $x_d/L$, $y_d/L$ |
| $\Delta\underline{t}^{local}$ | Local time stepping vector of time steps, one for each node |
| $\Delta t$ | Global time step determined to preserve stability for the smallest node spacing |
| $\gamma$ | Ratio of Specific Heats, 1.4 |
| $\rho$ | Density $\rho_d/\rho_\infty$ |

*Structural Symbols*

| | |
|---|---|
| $A(x)$ | Nozzle cross sectional area |
| $D$ | Plate stiffness, $E_s h^3/12(1-\nu^2)$ |
| $E_s$ | Young's modulus, dimensional |
| $h$ | Panel thickness, dimensional |
| $\underline{P}$ | Array of pressure differentials on panel |
| $s$ | Panel deflection speed, $\dot{w}$ |
| $w$ | Normal panel deflection, $w_d/L$ |
| $\underline{S}$ | Vectors of $s$ values on panel |
| $\underline{W}$ | Vectors of $w$ values on panel |
| $\underline{Y}_s$ | Structural state array, $[\underline{S}, \underline{W}]^T$ |
| $\lambda$ | Nondimensional dynamic pressure, $\rho_\infty L/\rho_s h$ |

| | |
|---|---|
| $\mu$, $\nu$ | Mass ratio $\rho_\infty u_\infty^2 L^3/D$, Poisson's ratio 0.3 |
| $\rho_s$ | Structural density, dimensional |

*Numerical Symbols*

| | |
|---|---|
| $C$ | Constraint type |
| $F()$ | Modified flow field expression for time accurate Newton's iterations |
| $G$ | Vector valued function for overlapping domains |
| $I$ | Identity matrix |
| $M$ | Number of DOFs for reduced order model |
| $N$ | Number of DOFs for full system |
| $Q$ | Number of snapshots used in POD/ROM |
| $R$ | Flux calculation from the Euler equations |
| $S$ | Matrix of flow field data, or snapshots |
| $T$ | Vector mapping overlapping variables |
| $V$ | Matrix of eigenvectors of $S^T S$ |
| $\underline{y}$, $\underline{\widehat{y}}$ | Flow variables $\underline{U}$, $\underline{\widehat{U}}$ augmented with LaGrange multipliers |
| $\Lambda$ | Matrix of singular values of $S^T S$ |
| $\lambda_i$ | Lagrange multiplier for $i^{th}$ constraint |
| $\Psi$ | Reduced order mapping matrix |

*Subscripts*

| | |
|---|---|
| 0 | Base flow |
| $d$ | Dimensional quantity |
| $DD$ | Domain Decomposition |
| $i$ | Constraint index |
| $s$ | Structural quantity |
| $ss$ | Snapshots |
| $S$ | Section number within domain |
| $t$ | Time derivative |

| | |
|---|---|
| $x$, $y$ | Spatial derivative |
| $\lambda$ | Overlapping domains |
| $\infty$ | Freestream quantity, dimensional |

*Superscripts*

| | |
|---|---|
| $init$ | Initial data |
| $n$ | Index to increment discrete time steps |
| $over$ | Overlapping portion of domain |
| $trans$ | Translated to new free stream condition |
| $\nu$ | Index to increment Newton iterations |

# REDUCED ORDER MODELING
# FOR HIGH SPEED FLOWS WITH MOVING SHOCKS

## *I. Introduction*

The development of aerospace vehicles requires analysis and testing. For many applications, design analysis involves mathematical approximation via linearization of the governing dynamic equations. This enables many powerful analytical methods for the design of aircraft structures, components and controllers. Linearized analysis also facilitates linear design optimization for which many closed form solutions exist. The validity of designs from linearized methods can be tested in wind tunnels with scaled vehicle models whose performance at varying conditions is quantified and extrapolated to full scale. Computer models are also used to supplement wind tunnel tests for aerospace designs. While quite effective for many problems, this design methodology breaks down under conditions when linearized dynamics models become inaccurate, when it is infeasible to replicate flight conditions in a laboratory, or when the computational fluid models are so large that the cost for multiple runs is prohibitive.

The primary cause of long run times for fluid models is the large number of degrees of freedom (DOFs) from a discretized computational mesh. The number of arithmetic operations to produce an approximate flow solution is proportional to the number of DOFs. This is why large DOF fluid models require long computing times. Some flow simulations take days of supercomputer processing time.

The desire to use computational fluid dynamics (CFD) in design analysis has motivated a search for faster CFD solvers. Proper orthogonal decomposition (POD) of flow simulations was introduced in the mid 1990s to reduce the number of DOFs of flow solvers. The literature refers to such POD based fluid models as POD reduced order models (ROMs), where the term "order" is used to imply the number of degrees of freedom. POD/ROM has enjoyed a great deal of success for low speed flow fields. POD has been successfully applied to subsonic flow, and supersonic aeroelastic problems with stationary shocks (e.g.,

see (1, 2, 3, 4, 5, 6, 7, 8, 9)). POD has also been used for low speed airfoil design (10). Computers using the POD reduced order models can produce data fast enough to enable some types of flow control for incompressible flow fields (e.g., see (11, 12, 13, 14, 15, 16)).

The objective of this research is to extend the use of POD/ROM to high speed flow fields with moving shocks. Reduced order modeling for transonic aeroelastic problems is a challenging area of research, because transonic aerodynamics are characterized by unsteady normal shock waves. Other reduced order methods rely on some form of linearization to limit the degrees of freedom required to resolve the nonlinear dynamics. Linearization of a transonic flow field about some shock position will produce errant results when the shock moves any significant amount. Due to these difficulties, reduced order methods have not been successfully applied to the transonic moving shock problem. A shock capturing technique is developed that uses domain decomposition (DD) in conjunction with POD. This technique is necessary to overcome difficulties encountered when using POD to translate discontinuous behavior in the flow field. The POD/ROM/DD approach mitigates this difficulty by isolating the region of the flow field containing the discontinuity. This region (referred to as the "shock region") is either modeled at full order, or treated with a special POD/ROM constructed with sufficient data to allow shock motion.

### *1.1 Overview of Proper Orthogonal Decomposition*

POD is a technique to identify a small number of DOFs that adequately reproduce the behavior of a large DOF flow simulation. Solving the reduced order problem requires a reduced number of arithmetic operations and hence less computing time. The greater the reduction in DOFs, the faster the computer produces an outcome.

The main product of the POD process is a mapping between the low and high dimensional outcomes. Here "dimension" is interchanged with "degree of freedom," and the full system numerical flow solver yields the high dimensional outcomes. Each degree of freedom in the large DOF flow simulation corresponds to a dimension in Euclidean "$N$-space." The mapping is used to transform the high dimensional problem into a low dimensional problem in Euclidean "$M$-space" where $M << N$. The low dimensional problem is then solved quickly, and transformed back to the high dimensional solution.

The full order flow simulation is denoted as an operator $R$, whose domain and range are all the vectors in Euclidean $N$-space that meet the boundary conditions for the flow field of interest. The domain and range of $R$ are approximated by collecting many outcomes from the flow simulation for the specific problem being studied. These outcomes are called "snapshots" (17) since they are like pictures of the flow field at selected times. The set containing all possible linear combinations of the snapshots forms a linear space that approximates the domain and range of $R$. These snapshots typically numbers around 80 or 100 vectors per fluid variable. They must be carefully chosen so that all the vectors in the domain and range of $R$ can be reasonably represented by some linear combination of the snapshots. Since the number of snapshots (80 or 100) is generally much smaller than the number of grid points used by the flow solver, the number of degrees of freedom are greatly reduced by solving for 80 or 100 coefficients (per fluid variable). Note that the snapshots must be linearly independent for this technique to work. The next step in the POD process eases this restriction.

Every vector in the domain and range of $R$ can be reasonably represented as a linear combination of the snapshots. This linear combination can be written as a matrix equation. The matrix of snapshots, where each snapshot is a separate column, forms a non-symmetric linear transformation between any vector of coefficients and the full order vector that those coefficients represent. The vector of coefficients is also known as the coordinates of the full order vector, and each full order vector has a corresponding vector of coefficients or coordinates. Now consider the collection of vectors of coordinates that map to full order vector for the fluid problem of interest. Call this collection the "set of all coordinates."

This set of all coordinates consists of column vectors with dimension equal to the number of snapshots. The set of all coordinates will have a great deal of redundancy if the snapshots are linearly dependent (even if they are nearly linearly dependent). Singular Value Decomposition (SVD) is used on the non-symmetric matrix equation to identify an orthogonal basis for this set of all coordinates. Much like the set of snapshots formed a basis for the domain and range of $R$, SVD provides eigenvectors which form a basis for the set of all coordinates (18), which is the domain and range of a reduced order operator $\widehat{R}$. These eigenvectors are orthogonal, and their corresponding eigenvalues give a measure of

the significance each eigenvector has when recreating the vectors in the set of all coordinates (19). The redundancy in the snapshots is eliminated by truncating the eigenvectors with the smallest eigenvalues. This reduced order set of coordinates has the dimension of the number of eigenvectors with large eigenvalues. This is generally much smaller than the number of snapshots.

A new linear matrix equation can be generated that relates vectors in the truncated set of coordinates to vectors in the domain and range of $R$. This resulting non-symmetric matrix is called the reduced order mapping. The columns of this matrix are called the POD solution modes. They represent the coherent structures of the flow field captured by the snapshots.

The final step is to recast the nonlinear operator using these POD solution modes by either Galerkin (20, 19) or non-Galerkin methods (21). The high dimensional nonlinear operator $R$ acting on $N$-space, becomes a low-dimensional calculation $\widehat{R}$ acting on a much smaller $M$-space. Once the $M$ coefficients are obtained, the full order solution is assembled via the linear combination of the modes multiplied by the modal coefficients. This reduced order process projects the full order vectors into the linear space defined by the POD modes. If the full order vector is well represented by the snapshots, then the reduced order result is extremely accurate. If the full order vector has spatial structures that cannot be replicated via linear combination of the snapshots, errors occur.

*Previous Work in Proper Orthogonal Decomposition.* Proper Orthogonal decomposition was introduced by multiple sources in the 1940's (22, 23). Until the 1980's, the majority of applications dealt with identifying principle structures in large sets of statistical data taken from complex problems. Applications of POD included image processing, signal analysis, data compression, and process identification (19). The first application of POD to analysis of fluid flow was in 1967 (24), in which POD was used to identify dominant coherent structures in turbulent flow fields from wind tunnel data.

In the 1980's, the computer was advanced enough to enable widespread numerical solution to systems of partial differential equations. This gave rise to the use of computer solutions to previously unsolved problems. Finite-element methods for structures, and a

variety of numerical schemes for solving fluid problems emerged. The utility of such large DOF numerical solvers was hampered by the computational burden. Both the monetary cost of computing time, and the long wait for solutions combined to relegate the utility of such methods to those with access to supercomputers, and the time to wait days or weeks for a solution. This precluded the use of such models for iterative design analysis and online applications such as control. However, it provided a new tool for fluid dynamics research. One advance, direct numerical simulation of turbulence, opened a way for turbulence researchers to characterize an entire turbulent flow field. Prior to this, researchers were limited by experimental data taken at distinct locations in the flow. Direct simulation of turbulence provided a new way to study turbulence using coherent structures (or modes) obtained from numerical simulations via the POD technique (see (25) as an early example).

The next advance was in the use of POD modes to reconstruct lower order dynamics models of turbulence to replace expensive full order computer representations (such as direct numerical simulations, or large eddy simulations). The use of POD in reduced order modeling for numerical simulation of turbulent flow dates back to 1987 (17, 26). Since then, other applications of POD to reduced order modeling of turbulence have been reported in the literature (27, 28, 29). A textbook has been published on the subject as well (19). Other disciplines where the underlying dynamics require large DOF numerical simulations have started to investigate the use of POD/ROM.

The field of aeroelastic analysis has embraced POD, beginning in 1996 (1). Aeroelastic analysis with POD followed from previous work using eigenfunctions from linearized aerodynamics equations in building reduced order aeroelastic models (30). In (1), POD modes were used in place of the eigenfunctions from (30). This resulted in a very accurate reduced order model, without the computational difficulties of generating eigenfunctions from the linearized problem. Several aeroelastic modeling applications have been subsequently reported in the literature ((2), (3), (4), (5), (6), (7), (8)).

*Advantages of POD for Reduced Order Modeling.* Of the many order reduction techniques published, truncated modal representations for full order flow solvers were the only ones considered for this research because they promised the greatest order reduction. Modal representations of fluid solutions come in a variety of forms. Known as the spec-

tral methods for fluid dynamics, these techniques approximate the Euler or Navier-Stokes equations with a variety of basis functions (or modes) (20). For reduced order aerodynamic problems, a spectral method must generate reasonable solutions with a very few basis functions. There are only four ways to generate such a set of basis functions currently reported in the literature. These are wavelets (31), eigenmodes (32), balanced modes (5), and POD modes. The use of a wavelet basis is not considered because there is no simple method for identifying a wavelet basis from existing flow solvers for the general Navier-Stokes case. Instead, use of wavelets requires a full system model specifically designed to identify and use a wavelet based reduced order model (31). POD is more desireable for fluid modeling since POD modes are more easily generated from existing flow solvers. For a description of eigenmodes and balanced modes see (5). Both are computed by linearizing the governing aerodynamics equations, and solving the eigenvalue problem for the linearized system. This linearization process does not retain shock motion. POD has three advantages over other linear methods. First, POD modes are obtained from the fully nonlinear system. As a result the POD modes can capture all the nonlinearities of the full order representation. Second, POD modes are obtained from existing flow solvers via a simple data taking procedure. The other methods require linearization of the full order solver about a solution of interest. The resulting computation of eigenmodes is too computationally burdensome for large DOF problems. Finally, POD modes are optimal, meaning no other linear modal representation of the same dimension can more accurately recreate the full order solution (19). Therefore, POD modes will achieve a larger order reduction than the other linear methods.

The focus of this research is to expand the use of POD to reduced order modeling of high speed fluid flows. This requires applying POD to flow fields with moving strong shocks. Domain decomposition has potential to allow the use of POD in generating reduced order models for flow fields with moving discontinuities.

### *1.2 Previous Work in Domain Decomposition*

The application of parallel computing to numerical methods in the 1980's was closely followed by the widespread use of domain decomposition. Domain decomposition provided

a means to spread a large numerical simulation over many computer processors by spatially dividing a large problem into many smaller problems that could be solved simultaneously. In this way, domain decomposition and parallel computing improved the overall compute time for large numerical simulations. Domain decomposition for numerical solutions to fluid problems has been studied in several publications (e.g., see (33, 34, 35, 36, 37, 38, 39)).

Domain sections must be linked across internal boundaries if the parallel solution method is to produce a meaningful outcome. In the past two years, domain coupling with optimization-based solvers has emerged in the literature (40, 41, 42, 43, 44). The technique involves setting constraints on the shared internal boundaries that are enforced with Lagrange multipliers. The constraints can be formed on either overlapping (41) or non-overlapping (40, 42, 43, 44) internal boundaries. Simultaneous solution of the domain sections and the constraint equations represent an optimization problem. This has been successfully demonstrated on a numerical solution to the incompressible Navier-Stokes equations (42).

This research extends domain decomposition with optimization-based solvers to reduced order modeling of high-speed compressible flow problems. The presence of moving shocks represents a much stronger non-linearity than previously applied. In addition, the formulation and solvability of the reduced order optimization problem is demonstrated. The use of domain decomposition is used for stability of the reduced order method, not for parallelization.

### *1.3 Scope*

This objective of this research is to use domain decomposition to generate a shock capturing POD/ROM that efficiently and accurately reproduces high speed flow fields with moving shocks. The accuracy, order reduction and computational performance of this approach is demonstrated on 1-D and 2-D applications. The scope of this research is quantified in terms of the following thesis statement, the governing assumptions, and the outlined research approach.

*Thesis: Domain Decomposition enables the use of Proper Orthogonal Decomposition to generate reasonable low order approximations of 2-D high speed fluid flow fields with shock movement over confined regions.* Order reduction is determined relative to the number of DOFs required by a full system model of the same flow field without the use of POD. The utility of such reduced order models for design analysis is demonstrated through application to 1-D and 2-D flow fields. The accuracy of POD/ROM/DD with changing boundary conditions, free stream conditions, and flow parameters is quantified along with the associated order reduction.

*Assumptions:* A CFD representation of the Euler equations is used as the full system model in this research. Inviscid methods can be readily applied to viscous flow fields with similar accuracy and order reduction. Accuracy of the POD/ROM/DD is quantified via comparison with the flow field solution from the full system model. This comparison is accomplished through appropriate error metrics. The correctness of the full system model is checked against analytical results and/or similar results presented in the literature. The success of POD is completely determined by how accurately it represents the full order flow field representation from which it was derived.

### *1.4 Research Approach*

Both quasi-steady and unsteady flows are considered in this research. Quasi-steady cases are treated first, with applications in both 1-D and 2-D presented in order of increased complexity. A 1-D application was solved first to identify and address the difficulties encountered when applying POD/ROM with moving shocks. This initial investigation involves application of the domain decomposition POD/ROM approach to a quasi 1-D nozzle problem. Next, the domain decomposition approach is incorporated into a 2-D research code for quasi-steady analysis. This application explores how the techniques employed for the 1-D problem extend to higher dimensionality. Lastly, the 2-D code is apadpted for unsteady analysis of a transonic aeroelastic problem. Unsteady applications require different implementations than those used for quasi-steady analysis.

*1.4.1 1-D Analysis.* The first effort is to explore the use of POD/ROM on a 1-D high-speed flow field with a strong, moving shock. A shock capturing technique is developed that exploits POD and also accurately treats moving shock waves. This technique decomposes the solution domain to isolate regions that contain shocks. A reduced order model for each region is developed independently, and the solution for the entire domain is formed by linking the boundaries of each region to systematically solve for the flow field.

*1.4.2 2-D Quasi Steady Analysis.* The next step is to apply the POD/ROM/DD approach to a high-speed 2-D inviscid flow field, and assess the performance in accuracy and order reduction. High-speed flow over a 2-D blunt nosed cone is the model problem for this study. Changes in inlet Mach number and angle of attack (AoA) result in quasi-steady motion of a strong shock wave off the nose of the blunt body.

*1.4.3 2-D Unsteady Analysis.* The final application is an aeroelastic panel in cross-flow. A nonlinear coupling of the 2-D Euler equations and the von Kármán equation is used to simulate the dynamics of flow over a flexible panel. Inviscid flow over an elastic plate produces transonic limit cycle oscillations (LCO) in the presence of certain panel parameters and free stream conditions. Panel flutter in the transonic regime produces a transonic shock that traverses the panel surface. Previous research (9) for this problem neglected reduced order modeling of the transonic case because of the difficulties that moving shocks pose to POD/ROM. The development of a POD/ROM to treat the transonic case is the subject of this effort. The performance of the coupled system is assessed in terms of accuracy, order reduction, and computational expense. The robustness of the POD/ROM/DD to changes in panel dynamic pressure ($\lambda$) is explored to evaluate the model's utility for design analysis.

## *1.5 Research Questions*

The following statements summarize they key items for investigation in this dissertation.

*Can POD be used to generate a reduced order model for unsteady high speed shocked flow fields with similar order reduction and accuracy that has been reported for low speed problems?* Since such unsteady flows involve moving strong shocks, this research investigates POD's accuracy, order reduction and robustness in the presence of flow discontinuities.

*What is the best way to integrate POD modes into a reduced order model for high speed flows?* The method used for many low speed applications was Galerkin projection. Galerkin projection is problematic for compressible flow problems due to difficulties in treating nonlinear boundary conditions, therefore non-Galerkin approaches are considered in this research. Explicit non-Galerkin methods provide proven accuracy, but the computational efficiency improvement is questionable. Implicit non-Galerkin approaches and collocation methods are investigated as well.

### *1.6 Document Organization*

The remainder of this document is organized as follows. Chapter 2 develops the theory for the use of POD on shocked flows. It also outlines the methodology for implementation of the various reduced order solvers. Chapter 3 records the analysis and results for the quasi 1-D nozzle application, Chapter 4 records the 2-D blunt body analysis and results, and Chapter 5 contains the analysis and results for the transonic aeroelastic panel application. Chapter 6 lists the overall conclusions, highlights the significant advances from this research, and proposes future work.

# II. Methodology

This chapter contains a description of the numerical simulation of the Euler equations used throughout this research. A mathematical development of Proper Orthogonal Decomposition is also included, followed by the implementation methods for using POD modes to synthesize optimization-based reduced order solvers. The developments contained in this chapter are general, and additional problem specific details are addressed in the formulation sections of subsequent chapters.

## 2.1 Fluid Modeling

The dynamics of fluid flows are governed by the Navier-Stokes equations for viscous flow, or the Euler equations for inviscid flow. The two-dimensional Euler equations are given below in strong conservation form (45).

$$U_t + E_x + F_y = 0 \tag{2.1}$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E_T \end{bmatrix} \tag{2.2}$$

$$E = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho u v \\ (E_T + P)u \end{bmatrix} \tag{2.3}$$

$$F = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + P \\ (E_T + P)v \end{bmatrix} \tag{2.4}$$

Here $\rho$ is density, $\rho u$ is x-direction momentum, $\rho v$ is y-direction momentum, and $E_T$ is total energy per unit mass. These four fluid variables are functions of both space and time.

The solution of the flow field can be approximated by a wide variety of numerical techniques. To do this, both the spatial domain and the time dimension are discretized. An approximate flow field solution at the discrete locations is obtained by finite-difference, finite volume, or finite element techniques. A finite volume approach was used in this research. It requires the integral form of the Euler equations shown below (45).

$$\frac{d}{dt}\int_{\mathsf{V}} U d\mathcal{V} + \int_{\partial\mathsf{V}} (E\hat{\imath} + F\hat{\jmath}) \cdot d\overline{\mathcal{S}} = 0 \tag{2.5}$$

Solutions of the full system are obtained by numerically solving the integral Euler equations for small finite volumes within the flow field. The spatial discretization results in a computational mesh with many finite volumes, or "cells."

Stability of the numerical scheme necessitates small discrete steps, both spatially and in time. The spatial discretization results in a computational grid with many nodes. The nodes of the grid represent corners of the finite volumes, and the elements of the numerical flow solution are attributed to locations at the center of each cell. The number of cells increases to the power of the number of dimensions required (i.e. $1,000$ cells for a 1-D problem would be $100,000$ cells for a 2-D problem, and $1,000,000,000$ cells for 3-D). Nyquist sampling rates also apply, so the grid size must consider the wavelength of the dynamics at the node location. Viscous boundary layers are very small, thus viscous grids require many nodes in the region of the boundary layer. For these reasons, a large number of cells are required for numerical approximation. Each fluid variable at every cell is a separate degree of freedom. Numerical solvers are computationally expensive due to the large number of DOFs.

For each cell, the integral form of the Euler equations reduce to the following assuming no grid deformation.

$$\frac{d}{dt}U_{i,j} + \sum_{sides} (E_{i,j}\hat{\imath} + F_{i,j}\hat{\jmath}) \cdot \frac{d\overline{\mathcal{S}}_{i,j}}{d\mathcal{A}_{i,j}} = 0 \tag{2.6}$$

The flux terms $E_{i,j}$ and $F_{i,j}$ are computed using first-order Roe averaging (45). Row-by-row ordering is used to collocate flow variables at each spatial cell location to a corresponding location in a column vector producing an overall vector of flow variables $\underline{U}(t)$.

Time integration across the computational mesh is used to obtain flow solutions. This is accomplished with a first order accurate forward Euler approximation. Local time stepping is incorporated into the solver to speed up convergence for steady analysis. The need for small time steps increases the computational expense. Local time stepping can limit the compute time required for steady state solutions, but time accurate analysis cannot use this technique because it generates a non-physical flow evolution. Small time steps combined with the large number of DOFs combine to make design analysis and control impractical with this type of numerical method.

Since the Euler equations are linear in the time derivative, and quasi-linear in the spatial derivative (45, 18), the spatial derivatives and the time derivatives in equation (2.1) can be separated to form an evolutionary system. To accomplish this, the spatial terms are grouped to form a nonlinear operator $R$ acting on the set of fluid variables. The fluid dynamics from equation (2.6) can then be expressed as,

$$\frac{d\underline{U}(t)}{dt} = R(\underline{U}(t)) \ . \tag{2.7}$$

Equation (2.7) is referred to as the full system dynamics.

External boundaries are handled with ghost cells. The fluid values for the ghost cells at the far field boundaries are determined using characteristic boundary conditions (46). The ghost cells for the solid wall are set to enforce a no-flow-through condition. Moving solid boundaries are enforced with a transpiration boundary condition. A transpiration boundary condition replaces the solid wall with blowing or suction designed to mimic the effect of a solid boundary. For a static case, the no-flow-through condition at a solid wall implies that the flow momentum is tangent to the solid surface for all the cells adjacent to the surface. Flow momentum is injected (blowing) or removed (suction) such that the flow momentum at the solid surface is turned (via vector addition) as though the solid wall were present. For dynamic cases, additional blowing is required since the wall velocity

imparts additional momentum to the flow, over and above the momentum needed to turn the flow tangent to the solid surface (7, 47). Transpiration boundary conditions are valid for small wall deflections. They are a convenient way to model dynamic solid boundaries without moving grids. Moving grids require special treatment in the POD/ROM (48). Solver validation is detailed in appendix D.

### *2.2 Domain Decomposition*

Domain decomposition is used to separate the fluid problem spatially into regions, and isolate the region of the flow experiencing the shock wave. This assumes that the shock is confined to a spatial region of the flow field that can be reasonably identified. For the regions of the flow field not containing a shock, no special treatment is necessary. The region of the flow field containing a moving shock is approached in two ways. The first approach is to us the full system model. Even though the shocked region of the flow had no reduction in the number of DOFs, the use of POD/ROM over the non-shocked portions of the flow field still provides a significant reduction in the number of DOFs relative to the original problem. A second approach is to develop a reduced order model for the shocked region of the domain decomposed flow field.

The motivation for this approach is to improve the robustness of the reduced order model in the presence of moving shocks. The number of modes required to reproduce a flow field with a stationary shock are few, and they are relatively simple to identify. Identifying modes for flows with moving shocks requires much more data, enough to cover all shock locations expected. In iterative design analysis, identifying all the solution structures a priori might not be feasible. In these cases, only way to ensure accuracy is to use the full system for those regions of the flow not adequately sampled by the snapshots.

Since the domain is divided into several sections, analysis for the POD/ROM/DD involves solving several smaller fluid problems that are linked by internal boundaries. The full system dynamics in equation (2.7) applies to each individual domain section. Internal boundaries are handled with ghost cells which are filled with the corresponding values from the adjacent domain. Domain overlap is necessary under certain conditions that will be outlined in later chapters. The domain decomposition approach produces flexibility in

choosing the set of snapshots and number of modes used to create the POD/ROM for each section.

### *2.3 Proper Orthogonal Decomposition*

POD is a technique to identify a small number of DOFs that adequately reproduce the behavior of the full system. The greater the reduction in DOFs, the faster the computer produces a solution for each time step. Time step size for stability can also be improved using POD, since the application of POD modes exclude some of the high frequency content that brings instability. A summary of POD as it applies to a spatially-discretized flow field follows. A detailed description of POD is contained in appendix A.

For simplicity, only consider one fluid variable, $w(t)$. Since the value of $w(t)$ is obtained using numerical approximation, $w(t)$ is spatially descritized using $N$ nodes, and collocated into a large array denoted $\underline{w}(t)$. For this fluid variable, the full system dynamics in equation (2.7) is depicted below,

$$\frac{d\underline{w}}{dt} = R(\underline{w}) \ . \tag{2.8}$$

POD produces a linear transformation $\Psi$ between the full system solution $\underline{w}$, and the reduced order solution for this fluid variable $\underline{\widehat{w}}$ is given by

$$\underline{w}(t) = \underline{W}_0 + \Psi\underline{\widehat{w}}(t) \ . \tag{2.9}$$

Note that $\Psi$ is not time varying, where $\underline{w}$ and $\underline{\widehat{w}}$ are functions of time $t$. The POD reduced order variable $\underline{\widehat{w}}(t)$ represents deviations of $\underline{w}(t)$ from a base solution $\underline{W}_0$. The subtraction of $\underline{W}_0$ will result in zero-valued boundaries for the POD modes wherever constant boundary conditions occur on the domain. It is required when POD modes are applied to flow problems with changing free stream conditions.

$\Psi$ is constructed by collecting observations of the solution $\underline{w}(t) - \underline{W}_0$ at different time intervals throughout the time integration of the full system dynamics. These observations are called snapshots and are generally collected to provide a good variety of flow field

dynamics while minimizing linear dependence. Alternatively, a collection of steady state solutions can be used as snapshots. The snapshot generation procedure is sometimes referred to as the POD training period (21).

A total of Q snapshots are collected from the full system dynamics. These are vectors of length $N$. The set of snapshots describe a linear space that is used to approximate both the domain and the range of the nonlinear operator $R$. The linear space is defined as the span of the snapshots (49). POD identifies a new basis for this linear space that is optimally convergent (19). To identify the POD basis, the snapshots are compiled into a $N \times Q$ matrix $S$, known as the snapshot matrix. The mapping function $\Psi$ is then developed using,

$$S^T S V = V \Lambda \tag{2.10}$$

$$\Psi = S V \tag{2.11}$$

Here $V$ is the matrix of eigenvectors of $S^T S$, and $\Lambda$ is the corresponding diagonal matrix of eigenvalues. To eliminate redundancy in the snapshots, the columns of $V$ corresponding to very small eigenvalues in $\Lambda$ are truncated. The matrix of eigenvalues $\Lambda$ is also resized to eliminate the rows and columns corresponding to the removed eigenvalues. If $Q - M$ columns of $V$ are truncated, the resulting reduced order mapping $\Psi$ will be an $N \times M$ matrix. This reduced order mapping is a modal representation of the flow field. The modes are the columns of $\Psi$ which are discretized spatial functions fixed for all time. The vector $\underline{\widehat{w}}(t)$ is a time dependent set of coefficients representing the coordinates of $\underline{w}(t)$ projected into the truncated linear space described by the POD basis.

The solution to the fluid dynamics requires simultaneous solution for the set of fluid variables. The reduced order mappings for each fluid variable are developed individually. Each fluid variable is collocated into a single array $\underline{U}$, and the corresponding $S$ and $V$ arrays are collocated as blocks into a larger set of arrays, also denoted $S$ and $V$. The reduced order mapping is now,

$$\underline{U}(t) = \underline{U}_0 + \Psi \underline{\widehat{U}}(t) \tag{2.12}$$

This version of equations (2.9) and (2.11) applies to the entire set of fluid variables simultaneously.

### *2.4 Reduced Order Modeling*

A reduced order model is obtained by recasting the full system dynamics to solve for the reduced order variable $\underline{\widehat{U}}(t)$. Several methods are contained in the literature (21), and selection of the appropriate implementation depends on the solution strategy. Both implicit and explicit time accurate methodologies will be developed in this research. Implicit time stepping will be an efficient way to handle the regions of the domain decomposed flow field when the Jacobian does not need to be recomputed regularly. Explicit time stepping will be more efficient in regions of the flow containing stronger nonlinearities.

*2.4.1 Galerkin vs. Subspace Projection Methods.* For reduced order modeling using POD modes, Galerkin projection is a common implementation method. It uses the POD modes to recast the original dynamics equations into a small set of non-linear ordinary differential equations (ODEs) in time (19). The recent applications of POD/ROM to low speed flow using Galerkin projection have all treated incompressible flow (50). This avoids complications inherent when applying Galerkin projection to the Euler equations. The difficulty lies in the appropriate treatment of boundary conditions at a solid surface for density and energy. For this analysis, a non-Galerkin approach was used for simplicity. The non-Galerkin approach, also known as the subspace projection method (3), uses the full system dynamics in equation (2.7), a forward difference approximation, and the left inverse of the reduced order mapping from equation (2.11) to yield the following reduced order flow solver.

$$\frac{d\underline{\widehat{U}}}{dt} = \Lambda^{-1}(V^T V)^{-1} V^T S^T R(SV\underline{\widehat{U}}) \tag{2.13}$$

The inverse of $\Lambda$ and pseudo inverse of $V$ exist assuming modal truncation is employed to eliminate the zero valued eigenvalues of $S^T S$ and their corresponding eigenvectors (see appendix A for a full development). The subspace projection method relies on the full order function evaluation $R$. As such, the order of the operator is not actually reduced. However, this allows for study of POD accuracy and order reduction without the use of

Galerkin projection. Also, compute time savings can be achieved for both explicit and implicit solver implementations. This will be demonstrated in the results. The application of Galerkin projection to compressible flow problems with nonlinear, coupled boundaries will be discussed in Future Research.

*2.4.2 Steady Formulation.* Although the equations governing the reduced order operator $\widehat{R}(\underline{\widehat{U}})$ are never explicitly obtained by the subspace projection method, the value of the reduced order operator at any time can be obtained from equation (2.13) using the full order function evaluation $R(\underline{U})$,

$$\widehat{R}(\underline{\widehat{U}}) = \Lambda^{-1}(V^T V)^{-1} V^T S^T R(SV\underline{\widehat{U}}) \ . \tag{2.14}$$

For a reduced order vector of flow variables $\underline{\widehat{U}}$, the steady-state solution is obtained when the reduced order operator $\widehat{R}(\underline{\widehat{U}})$ vanishes. This solution can be obtained by either explicitly time marching to steady state, or by solving implicitly with Newton iterations.

*2.4.2.1 Explicit Method.* Explicit integration to steady state for the reduced order solver is accomplished using forward Euler time integration,

$$\underline{\widehat{U}}^{n+1} = \underline{\widehat{U}}^{n} + \Delta \underline{t}^{local} \Lambda^{-1}(V^T V)^{-1} V^T S^T R(SV\underline{\widehat{U}}^{n}) \ . \tag{2.15}$$

Here $\Delta \underline{t}^{local}$ denotes a vector of time steps, one for each node that is maximized for local stability of the time integration. This local time stepping scheme greatly reduces the number of time steps required to reach steady state.

*2.4.2.2 Implicit Method.* Implicit steady analysis is accomplished using a chord method implementation (21). The chord method is essentially Newton's method with numerically approximated Jacobians. The Newton iterations (for iteration $\nu$ to $\nu+1$)

at reduced order are,

$$
\begin{aligned}
\widehat{R}(\widehat{\underline{U}^{\nu}}) &= \Lambda^{-1}(V^T V)^{-1} V^T S^T R(\underline{U}^{\nu}) \\
\Delta\widehat{\underline{U}} &= \left[\frac{d\widehat{R}(\widehat{\underline{U}})}{d\widehat{\underline{U}}}\right]^{-1} \cdot \widehat{R}(\widehat{\underline{U}^{\nu}}) \\
\widehat{\underline{U}}^{\nu+1} &= \widehat{\underline{U}}^{\nu} + \Delta\widehat{\underline{U}} \\
\underline{U}^{\nu+1} &= \underline{U}_0 + \Psi \cdot \widehat{\underline{U}}^{\nu+1} \ .
\end{aligned}
$$

Notice that the full order function call is required for each Newton iteration. The reduced order Jacobian is not updated between Newton iterations to reduce computational cost. The reduced order Jacobian can be quickly identified using the full order solver,

$$
\begin{aligned}
\frac{d\widehat{R}(\widehat{\underline{U}})}{d\widehat{\underline{U}}} &= \frac{d}{d\widehat{\underline{U}}}\left[\Lambda^{-1}(V^T V)^{-1} V^T S^T R(\underline{U})\right] \\
\frac{d\widehat{R}(\widehat{\underline{U}})}{d\widehat{\underline{U}}} &= \Lambda^{-1}(V^T V)^{-1} V^T S^T \frac{dR(\underline{U}))}{d\widehat{\underline{U}}} \ .
\end{aligned} \tag{2.16}
$$

Next the chain rule is applied,

$$
\begin{aligned}
\frac{dR(\underline{U})}{d\widehat{\underline{U}}} &= \frac{dR(\underline{U})}{d\underline{U}}\frac{d\underline{U}}{d\widehat{\underline{U}}} \\
\underline{U} &= \underline{U}_{IC} + \Psi\widehat{\underline{U}} \\
\frac{d\underline{U}}{d\widehat{\underline{U}}} &= 0 + \frac{d}{d\widehat{\underline{U}}}\left[\Psi\widehat{\underline{U}}\right] \\
\frac{d\underline{U}}{d\widehat{\underline{U}}} &= 0 + \Psi\frac{d\widehat{\underline{U}}}{d\widehat{\underline{U}}} \\
\frac{d\underline{U}}{d\widehat{\underline{U}}} &= \Psi \\
\frac{dR(\underline{U}))}{d\widehat{\underline{U}}} &= \frac{dR(\underline{U})}{d\underline{U}}\Psi \ ,
\end{aligned}
$$

and remembering that $\Psi = SV$ yields the final relationship,

$$
\frac{d\widehat{R}(\widehat{\underline{U}})}{d\widehat{\underline{U}}} = \Lambda^{-1}(V^T V)^{-1} V^T S^T \frac{dR(\underline{U}))}{d\underline{U}} SV \ . \tag{2.17}
$$

For computational purposes it was much more efficient to obtain the reduced order Jacobian numerically from equation (2.16). The Jacobian was obtained using a central difference approximation. Equation (2.16) only required $2M$ function calls to the full order solver ($M$ being the number of reduced order variables) in comparison to $2N$ function calls ($N$ being the total number of DOFs) required when using equation (2.17). Once obtained, the reduced order Jacobian did not need to be recalculated between Newton iterations for domain sections exhibiting linear flow field behavior.

*2.4.3 Time Accurate Formulation.* Time accurate formulation approximates the flow field evolution from initial condition to fully developed flow. Convergence enhancements which can make steady solvers very efficient are not useful for the unsteady case. This makes time accurate formulation the more computationally cumbersome implementation. Consequently, the potential speed-up from the use of POD/ROM is most needed for unsteady analysis.

*2.4.3.1 Explicit Method.* Time accurate explicit integration for the reduced order solver is accomplished using the first order accurate, forward Euler, time integration,

$$\underline{\widehat{U}}^{n+1} = \underline{\widehat{U}}^{n} + \Delta t \Lambda^{-1} (V^T V)^{-1} V^T S^T R(SV\underline{\hat{U}}^{n}) \ . \tag{2.18}$$

Here $\Delta t$ denotes a global time step which is generally small to preserve stability for the smallest node spacing in the domain. This time accurate integration scheme is computationally burdensome, however the CFL condition for stability is increased by the subspace projection method (3). This simple integration scheme has been used to explore the use of POD for a variety of applications (3, 21, 9).

*2.4.3.2 Implicit Method.* The chord method is used to implicitly obtain time accurate solutions. Consider the implicit time integration function $F$:

$$F(\underline{U}^{n+1}) = \underline{U}^{n+1} - \underline{U}^{n} - \Delta t R(\underline{U}^{n+1}) \ . \tag{2.19}$$

The value of $\underline{U}^{n+1}$ that results in $F(\underline{U}^{n+1}) = 0$ is the solution for the flow field at time $t_n + \Delta t$ from $\underline{U}^n$. The Newton iterations to obtain this solution (for sub-iterate $\nu$ to $\nu+1$) at reduced order are shown below.

$$
\begin{aligned}
\widehat{F}(\widehat{\underline{U}^{\nu}}) &= \Lambda^{-1}(V^T V)^{-1} V^T S^T F(\underline{U}^{\nu}) \\
\Delta\widehat{\underline{U}} &= \left[\frac{d\widehat{F}(\widehat{\underline{U}})}{d\widehat{\underline{U}}}\right]^{-1} \widehat{F}(\widehat{\underline{U}^{\nu}}) \\
\widehat{\underline{U}}^{\nu+1} &= \widehat{\underline{U}}^{\nu} + \Delta\widehat{\underline{U}} \\
\underline{U}^{\nu+1} &= \underline{U}_0 + \Psi\widehat{\underline{U}}^{\nu+1}
\end{aligned}
$$

As in the steady formulation, the full system function call is required for each Newton iteration. The reduced order Jacobian can be quickly identified using equation (2.16),

$$
\begin{aligned}
\frac{d}{d\widehat{\underline{U}}} F(\underline{U}) &= \frac{d}{d\widehat{\underline{U}}} \left[\underline{U} - \underline{U}_{last} - \Delta t R(\underline{U})\right] \\
\Psi\widehat{\underline{U}} &= \underline{U} - \underline{U}_0 \\
\frac{dF(\underline{U})}{d\widehat{\underline{U}}} &= \Psi - \Delta t \frac{dR(\underline{U})}{d\widehat{\underline{U}}}
\end{aligned} \tag{2.20}
$$

and,

$$
\begin{aligned}
\frac{d\widehat{F}(\widehat{\underline{U}})}{d\widehat{\underline{U}}} &= \frac{d}{d\widehat{\underline{U}}} \left[\Lambda^{-1}(V^T V)^{-1} V^T S^T F(\underline{U})\right] \\
\frac{d\widehat{F}(\widehat{\underline{U}})}{d\widehat{\underline{U}}} &= \Lambda^{-1}(V^T V)^{-1} V^T S^T \frac{dF(\underline{U}))}{d\widehat{\underline{U}}} \ .
\end{aligned} \tag{2.21}
$$

### *2.5 Domain Coupling with Optimization*

Under certain conditions, constraints must be introduced into the reduced order solver to enforce smoothness on the internal boundary between adjacent domains. This is necessary when the domain coupling provided by the use of ghost cells and the full system dynamics is not sufficient to ensure continuity. Consider the case where the internal boundary between two adjacent regions overlap, such that the outer boundary of one region extrudes beyond the inner boundary of the adjacent region by a few cells. When computing solutions for each individual domain at reduced order, the flow field from both domains

should match in the overlapping portion of the flow field. When this does not occur, a constraint can be included to force the flow field sections to match in the overlap region. A constrained optimization method is developed below, which is later applied to steady, quasi-steady, and unsteady analysis of the POD/ROM/DD.

Consider $\underline{U}_{S1}$ and $\underline{U}_{S2}$ as adjacent domains with a shared internal boundary $\Gamma$. The domains overlap on the internal boundary, and the sections combine to form the fluid vector $\underline{U}$,

$$\underline{U} = \begin{bmatrix} \underline{U}_{S1} \\ \underline{U}_{S2} \end{bmatrix} . \tag{2.22}$$

Two constraint options are considered. The first constraint $C^1$ can be written as

$$C^1(\underline{U}) = \sum_{i \in \Gamma} \underline{U}(i)_{S1} - \underline{U}(i)_{S2} = 0. \tag{2.23}$$

This is similar to requiring the mean difference be zero on the boundary, and $C^1$ approximates the $L^1$ norm. Notice that the absolute value is not included. This weakens the constraint since negative errors can be cancelled by positive errors of equal magnitude; however, neglecting the absolute value improves computational performance. The solver formulation will use the first and second derivatives of the constraints (look ahead to equation (2.31). When the $C^1$ constraints are used without the absolute values, the first and second derivatives are a constant function and zero respectively. This allows the Jacobian to be pre-computed prior to solver integration, significantly reducing compute time. This is discussed more thoroughly in chapter 5. If an absolute value is introduced, the first and second derivatives are both functions of the flow variables, and pre-computing the Jacobian is not possible. Instead of using the absolute value on the $C^1$ constraint, the squared difference is constrained to zero which is written as

$$C^2(\underline{U}) = \sum_{i \in \Gamma} (\underline{U}(i)_{S1} - \underline{U}(i)_{S2})^2 = 0. \tag{2.24}$$

This is similar to requiring the variance be zero on the boundary, and $C^2$ approximates the $L^2$ norm. Like including the absolute value on the $C^1$ constraint, the use of the $C^2$

expression provides a much stronger constraint, and the computational load is increased because the first and second derivatives are functions of flow variables. However, the derivative expressions are much simpler with $C^2$ constraints than the case of absolute value on the $C^1$ constraints, which is why this formulation is preferred.

For the time accurate case, a functional $\ell(\underline{U})$ is defined such that,

$$\frac{d\ell(\underline{U})}{d\underline{U}} = F(\underline{U}) \ , \tag{2.25}$$

where $F$ comes from equation (2.19). For steady analysis, $\ell(\underline{U})$ is defined such that $\frac{d\ell(\underline{U})}{d\underline{U}} = R(\underline{U})$ where $R$ is the nonlinear, spatial operator from the full system dynamics. In either case, solving for the critical values of $\ell(\underline{U})$ is equivalent to solving either $F(\underline{U}) = 0$, or $R(\underline{U}) = 0$, i.e. finding the flow solution of interest. Either $C^1$ or $C^2$ constraints are introduced to force the flow field from both sections to match in the overlapping portion of their respective domains. In some cases a series of constraints is used. The constraints are themselves functionals that produce scalar outcomes. These scalar outcomes tend to zero when the flow variables for the overlapping sections approach equality. For overlapping fluid variables that are not identical, the constraint functionals produce a small scalar residual.

The following development is for a single constraint, $C(\underline{U})$. Lagrange constrained optimization minimizes $\ell(\underline{U})$ subject to the constraint through the use of a Lagrange multiplier, $\lambda$, introduced as an additional degree of freedom. Lagrange-constrained optimization (51) modifies $\ell(\underline{U})$ by adding the linear constraint to form the function $Q(\underline{y})$:

$$Q(\underline{y}) = \ell(\underline{U}) + \lambda \ C(\underline{U}) \ . \tag{2.26}$$

The solution vector is augmented to include $\lambda$,

$$\underline{y} = \begin{bmatrix} \underline{U}_{S1} \\ \underline{U}_{S2} \\ \lambda \end{bmatrix} = \begin{bmatrix} \underline{U} \\ \lambda \end{bmatrix} . \tag{2.27}$$

The critical values of $Q$ are the values of $\underline{U}$ and $\lambda$ such that

$$G(\underline{y}) = \frac{dQ(\underline{y})}{d\underline{y}} = \begin{bmatrix} F(\underline{U}) + \lambda \frac{dC(\underline{U})}{d\underline{U}} \\ C(\underline{U}) \end{bmatrix} = [0]. \tag{2.28}$$

As long as the constraints are linearly independent, and the fluid problem has a unique solution, the critical value will be the unique minimizing function for $Q$ (51). The reduced order mapping includes the Lagrange multiplier:

$$\underline{y} = \begin{bmatrix} \Psi_1 & [0] & 0 \\ [0] & \Psi_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{\widehat{U}}_{S1} \\ \underline{\widehat{U}}_{S2} \\ \lambda \end{bmatrix} \tag{2.29}$$

which is rewritten as,

$$\underline{y} = \Psi_\lambda \underline{\widehat{y}} \ . \tag{2.30}$$

Newton iterations are used to solve the reduced order system for $\widehat{y}$. The flow field is obtained by expanding $\underline{\widehat{U}}_{S1}$, and $\underline{\widehat{U}}_{S2}$ with reduced order mappings for each section, after which $\lambda$ is discarded. The reduced order Jacobian comes from the full order Jacobian,

$$\frac{dG(\underline{y})}{d\underline{y}} = \begin{bmatrix} \frac{dF(\underline{U})}{d\underline{U}} + \lambda \frac{d^2C(\underline{U})}{d\underline{U}^2} & \frac{dC(\underline{U})}{d\underline{U}} \\ \frac{dC(\underline{U})}{d\underline{U}} & 0 \end{bmatrix} \tag{2.31}$$

$$\frac{d\widehat{G}(\underline{\widehat{y}})}{d\underline{\widehat{y}}} = \left(\Psi_\lambda^T \Psi_\lambda\right)^{-1} \Psi_\lambda^T \frac{dG(\underline{y})}{d\underline{y}} \Psi_\lambda \ . \tag{2.32}$$

### *2.6 Summary*

The applications that follow all employ the theory contained in this chapter. The discussion of optimization-based reduced order solvers was general, and additional problem specific details are addressed in the formulation sections of subsequent chapters. In the next chapter, the use of POD/ROM with moving shocks is explored in the most simple of applications: a quasi 1-D nozzle flow.

# III. Analysis of a One-Dimensional Nozzle Flow

## 3.1 Introduction

This chapter addresses the application of POD/ROM to a 1-D high-speed flow field with a strong, moving shock. The challenge in using POD for high-speed flow fields is presented by the presence of moving shock waves in such flows. Current techniques for generating POD/ROMs, as used for low speed analysis, do not generate useful ROMs for the high-speed case with moving shocks. POD/ROMs generated in this manner required too much data to be useful, and they cannot be relied upon to track changes in shock location as the boundary conditions or flow parameters are changed. Instead, the methods from the previous chapter are applied to accurately treat the moving shock. Domain decomposition is used to isolate the shock. A reduced order model for each region is developed independently, and the solution for the entire domain is formed by linking the boundaries of each region to systematically solve for the flow field.

## 3.2 Quasi 1-D Nozzle

A quasi 1-D divergent nozzle was used for the model problem as shown in Figure 3.1. A quasi 1-D nozzle accounts for the area change of the nozzle by inserting a forcing function into the 1-D Euler equations. The resulting solution is interpreted as the flow field at the centerline of the nozzle. A numerical solver based on Roe's scheme for this simplified flow field was used as the full system from which various POD/ROMs were generated and compared. The numerical solver for this application is described in detail because it was slightly different from the finite volume solver described in the previous chapter.

For unsteady one-dimensional flow in a duct of variable (but known) cross-sectional area, the Euler equations (in conservation form) reduce to the equation set given below (52).

$$U_t + E_x = Z \tag{3.1}$$

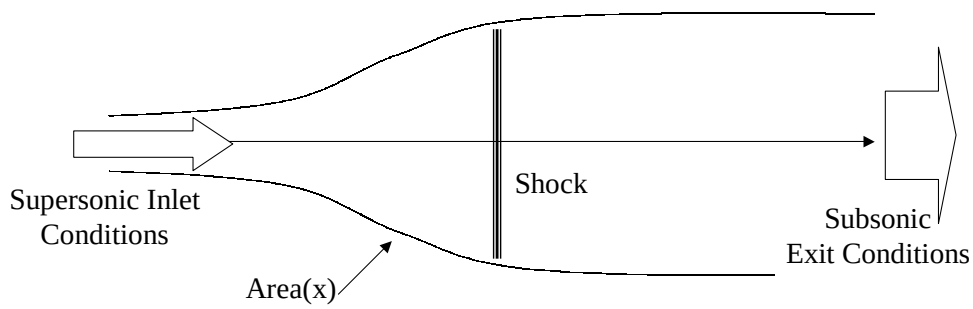



Figure 3.1 Quasi 1-D Nozzle

$$U = \begin{bmatrix} \rho A \\ \rho u A \\ \rho E_T A \end{bmatrix} \tag{3.2}$$

$$E = \begin{bmatrix} \rho u A \\ (\rho u^2 + P)A \\ (\rho E_T + P)uA \end{bmatrix} \tag{3.3}$$

$$Z = \begin{bmatrix} 0 \\ -P\frac{dA}{dx} \\ 0 \end{bmatrix} \tag{3.4}$$

The following profile was used for nozzle area $A(x)$,

$$A(x) = 1.398 + 0.347 \tanh(0.8x - 4) \ . \tag{3.5}$$

Roe's scheme (45) was used to explicitly solve the entire flow field by discretizing the spatial domain into a fine grid, and time marching from a known initial condition via small time steps. The inlet was at $x = 0$, and the inlet boundary conditions (non-dimensional) were $\rho_{in} = 0.5020$, $u_{in} = 1.299 \Rightarrow (\rho u)_{in} = 0.6521$, and $E_{Tin} = 1.3758$. The characteristic length $L$ was 1. The nozzle exit was at $x = 10$, and the exit boundary condition was

$\rho_{out} = 0.776$. The fluid modeled was air with $\gamma$ equal 1.4. For these conditions, a shock forms in the nozzle at $x = 4.816$ (52).

### *3.3 Shocks with POD*

For the initial analysis, POD/ROMs were generated without domain decomposition, and without any special treatment for the shock. A POD/ROM was constructed using 100 snapshots taken from time integration of the full system for the quasi 1-D nozzle problem at $1s$ intervals. For the chosen initial conditions, the solver very closely reached steady state by $100s$. For this experiment, an evenly spaced 63 grid point mesh was used.

The chosen initial condition ($\rho(x,0)$, $(\rho u)(x,0)$, $E_T(x,0)$) was simply a linear function of $x$ connecting the inlet and exit boundary conditions with reasonably close estimates for the final values of the unspecified inlet and exit flow variables. For example, $\rho$ was specified only at the nozzle exit. The initial condition for density was a line connecting this specified exit value with an approximation of the steady state value of $\rho$ at the inlet. As repeated integrations were applied to this initial condition, the inlet values $\rho(0,t)$ were unconstrained, but the exit value $\rho(x_f,t)$ stayed fixed. The resulting flow field took a wide range of functional forms; from smooth linear functions to discontinuous ones.

The flow field transient converged to steady state after a $100s$ time integration. For brevity, density is the lone fluid variable considered. The first $7s$ are shown in the first frame of Figure 3.2. Each line is a snapshot of the full order density solution at $1s$ intervals in time. Notice that the shock wave formed in the first $7s$, and it formed about 1 distance unit down stream ($x \approx 5.8$) of the steady state shock location ($x = 4.816$). The next $23s$ are shown in the middle frame of Figure 3.2. During this time period, the shock wave moved up stream to $x \approx 4.3$, which was well up stream of the final steady state shock location. The final portion is depicted in the final frame of Figure 3.2. The shock locations underwent smaller oscillations and finally settled at the steady state value of $x = 4.816$.

The POD/ROM was able to reproduce these dynamics, including the shock motion. An exceptionally good representation was obtained using 58 total modes generated from the set of 100 snapshots (20 modes for $\rho$, 19 modes for $\rho u$, and 19 modes for $E_T$). Every

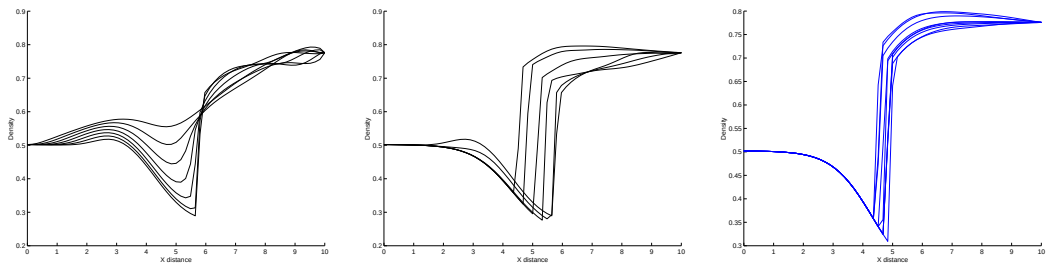



Figure 3.2 Density Function $t = 0 \rightarrow 7s$, $t = 7 \rightarrow 30s$, and $t = 30 \rightarrow 100s$

metric considered indicated a close adherence to the true spatial solution for each fluid variable at each point in time. The largest $L_{max}$ for any fluid variable was only 1.15%. This indicates that there is no place or time that the reduced order model does not closely match the true solution. The largest $L^2_{err}$ for any fluid variable at any time was only 0.404%, indicating that spatial features of the solution were also closely tracked for all time. POD/ROM provided exceptional solution accuracy while reducing the order from 189 degrees of freedom to 58.

Depending on the application, errors of 5% on the average might be acceptable. This was achieved with about 45 modes (15 modes for $\rho$, 15 modes for $\rho u$, and 15 modes for $E_T$). From 36 to 28 modes, the reduced order model generated a reasonable approximation of the full order solution for most of the trajectory. Reduced order models below 25 modes were of questionable use. Also, the POD/ROM became less accurate when too many modes were included. Instabilities arose that prevented the reduced order model from reaching steady state when too many modes were included.

The columns of $\Psi$ were the basis functions (or modes) that the reduced order model used to linearly reconstruct the time evolution of the flow. These modes were ordered by the size of their eigenvalues, with the modes contributing the most geometrically to the snapshot data occupying the earliest columns of $\Psi$. The initial function $\rho(x, 0)$, a linear initial density function, was subtracted from each snapshot in this implementation. Thus, the modes represented the residual spatial dynamics of the time evolution. Time integration of the reduced order model sequentially updated the solution with a POD based residual. The largest 10 modes are shown below in Figure 3.3. Notice that there were two distinct regions in the spatial modes for density. Between $x \approx 4.5$ and $x \approx 6$

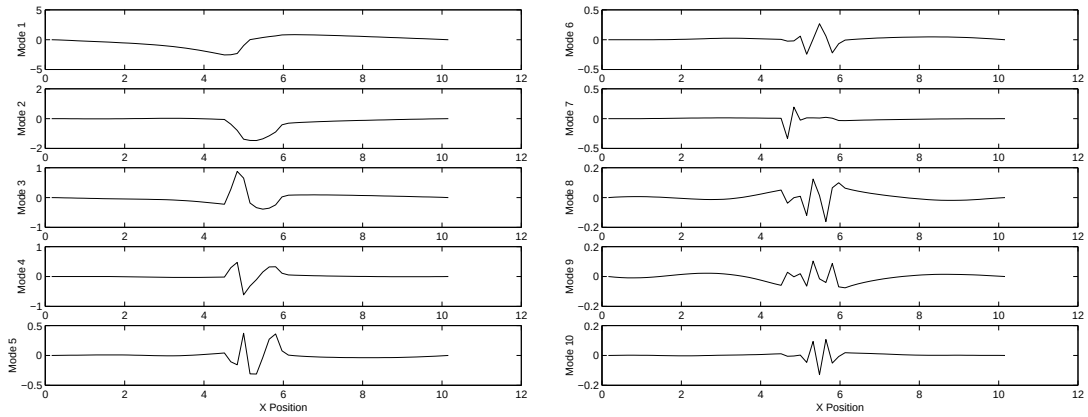



Figure 3.3 Density Modes $100s$ Case $1 \rightarrow 10$

sharp gradients and jagged transitions were prominent. Observing the y-axis scaling, the magnitude of these dynamics decreased slowly from modes 1 to 10. In contrast, outside this region a great majority of the non-zero modal contribution was contained in the first mode. Modes 8 and 9 also had a noticeable contribution to make. The major contribution to the need for a large number of modes in the POD/ROM was the region of the flow field containing the shock. Outside this range a 4 or 5 mode solution would probably have sufficed.

Initially it was unclear whether the existence of a shock in the flow field intrinsically required more modes, much like the need for a large number of Fourier sine modes to recreate a step function. As is evident in Figure 3.3, POD identifies discontinuous modes to track discontinuous functional behavior. Many modes were required in the region between $x \approx 4.5$ and $x \approx 6$ because of the motion of the shock wave. Few modes were required outside this region because there was not a great disparity between the initial and final states. It was the dynamics of the spatial solution in time that mattered to the POD/ROM, not the presence of spatial discontinuities in the solution.

To illustrate that relatively stationary shocks do not require a large number of modes, a POD/ROM was developed to replicate the last $30s$ of the $100s$ time integration. This comprised a sequence of functions that is nearly converged to the steady state value. For this case there was no formation of a shock from a smooth initial condition. Also, the movement of the shock was limited to two grid points. Even though the flow field

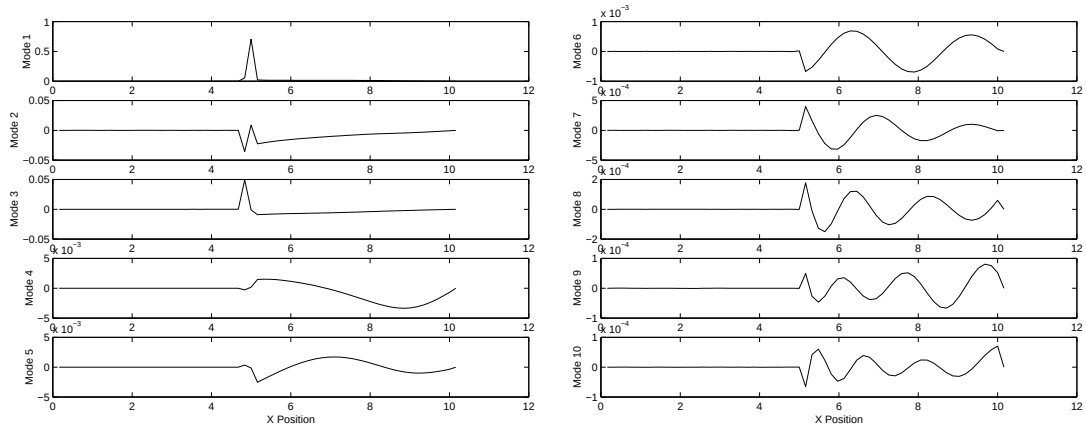



Figure 3.4 Density Modes 30$s$ Case 1 $\rightarrow$ 10

contained a spatial discontinuity, it should be possible to generate an accurate POD/ROM with much fewer modes than necessary for the 100$s$ case. A POD/ROM was generated from a set of 30 evenly spaced snapshots.

An exceptionally good representation of the full order trajectory was obtained using only 9 total modes (3 modes for $\rho$, 3 modes for $\rho u$, and 3 modes for $E_T$). The largest $L_{max}$ for any fluid variable was only 1.4% indicating that there was no place or time that the reduced order model did not closely match the true solution. The largest $L^2_{err}$ was only 0.707%, indicating that spatial features of the solution were also closely tracked for all time. Also, a very useful reduced order model (average errors $< 5\%$) was generated from even fewer modes (2 modes for $\rho$, 3 modes for $\rho u$, and 2 modes for $E_T$ for a total of 7 degrees of freedom). This is in stark contrast to the 58 modes required for exceptionally accurate reproduction of the 100$s$ case. The fact that both cases involved flow fields with discontinuities shows that the number of required modes is not intrinsically increased for shocked flows. Consider the density modes generated for the 30$s$ case shown in Figure 3.4. Unlike the modes for the 100$s$ case, the region of sharp gradients and jagged edges is very small ($x \approx 4.75 \rightarrow 5.1$) and the amplitude diminishes quickly for increasing mode numbers. By the fourth mode, the spatial contribution of the modes are greater in the smooth regions of the flow field than for the location of the shock. Clearly the presence of the discontinuity was readily recreated with only the first three modes. This was also the case for the smooth portions of the flow field, which was recreated with three modes as well.

Since the POD/ROM created discontinuous modes to represent discontinuous flow fields, a very few of these modes were adequate in representing the relatively stationary shock. Traditional spectral methods attempt to recreate discontinuous flow fields with smooth basis functions (complex exponentials, Chebyshev polynomials, and Legendre polynomials) (20). A great many modes are required to accurately recreate a discontinuous function from a set of smooth functions.

### *3.4 Robustness of POD/ROM*

POD modes are generated to match the geometry of all the functions that can be created by a linear combination of the set of snapshots. As shown above, replicating discontinuous behavior of functions sampled by the snapshots was no problem for POD. However, design iteration using POD/ROMs may require POD to produce results that wander outside the original set of candidate snapshots from which the POD/ROM was formed. This is because the set of snapshots is formed using outcomes of the full system, which are very expensive (computationally) to generate. The value of the reduced order model will be its ability to generate accurate flow solutions that have not been produced by the full system. For low-speed flows, flow structures inherent in POD modes are adequate for a wide range of full order solutions. As a result POD reduced order models are reasonably robust for low-speed incompressible flows (19). As will be shown, this is not the case for high-speed compressible flows with moving strong discontinuities. A function containing a discontinuity at one location cannot translate that discontinuity to another location under linear addition with another function, unless that second function has a discontinuity at the new location as well. Since POD is a linear technique, for POD to move a shock, both shock locations must reside in the original set of snapshots. It would seem this requires a full system outcome for every shock location of interest, which might defeat the purpose of the reduced order model.

To illustrate this condition, the POD/ROM from the $100s$ case was used to model a flow field slightly different from the flow field used to form it. Small changes in $\gamma$ were made to move the steady state shock location incrementally downstream. The steady state solution for density at several values of $\gamma$ is shown in Figure 3.5. These results were

obtained from the full system. Using the same linear initial condition described earlier,

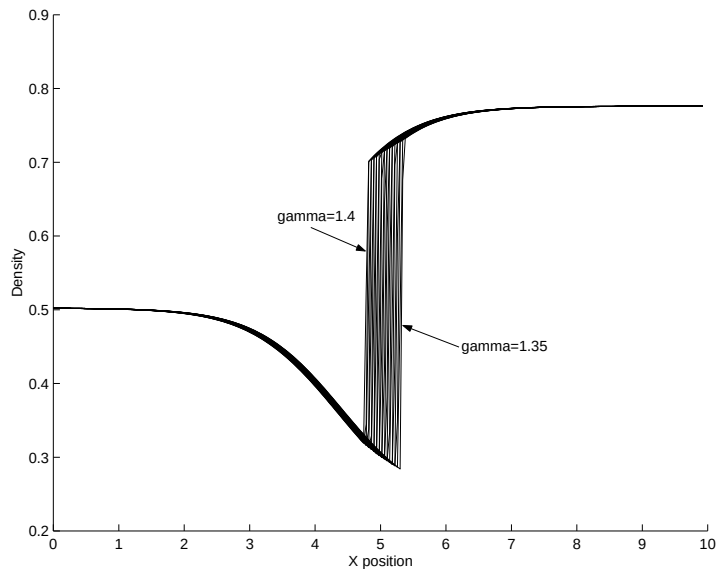



Figure 3.5 Shock Motion with Varying $\gamma$

the POD/ROM from the $100s$ case was time integrated with the new value of $\gamma$. The POD/ROM for the $100s$ case would not converge to a steady state solution for a drop in $\gamma$ beyond 1.398. For larger drops in $\gamma$, the POD/ROM time integrations went unstable during the transient period as the shock attempted to settle at the new downstream shock location. This occurred because the spatial region containing the shock motion captured in the snapshots at $\gamma = 1.4$ did not coincide with the range of shock motion as $\gamma$ was reduced. None of the modes in the POD/ROM contained a discontinuity outside the original range of shock motion, so the shock could not be formed at the proper location. As a result, the POD/ROM went unstable.

To further illustrate this lack of robustness, a POD/ROM for the $100s$ case with $\gamma = 1.4$ was created on a refined grid of 250 evenly spaced grid points. Previously, 100 snapshots at $1s$ intervals were adequate to produce a very accurate POD/ROM when the full order solution contained 63 evenly spaced grid points. However, with the grid refined to 250 grid points, no number or combination of modes from a set of 100 snapshots taken at $1s$ intervals would produce a working POD/ROM. In every instance the POD/ROM would go unstable during the time integrations as the shock was moving in the transient

period. This instability occurred because the shock motion during the transient period covered a great many more grid points when the grid was refined. Since the modes must contain a discontinuity for every shock location, many more than the 100 snapshots would be required for this POD/ROM to function.

These experiments confirmed that current techniques for generating POD/ROMs, as used for low-speed analysis, were inadequate for high speed analysis of flows with moving shocks. POD/ROMs generated this way required a great many snapshots; one for each grid point traversed by a shock wave. POD/ROMs generated in this fashion were not robust to parameter changes in the flow field. Essentially, this POD/ROM could only reproduce shock behavior obtained from expensive runs of a full system model. It could not extend the full order results to study new shock behavior.

### *3.5 Domain Decomposition*

Domain decomposition was introduced to isolate the region of the flow field containing the moving shock for special treatment. For the regions of the flow field not containing a shock, a POD ROM can be generated using the methods developed for low-speed flow problems. The region (or regions) of the flow field containing moving shocks was approached in two ways. The first method was to us the full order simulation. Even though the shocked region of the flow had no reduction in the number of DOFs, the use of POD/ROM over the non-shocked portions of the flow field still provided a significant reduction in the number of DOFs relative to the original problem. Another possibility involved developing a reduced order model for the shocked region of the domain decomposed flow field. POD was used as a tool to search for a modal representation that allowed for both order reduction and accurate shock movement. Since the isolated shock domain was a very small portion of the entire solution domain, snapshot collection and POD/ROM development for this region was more efficient, allowing the reduced order model to be generated without excessive expense.

For the quasi 1-D problem, the domain was decomposed into three regions as shown in Figure 3.6. Section I started at the nozzle inlet and ended prior to the shock, thus the flow was supersonic throughout section I. Section II began at the exit of section I and

ended after the shock, thus section II had a supersonic inlet and a subsonic exit. Section III consisted of the rest of the nozzle to the exit, and contained all subsonic flow. Section I and III were modeled with POD/ROMs. Both full order and reduced order approaches for section II were studied.

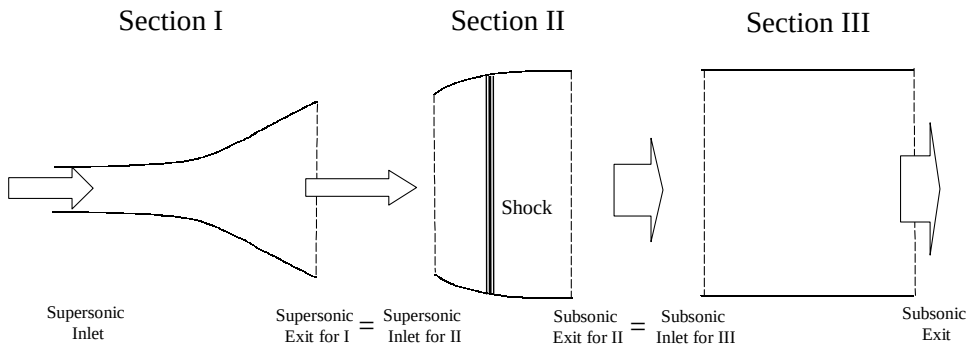



Figure 3.6 Domain Decomposition for the Quasi 1-D Nozzle

*3.5.1 Explicit Formulation and Results.* Initially, an explicit time integration of the three sections was used to produce a reduced order solution for the POD/ROM/DD with a full order section II. Time integration for the POD/ROMs of sections I and III was accomplished using equation (2.15). To facilitate the simultaneous integration of the three sections, the inlet and exit grid points for section II were coincident with the exit grid point for section I and the inlet grid point for section III. For the 100$s$ case with 63 grid points, the explicit POD/ROM/DD was able to accurately reproduce the full order solution with 81 DOFs, down from 189 DOFs for the full order nozzle. For the 30$s$ case, there was a smaller range of shock motion so the number of grid points modeled at full order in section II was reduced. For this case the number of evenly spaced grid points was expanded to 250. A very accurate solution was obtained using the following domain decomposition: section I contained 116 grid points, section II contained 9 grid points, and section III contained 125 grid points. The density solution at the time of largest $L^2_{err}$ is shown in Figure 3.7. This reduced order model used 8 modes per fluid variable in sections

I and III. Section II contained 9 grid points. The resulting reduced order model had 75 degrees of freedom, down from 750 for the full system.

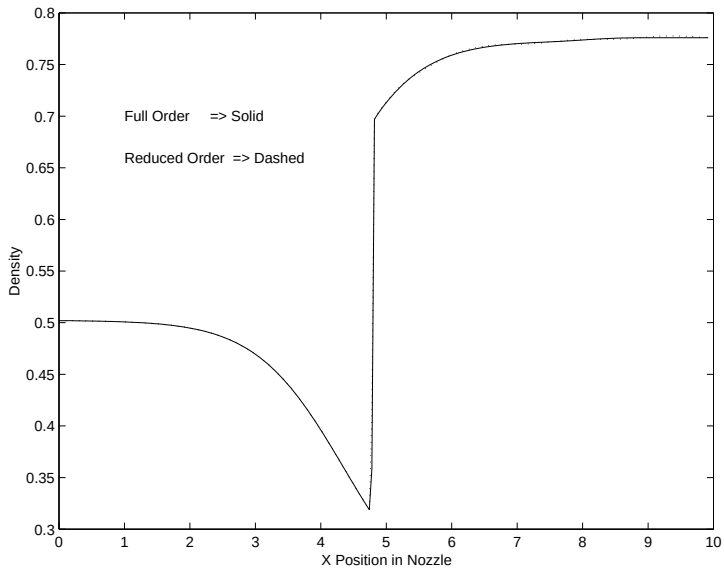



Figure 3.7 POD/ROM/DD Density Comparison with 250 Grid Points

*3.5.2 Quasi-Steady Formulation with Full Order Shock Region.* Next an implicit formulation was developed for the POD/ROM/DD with a full order shock region. Domain decomposition required a reordering of the full order solution vector $\underline{U}(t)$. Instead of ordering the members of $\underline{U}(t)$ by fluid variables distributed spatially across the entire nozzle, $\underline{U}_{DD}(t)$ stacked the fluid variables across each section. Each of the three sections was treated as an independent full order solution $\underline{U}_{S1}$, $\underline{U}_{S2}$, and $\underline{U}_{S3}$, so each had its own reduced order mapping $\underline{U}_{S1} = \Psi_1 \cdot \widehat{\underline{U}}_{S1}$, $\underline{U}_{S2} = \Psi_2 \cdot \widehat{\underline{U}}_{S2}$, and $\underline{U}_{S3} = \Psi_3 \cdot \widehat{\underline{U}}_{S3}$. Since section II was modeled at full order, $\widehat{\underline{U}}_{S2} = \underline{U}_{S2}$ and $\Psi_2$ was the identity mapping.

The domain decomposition reduced order mapping combines each independent section,

$$\underline{U}_{DD} = \begin{bmatrix} \underline{U}_{S1} \\ \underline{U}_{S2} \\ \underline{U}_{S3} \end{bmatrix} = \begin{bmatrix} \Psi_1 & [0] & [0] \\ [0] & I & [0] \\ [0] & [0] & \Psi_3 \end{bmatrix} \cdot \begin{bmatrix} \widehat{\underline{U}}_{S1} \\ \underline{U}_{S2} \\ \widehat{\underline{U}}_{S3} \end{bmatrix} ,$$

where,

$$\Psi_{DD} = \begin{bmatrix} \Psi_1 & [0] & [0] \\ [0] & I & [0] \\ [0] & [0] & \Psi_3 \end{bmatrix} .$$

The reduced order Jacobian was obtained from the full order Jacobian. The full order Jacobian $\frac{dF(\underline{U}_{DD})}{d\underline{U}_{DD}}$ was approximated numerically using central differences from evaluations of $F(\underline{U}_{DD})$ about $\underline{U}_{DD} \pm \Delta\underline{U}$. This non-Galerkin implicit method improves numerical efficiency because the reduced order Jacobian is more easily inverted . The Jacobian update is still accomplished at full order; however, this update does not need to be accomplished for every Newton iteration. Once the Newton iterations provide a solution sufficiently close, repeated iteration on the same Jacobian will converge quickly within the desired error threshold.

*3.5.3 Quasi-Steady Formulation with Reduced Order Shock Region.* Attempts were made to obtain a solution using a POD/ROM for section II generated from 25 snapshots of steady state flow solutions at values of $\gamma$ from $1.4 \rightarrow 1.35$ (shown in Figure 3.5). As $\gamma$ was dropped from 1.4, Newton iterations introduced a non-physical discontinuity at the intersection between sections II and III. Clearly a method to enforce smoothness at this intersection was required to use a POD/ROM for section II with the implicit solver.

This was accomplished using the constrained optimization technique outlined in Chapter 2. First, the domains in sections II and III were allowed to overlap by one, two or three grid points. Initially, it seemed increasing the amount of overlap would improve the solution because additional grid points contained information to equate first and second derivatives in the overlap region. For this reason, three grid points of overlap were used to generate the results that will be shown. However, the number of grid points in the overlap region (from one to three) had no effect on the solution. The same results could have been achieved with only one grid point of overlap. Very large amounts of overlap were not used because this would have polluted the modal representation with flow dynamics from the adjacent domain and nullified the domain decomposition.

An $L^1$ constraint was introduced to enforce smoothness at the intersection of sections II and III. A vector $T$ with the same dimensions of $\underline{U}_{DD}$ was defined such that $\underline{U}_{DD}^T T = 0$ when the flow variables for the overlapping sections had the same value. This was achieved by placing a 1 or $-1$ in each fluid variable location corresponding to the overlap in sections II and III respectively. Zeros were placed everywhere else in $T$. The dot product of $T$ with $\underline{U}_{DD}$ resulted in cancellation of the fluid variables when the overlapping portion of the solution domain was equivalent. If the overlapping fluid variables were not identical, the dot product produced a small scalar residual.

A Lagrange multiplier $\lambda$ was used to formulate the constrained optimization,

$$\begin{aligned} \underline{y} &= \begin{bmatrix} \underline{U}_{DD} \\ \lambda \end{bmatrix} \\ G(\underline{y}) &= \begin{bmatrix} F(\underline{U}_{DD}) + \lambda T \\ \underline{U}_{DD}^T T \end{bmatrix} . \end{aligned}$$

The modified Jacobian was,

$$\frac{dG(\underline{y})}{d\underline{y}} = \begin{bmatrix} \frac{dF(\underline{x}_{DD})}{d\underline{x}_{DD}} & T \\ T^T & 0 \end{bmatrix} .$$

The reduced order mapping was modified to include the reduced order mapping $\underline{U}_{S2} = \Psi_2 \cdot \widehat{\underline{U}}_{S2}$ and the Lagrange multiplier,

$$\underline{y} = \begin{bmatrix} \underline{U}_{S1} \\ \underline{U}_{S2} \\ \underline{U}_{S3} \\ \lambda \end{bmatrix} = \begin{bmatrix} \Psi_1 & [0] & [0] & 0 \\ [0] & \Psi_2 & [0] & 0 \\ [0] & [0] & \Psi_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \widehat{\underline{x}}_{S1} \\ \widehat{\underline{x}}_{S2} \\ \widehat{\underline{x}}_{S3} \\ \lambda \end{bmatrix} , \tag{3.6}$$

which was rewritten,

$$\underline{y} = \Psi_\lambda \cdot \widehat{\underline{y}} .$$

The constrained reduced order Jacobian was obtained from the full order constrained Jacobian,

$$\frac{d\widehat{G}(\underline{\widehat{y}})}{d\underline{\widehat{y}}} = \left(\Psi_\lambda^T \Psi_\lambda\right)^{-1} \Psi_\lambda^T \frac{dG(\underline{y})}{d\underline{y}} \Psi_\lambda \ .$$

### *3.6 Quasi-Steady Results*

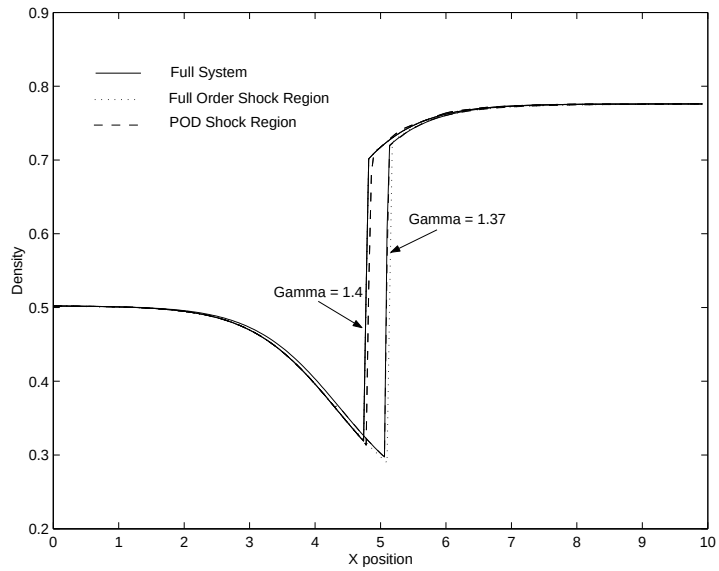



Figure 3.8 Shock Motion with Varying $\gamma$

The implicit methods were successfully used to obtain results for the POD/ROM/DD with shock regions modeled at both full order and with POD/ROM. The 250 grid point full order nozzle problem was decomposed into three sections: section I had 117 grid points, section II had 15 grid points plus 2 coincident grid points for a total of 17, and section III had 118 grid points. Two POD/ROM/DD models were analyzed. The first modeled section II at full order with 17 grid points. Section I used one mode per fluid variable, and section III used one mode for both density and momentum, and two modes for energy. The resulting POD/ROM/DD had 58 DOFs, down from the original 750. The second POD/ROM/DD used a POD/ROM for section II generated from the 25 snapshots of steady state flow solutions at values of gamma from $1.4 \rightarrow 1.35$ described earlier. An additional four grid points were added to section II as overlap with section III. The POD/ROMs for sections I and III were identical to that described above for the first case.

Upon examination of the eigenvalues associated with the modes for section II, significant energy (order $10^{-3}$) was found in the $16^{\text{th}}$ mode. Energy in the $17^{\text{th}}$ mode was order $10^{-7}$. Very good results were obtained using 16 modes per fluid variable. The resulting POD/ROM/DD had 55 DOFs, 7 for sections I and III, and 48 for section II.

Steady state solutions from both POD/ROM/DDs were obtained as $\gamma$ was varied from 1.4 down to 1.37. Note that the POD/ROMs for sections I and III were trained at $\gamma = 1.4$. The density results shown in Figure 3.8 show that both POD/ROM/DD's tracked the quasi-steady shock motion very closely. The POD/ROM/DD with the full order section

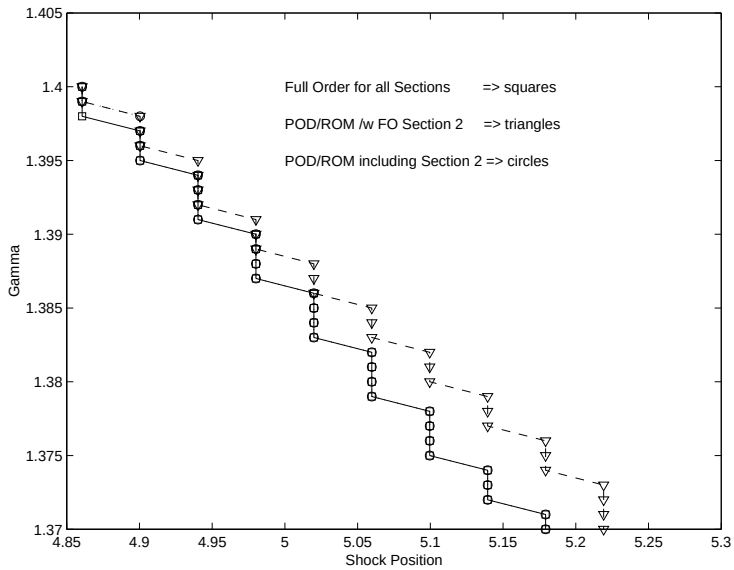
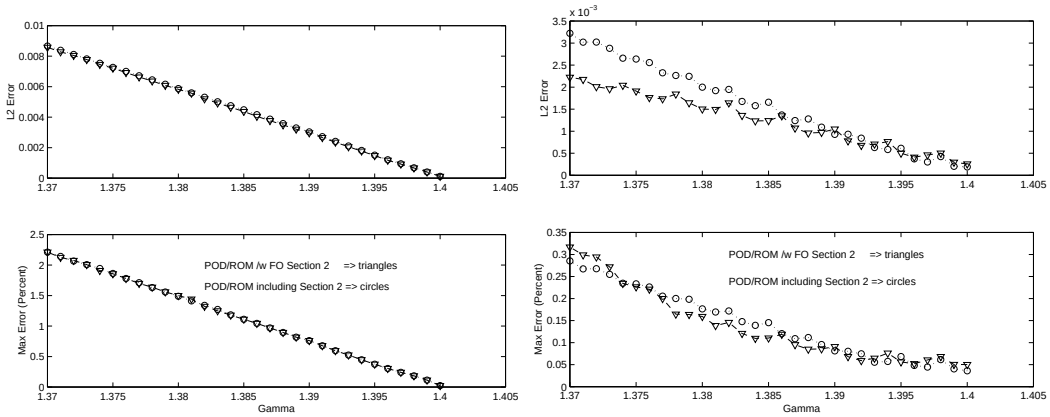



Figure 3.9 POD/ROM/DD Shock Location in the Nozzle

II misplaced the shock location one grid point down stream as $\gamma$ was reduced, while the POD/ROM/DD with a POD/ROM for section II tracked shock location nearly exactly to the full order result. This is depicted in Figure 3.9. Both POD/ROM/DDs accurately recreated the flow solution before and after the shock. Errors in density after the shock for both POD/ROM/DDs are depicted in Figure 3.10.b. Notice that errors grew as gamma was steadily decreased, which occurred because the POD/ROMs for sections I and III were trained at $\gamma = 1.4$. Even with the small error growth, the error in density was less than 1% for either the $L_2$ or the maximum norms.




( a. Before Shock) (b. After Shock)

Figure 3.10 POD/ROM/DD Density Error

Errors in density before the shock for both POD/ROM/DDs are depicted in Figure 3.10.a. Notice that error growth was greater for this portion of the nozzle. Also notice that both POD/ROM/DDs exhibited the same performance. For both models, a small discontinuity formed at the intersection of sections I and II as gamma was steadily decreased. This discontinuity slowly grew as $\gamma$ was decreased further. The growing discontinuity is reflected in the larger maximum error norm in Figure 3.10.b. The constrained implicit formulation did not include overlap between sections I and II, which allowed for the non-physical discontinuity to occur.

## *3.7 Conclusions*

The ability of POD/ROM to replicate the shocked flow solution from which it was formed was demonstrated. It was shown that POD/ROM can accurately recreate a flow solution with strong shocks, given that the appropriate data is present in the matrix of snapshots. If the shock was reasonably stationary, POD/ROM could accurately model the flow field with as few as three modes per fluid variable. It was also shown that the POD/ROM could very accurately reproduce flow dynamics involving a transition from smooth to shocked flow, including a shock with significant movement. Recreating this case required many more modes for accurate representation.

In high-speed design analysis, shock movement might force the POD/ROM to produce dynamics beyond the condition under which it was trained. Even small excursions of the shock location beyond the data result in failure of the POD/ROM to arrive at a solution. In these cases the POD/ROM goes unstable attempting to form the shock. The desire to use POD/ROM in design analysis of shocked flows motivates the domain decomposition shock capturing approach. The accuracy and order reduction of the domain decomposition approach was demonstrated for a quasi 1-D nozzle flow. The non-shocked regions of this flow field were modeled using POD/ROM. These POD/ROMs were trained using $\gamma = 1.4$. The shocked region of the flow field was modeled both by POD/ROM and by the full order CFD model adapted for this region. The accuracy of both models was examined for quasi-steady shock motion as $\gamma$ was varied from $1.4 \rightarrow 1.37$. Both cases produced accurate flow fields and shock motion. Flow field errors were less than 2%, and the shock movement was tracked within one grid point of the true shock location.

Both methods exhibited similar order reduction. The full order solution had 750 DOFs, the model with a full order shock region had 58 DOFs, and the model with a POD/ROM for the shock region had 55 DOFs. Sixteen modes per fluid variable were required for an accurate POD/ROM of the shocked region, resulting in the insignificant additional order reduction relative to the POD/ROM/DD with a full order shock region. Because of the computational expense of generating snapshots and the large number of modes required, there are no advantages motivating POD/ROM for the shocked region for this 1-D case. However, in 2-D and 3-D cases there might be a significant order reduction gained by constructing a POD/ROM for the regions containing shocks. In such situations, the computational expense of obtaining snapshots could be reduced by using a POD/ROM/DD with full order shock regions to generate snapshots, instead of the full order model. This is explored more fully in the next chapter.

# IV. Analysis of a 2-D Blunt Body Problem

## 4.1 Introduction

The 1-D analysis described in the previous chapter used the domain decomposition technique to accurately track a moving strong shock wave with POD/ROM. This chapter considers the application of the POD/ROM/DD approach to steady analysis of a 2-D problem. High-speed flow over a 2-D blunt nosed cone is the model problem for this study. Changes in inlet Mach number and angle of attack (AoA) results in quasi-steady motion of a strong shock wave off the nose of the blunt body. The POD/ROM/DD technique is applied to study the blunt body problem across this parameter space. The accuracy, order reduction, and computational savings are quantified.

## 4.2 2-D Blunt Body Problem

This analysis was conducted on an inviscid 2-D flow field over a blunt body. The structured finite volume numerical flow solver described in Chapter 2 was used as the full system model. The blunt body geometry was a 15 degree wedge with a circular nose, generated using two line segments and a circumscribed circle of radius 1.08501. The top surface was a line segment starting at $x = 0$, $y = 1$ and terminating at $x = 3$, $y = 2.0919107$. Similarly, the bottom surface was a line segment starting at $x = 0$, $y = -1$ and terminating at the $x = 3$, $y = -2.0919107$. A structured grid using 53 nodes along the solid surface and 35 nodes extending to the free stream was generated as shown in Figure 4.1.

## 4.3 Domain Decomposition

The solution domain in Figure 5.1 was broken into sections to facilitate the use of POD with strong moving shocks. Isolation of the standing strong bow shock was the goal of the domain decomposition. Changes in free-stream Mach and angle of attack produced quasi-steady shock motion. Domain decomposition was used to isolate a region of the computational mesh that contained the shock. A range of quasi-steady shock motion between Mach 2.7 and 4.2 with AoA between $-5$ and 5 degrees was chosen as the parameter

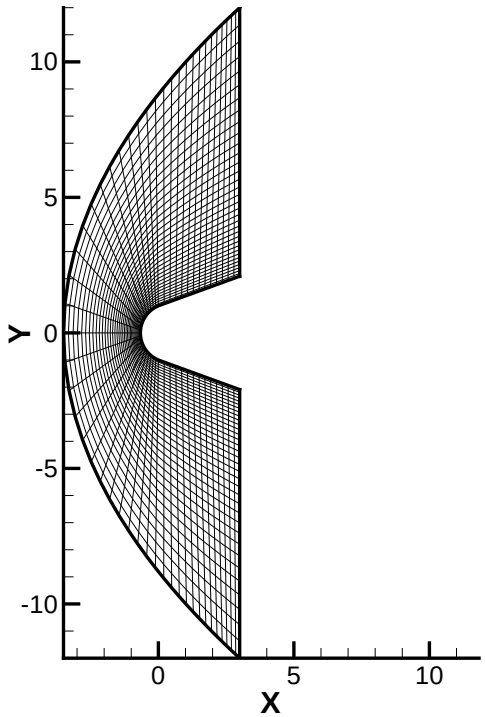



Figure 4.1 Blunt Body Geometry (non-dimensional)

space for this analysis. Steady state outcomes of the full system model showed that quasi-steady shock motion for this parameter space would always fall within the steady state shock locations for Mach 2.5 AoA 0 degrees, and Mach 5 AoA 0 degrees. Steady state pressure contours for these cases are shown in Figure 4.2.

A simple algorithm was used to search for the shock locations and allocate cells to sections. Section I contained the region of the flow field prior to the shock at Mach 2.5 and AoA 0 degrees, section III contained the region of the flow field behind the shock at Mach 5 and AoA 0 degrees, and section II comprised the remainder of the computational mesh. Section II was the region containing the shock motion. The allocation of cell centers to sections is shown in Figure 4.3. The computational mesh contained 1768 cell centers. The domain decompositions allocated 562 cell centers to section I, 408 cell centers to section II, and 798 cell centers to section III. For the regions of the flow field not containing a shock,

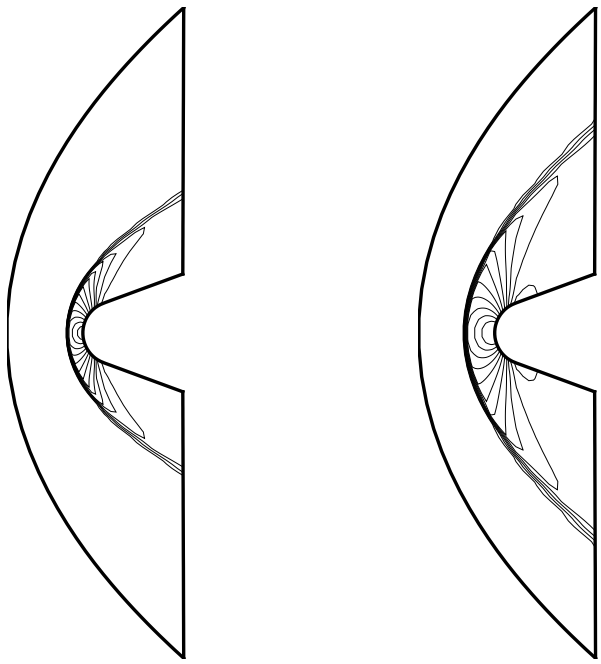


Figure 4.2 Steady State for Mach 5 (Left) and Mach 2.5 (Right)

a POD ROM was generated using the methods developed for low-speed flow problems. Both full order and reduced order approaches for section II were studied.

### *4.4 Steady Formulation*

The reduced order mapping from equation (2.12) is reproduced below. The reduced order vector $\widehat{U}$ represents the deviations of the full order flow field from the base flow $\underline{U}_0$, therefore the base flow must be added back when re-constituting the flow field.

$$\underline{U}(t) = \underline{U}_0 + \Psi\underline{\widehat{U}}(t)$$

The base flow was slug flow at the free stream conditions. An additional complication was introduced because the parameter space required changes in free stream values. Consider a

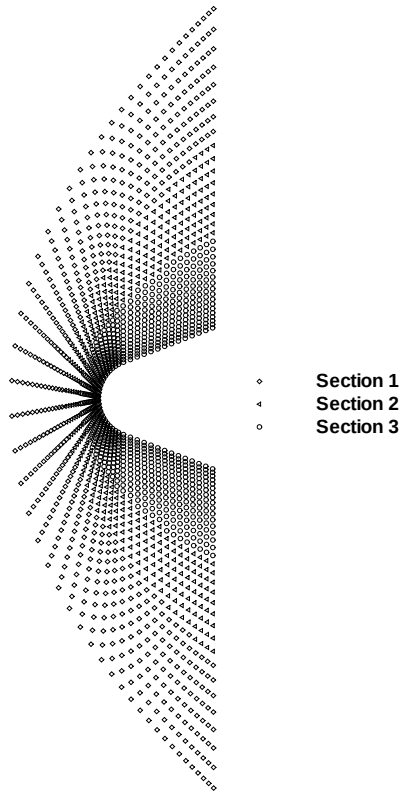



Figure 4.3 Domain Decomposition

steady state flow field for some value of Mach number and angle attack $\underline{U}_{IC}$, and the base flow for the same inlet conditions $\underline{U}_0^{IC}$. Now consider time integrating the POD/ROM/DD to steady state starting at $\underline{U}_{IC}$, but for a different value of Mach number and angle of attack and a new corresponding base flow $\underline{U}_0^{run}$. Equation (2.12) was modified as follows to accommodate this complication.

$$\begin{aligned} \underline{U}(t) - (\underline{U}_{IC} - \underline{U}_0^{IC}) &= \underline{U}_0^{run} + \Psi\widehat{\underline{U}}(t) \\ \Delta U_{IC} &= (\underline{U}_{IC} - \underline{U}_0^{IC}) \\ \underline{U}_{IC}^{trans} &= \Delta\underline{U}_{IC} + \underline{U}_0^{run} \end{aligned} \tag{4.1}$$

This is equivalent to,

$$\underline{U}(t) = \underline{U}_{IC}^{trans} + \Psi\widehat{\underline{U}}(t) \ . \tag{4.2}$$

The new initial condition $\underline{U}_{IC}^{trans}$ extracts the form of the initial flow field by subtracting off the base flow. This functional form ($\Delta\underline{U}_{IC}$) is translated to the new inflow conditions by adding the new base flow for the inflow conditions of interest.

Notice that equation (2.12) is returned when $\underline{U}_{IC} = \underline{U}_0$ in equation (4.1). The POD modes spatially compensate for the difference between the initial flow field $\underline{U}_{IC}^{trans}$ and the flow field at the time of interest $\underline{U}(t)$. If the initial condition is very near to steady state, then there is very little modal contribution to the reduced order solution. If the initial flow field is very far from steady state, a great deal of modal participation is required to define the flow field. This phenomenon appeared to be a uniqueness problem, since changing initial conditions could produce very different steady-state solutions (depending on the adequacy of the modes). However, the initial condition was part of the dynamics for the reduced order integration since $\underline{U}_{IC}^{trans} + \Psi\widehat{\underline{U}}(t)$ in equation (4.2) was the value for $\underline{U}(t)$ used in the flux calculation $R(\underline{U}(t))$. Therefore steady-state solutions varied with initial condition because changing the initial condition fundamentally changed the problem.

*4.4.1 Domain Decomposition Solver.* Since the domain was divided into three sections, the steady formulation for the POD/ROM/DD involved solving three smaller fluid problems that were linked by internal boundaries. External boundaries were handled with ghost cells as described earlier. Ghost cells for internal boundaries were filled with the corresponding values from the adjacent domain using data from the most recent time iteration. No domain overlap was necessary for the time integration implementation of the POD/ROM/DD with a full order shock region. Each of the three sections used its own vector of conserved flow field variables $\underline{U_{S1}}(t)$, $\underline{U_{S2}}(t)$, and $\underline{U_{S3}}(t)$. After each time integration step the entire flow field was re-constituted by combining and reordering the data from each section $\underline{U}(t) = \underline{U_{S1}}(t) \bigcup \underline{U_{S2}}(t) \bigcup \underline{U_{S3}}(t)$.

### *4.5 Results*

This section presents results for cases where the shock region is either modeled at full order, or treated with a POD/ROM. Additional formulation is included in the case using POD/ROM for the shock region to describe the use of constrained optimization.

*4.5.1 Full Order Shock Region Results.* A steady formulation involved explicit time integration and local time stepping for the full order shock region. The POD/ROM in section I was solved using a single time integration step to produce slug flow at the corresponding inlet conditions. The POD/ROM in section III was solved using the chord method. No domain overlap was used for this implementation.

Section I was solved first to provide the proper inlet conditions for section II. As the full order section II was time integrated, the POD/ROM for section III was periodically updated every 30 iterations using inlet boundary data from the exit of section II at the latest iteration. The value of 30 was determined through trail and error to provide the best convergence performance. This periodic update was necessary because a good portion of the internal boundary between sections II and III was subsonic. Information from section III had to flow across the boundary into section II to obtain the proper steady-state solution.

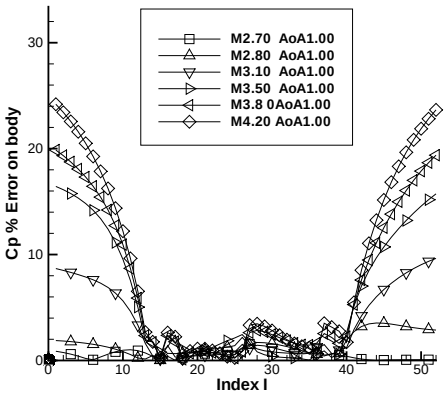
The POD/ROM/DD was trained using steady-state flow field solutions. Steady state solutions were obtained from the full order solver at AoAs of $-2$, $-1$, 0, 1, and 2 degrees, and Mach numbers of 2.5, 2.6, and 2.7. All combinations of these parameters produced 15 snapshots. For all snapshots, the base flow was slug flow at the corresponding inlet conditions. The POD/ROM/DD for steady analysis was obtained from this set of snapshots. The POD/ROM for section I used one Mode Per Fluid Variable (MPFV), while the POD/ROM for section III used two MPFV. With section II remaining at full order, the POD/ROM/DD totaled 1644 DOFs. This was a 4.3 to 1 reduction in DOFs relative to the full order problem. Steady state flow fields were generated for a variety of parameter values. The initial condition for the steady analysis was the steady-state flow field at Mach 2.7 and AoA 2 degrees, translated to the new inlet parameters.

The numerical efficiency of this implementation was evaluated by comparing wall-clock time for comparable runs of both the full system, and the POD/ROM/DD with two MPFV. A realistic expectation for computational improvement comes from the reduction in DOFs. For this problem, the reduction in DOFs was 4.3 to 1. A similar improvement in computational efficiency can be expected from a reasonable implementation of the reduced order model. Two bench-marking cases were considered. The first involved steady-state solution for inflow parameters of Mach 2.8 and AoA 0 degrees starting from steady-state flow at Mach 2.7 and AoA 2 degrees. The full order solver reached the convergence threshold after 1300 iterations of the full order solver. Wall-clock time was 37.73 seconds to obtain the full order solution. The computer specifications are provided in Table 4.1 The

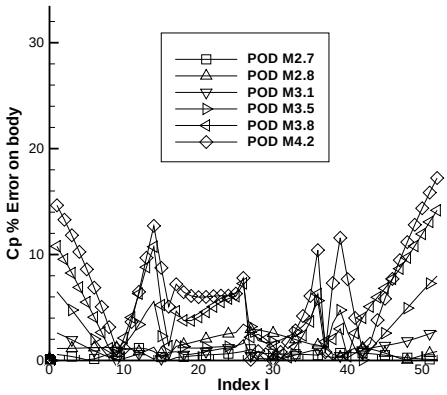
Table 4.1 PC Specifications

| | |
|---|---|
| Processor | 450 MHz Pentium 3 |
| Operating System | Windows 2000 |
| Memory | 128M |
| Fortran Compiler | Visual Fortran v6.1 |
| Optimizations | Optimized for speed |

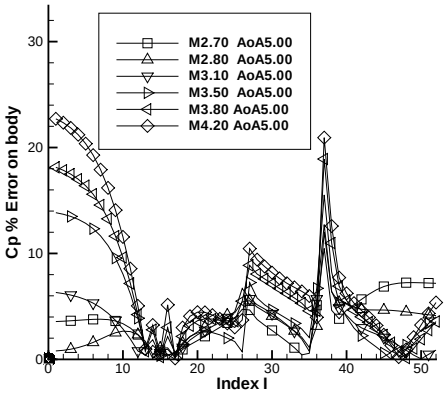
POD/ROM/DD converged in 900 iterations of the the full order section II in 8.51 seconds. The computational improvement for this case was 4.43 to 1. The POD/ROM/DD converged to steady state in fewer total iterations than the full order solver due to the chord method implementation in section III. The POD/ROM in section III converged to steady state very quickly and did not require any Jacobian updates throughout the solution procedure. This accounts for the wall-clock time savings being slightly better than expected when considering the reduction in DOFs. A second bench-marking case extended the POD/ROM/DD far outside the training window. This case required the computation of a steady-state solution for inflow parameters of Mach 4.2 and AoA 5 degrees starting from steady-state flow at Mach 2.7 and AoA 2 degrees. The full order solver reached the convergence threshold after 1900 iterations of the full order solver. 53.22 seconds wall-clock time was necessary to obtain the full order solution. The POD/ROM/DD converged in 1100 iterations of the the full order section II in 11.26 seconds. The computational improvement for this case was 4.726 to 1. Again the improvement in convergence history due to the POD/ROM in section III accounted for the better than expected time performance.



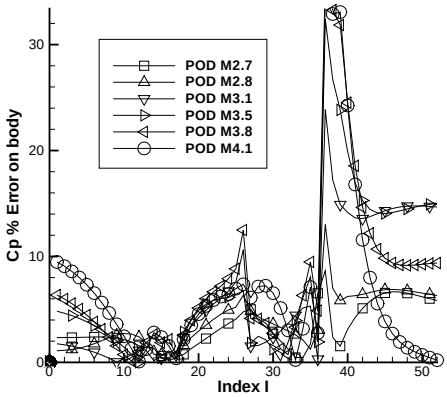

( a. AoA 1 Deg, 2 MPFV)




(b. AoA 1 Deg, 2 MPFV)




( c. AoA 5 Deg, 2 MPFV)

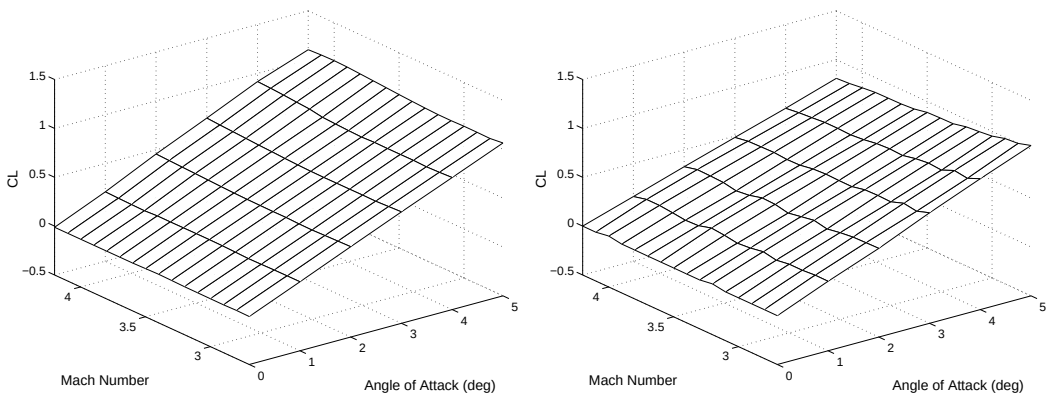



(d. AoA 5 Deg, 5 MPFV)

Figure 4.4 $C_P$ Percent Error on Blunt Body

The accuracy was examined across a parameter space well beyond that for which the POD/ROM in section III was trained. Errors in $C_P$ by index number on the body for a range of Mach numbers at AoA 1 degree are shown in Figure 4.4.a. The POD/ROM in section III was only trained to Mach 2.7, but $C_P$ errors on most of the body were small well beyond this Mach number. Between Mach 2.7 and 4.2 the shock moves 5 cells closer to the body. The flow between the nose and the shock is subsonic, requiring the sharing of information across the internal boundary between sections II and III to obtain a solution.The modal information collected during the training period was less adequate at higher Mach numbers. Notice in Figure 4.4.a that as Mach number increased, the

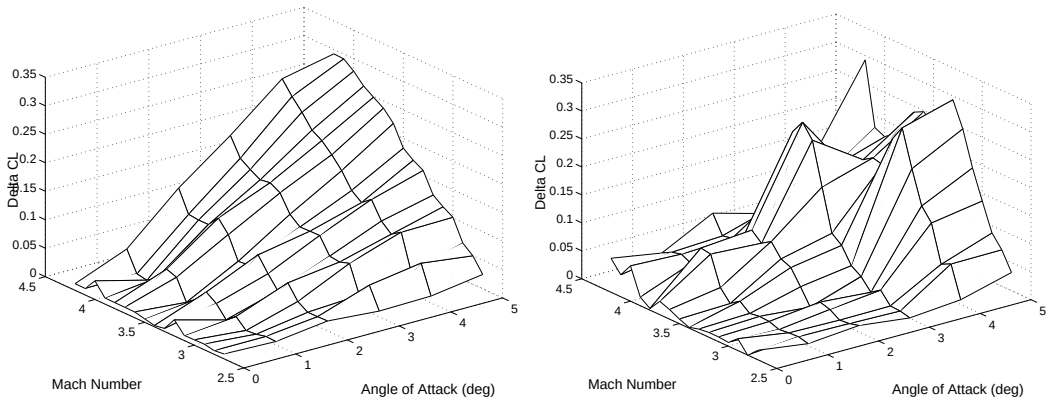
POD/ROM/DD has difficulty generating the proper pressures on the aft portions of the body. This performance was slightly improved by adding more modes. The same plot for a POD/ROM/DD with 5 MPFV instead of 2 is shown in Figure 4.4.b. At an AoA of 5 degrees and using 2 MPFV, the POD/ROM/DD shows a similar degradation in accuracy with Mach number. The $C_P$ error percentages on the body for this case is shown in Figure 4.4.c. The large error spike is due to a single cell error in the location of the sonic transition point on the suction surface. This case extends the POD/ROM/DD beyond its trained values in both Mach number and AoA. The addition of more modes improves the aft body errors on the lower surface, but introduced new errors on the suction surface after sonic transition. This is shown in Figure 4.4.d. In addition, the POD/ROM/DD with 5 MPFV became unstable at Mach 4.2 and AoA 5 degrees and was unable to produce a steady-state solution.




( a. POD/ROM/DD) (b. Full System)

Figure 4.5 $C_L$ from Steady Analysis

The POD/ROM/DD performed well at parameter values within or near the limits of the training. Errors were often less than 1% for AoAs less or equal to 2 degrees and Mach numbers between 2.5 and 2.7. The errors were larger the farther POD/ROM/DD was removed from the training window in the parameter space. To illustrate this, steady-state solutions from the 2 MPFV POD/ROM/DD were obtained across the parameter space. The steady-state pressure distributions were integrated across the body surface to produce $C_L$. The POD/ROM/DD lift coefficients are shown in Figure 4.5.a. For comparison, lift coefficients from the full order solver are shown in Figure 4.5.b. The difference between




( a. POD/ROM/DD, 2 MPFV) (b. POD/ROM/DD, 5 MPFV)

Figure 4.6 Error in $C_L$ for Steady Analysis

these results and the results from the full order solver are shown in Figure 4.6.a. Note that the errors grew as the parameter space was extended beyond the training window in both Mach and AoA. This error growth was reduced by adding more modes from the same training window. The same $C_L$ error plot for a 5 MPFV POD/ROM/DD is shown in Figure 4.6.b.

*4.5.2 Reduced Order Shock Region Results.* Next a steady implementation was developed for the POD/ROM/DD with a reduced order shock region. This implementation involved coupling the POD/ROMs for sections II and III into a single chord method solver. Enforcing smoothness on the boundary for solutions of adjoining POD/ROM regions required overlapping the domains by three cells to mimic the 1-D implementation from chapter 3. Smoothness in the overlap regions was enforced by using Lagrange constrained optimization. In addition, any substantial change in the shock location and strength within section II tended to destabilize the reduced order Newton iterations. As a result, steady state solutions over a range of Mach numbers had to be obtained by incrementing the shock motion. This required many solutions to the reduced order problem to move the shock any significant distance. Attempts to use small incremental steps in inlet Mach number to move the shock in a quasi-steady fashion to its appropriate location were unsuccessful. To increment the shock motion in a physically meaningful way, a time accurate chord method solver was implemented. The solution strategy was to time march the implicit solver un-

til the shock reached its steady state location for the inlet Mach number, and then use the steady solver to jump to the steady state solution. This implementation is developed below.

The domain decomposition reduced order mapping combined sections II and III as follows.

$$\underline{U}_{DD} = \begin{bmatrix} \underline{U}_{S2} \\ \underline{U}_{S3} \end{bmatrix} = \begin{bmatrix} \Psi_{S2} & [0] \\ [0] & \Psi_{S3} \end{bmatrix} \cdot \begin{bmatrix} \widehat{\underline{U}}_{S2} \\ \widehat{\underline{U}}_{S3} \end{bmatrix} \tag{4.3}$$

$$\Psi_{DD} = \begin{bmatrix} \Psi_{S2} & [0] \\ [0] & \Psi_{S3} \end{bmatrix} \tag{4.4}$$

The outputs of the flux calculations $R_{S2}(\underline{U}_{S2})$ and $R_{S3}(\underline{U}_{S3})$ from the full system were stacked by section, and the resulting vector was denoted as $R(\underline{U}_{DD})$. The implicit time accurate solution for the Euler equations was obtained using $F(\underline{U}_{DD})$ defined in equation (2.19)

$$F(\underline{U}_{DD}) = \underline{U}_{DD} - \underline{U}_{DD}^{init} - \Delta t R(\underline{U}_{DD})$$

The chord method implementation described in Chapter 2 was applied to obtain the solution of $F(\underline{U}_{DD}) = 0$. Once the shock motion ended, the chord method solution to $R(\underline{U}_{DD}) = 0$ could be solved to obtain the steady state solution.

The use of Lagrange constrained optimization to enforce smoothness in the overlapping regions of the two domains is developed below for the time accurate case. The Jacobian for the domain decomposed system was block diagonal,

$$\frac{dF(\underline{U}_{DD})}{d\underline{U}_{DD}} = \begin{bmatrix} \frac{dF_{S2}(\underline{U}_{S2})}{d\underline{U}_{S2}} & [0] \\ [0] & \frac{dF_{S3}(\underline{U}_{S3})}{d\underline{U}_{S3}} \end{bmatrix} . \tag{4.5}$$

Since the two solution domains were decoupled, the Lagrange constraint ensured smoothness between the domains by forcing the overlapping portion of both domains to have the same solution. The domains in Sections II and III were overlapped by a three cells. An $L^1$ constraint was introduced to enforce smoothness at the intersection of Sections II and III. A vector $T$ with the same dimensions of $\underline{U}_{DD}$ was defined such that $\underline{U}_{DD}^T T = 0$ when the flow variables for the overlapping sections had the same value. This was achieved by

placing a 1 or $-1$ in each fluid variable location corresponding to the overlap in Sections II and III respectively. Zeros were placed everywhere else in $T$. The dot product of $T$ with $\underline{U}_{DD}$ resulted in cancellation of the fluid variables when the overlapping portion of the solution domain was equivalent. If the overlapping fluid variables were not identical, the dot product produced a small scalar residual.

A Lagrange multiplier $\lambda$ was used to formulate the following constrained optimization.

$$\underline{y} = \begin{bmatrix} \underline{U}_{DD} \\ \lambda \end{bmatrix} \tag{4.6}$$

$$G(\underline{y}) = \begin{bmatrix} F(\underline{U}_{DD}) + \lambda T \\ \underline{U}_{DD}^T \cdot T \end{bmatrix} \tag{4.7}$$

The modified Jacobian was obtained as follows,

$$\frac{dG(\underline{y})}{d\underline{y}} = \begin{bmatrix} \frac{dF(\underline{U}_{DD})}{d\underline{U}_{DD}} & T \\ T^T & 0 \end{bmatrix} . \tag{4.8}$$

The reduced order mapping was modified to include the Lagrange multiplier,

$$\underline{y} = \begin{bmatrix} \underline{U}_{S2} \\ \underline{U}_{S3} \\ \lambda \end{bmatrix} = \begin{bmatrix} \Psi_2 & [0] & 0 \\ [0] & \Psi_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \widehat{\underline{U}}_{S2} \\ \widehat{\underline{U}}_{S3} \\ \lambda \end{bmatrix} . \tag{4.9}$$

The reduced order Jacobian can be quickly identified using the full system as originally introduces in equation (2.32),

$$\frac{d\widehat{G}(\widehat{\underline{y}})}{d\widehat{\underline{y}}} = (\Psi_\lambda^T \Psi_\lambda)^{-1} \Psi_\lambda^T \frac{dG(\underline{y}))}{d\widehat{\underline{y}}}$$

where $\frac{dG(\underline{y}))}{d\underline{\widehat{y}}}$ is given by the following,

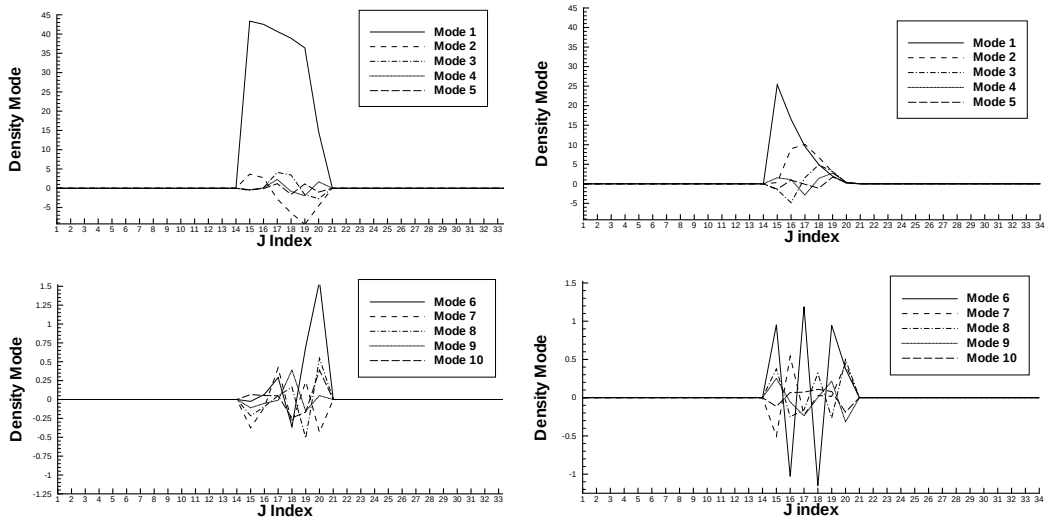
$$\frac{dG(\underline{y}))}{d\underline{\widehat{y}}} = \begin{bmatrix} \frac{dF(\underline{U}_{DD}))}{d\underline{\widehat{U}}_{DD}} & T \\ T^T \Psi_{DD} & 0 \end{bmatrix} \tag{4.10}$$

This result can be obtained directly from equation (2.20) using $\frac{dR(\underline{U}_{DD})}{d\underline{\widehat{U}}_{DD}}$ for the steady state solver.

$$\frac{dF(\underline{U}_{DD})}{d\underline{\widehat{U}}_{DD}} = \Psi_{DD} - \Delta t \frac{dR(\underline{U}_{DD})}{d\underline{\widehat{U}}_{DD}}$$

Initial attempts to obtain solutions were made by collecting 90 steady state solutions of the flow field evenly incremented in Mach number from Mach 2.7 to Mach 4.5. The angle of attack was zero degrees for all cases considered in this section. The shock traversed 6 cell centers over this range of Mach number, and the shock strength increases 38% along the stagnation line. The POD/ROM/DD test case involved obtaining the steady state solution for Mach 2.7 from an initial condition based on the steady state solution at Mach 3.65 translated to Mach 2.7 by shifting the base flow as described earlier. This created an initial condition with the shock location and strength of the Mach 3.65 solution, but the free stream conditions of Mach 2.7. Obtaining this solution required the POD/ROM/DD to move the bow shock 5 cell centers forward of the nose, and decrease the shock strength by 20%. No combination of modes from the set of 90 snapshots could be identified that resulted in shock motion. Instead, the shock tended to smear, grow or decline in strength, and eventually destabilize the solver.

A second attempt was made using 250 snapshots collected at even intervals from the full order time integration for the test case described above. From these snapshots a POD/ROM/DD was constructed using 10 MPFV in section II and 4 MPFV in section III. The flow field in section I remained constant at free stream conditions for this case. Total degrees of freedom for the POD/ROM/DD were 56, down from 7072 for the full order model for a 126 to 1 reduction in DOFs. This POD/ROM/DD was able to accurately move the shock and reach a steady state solution for Mach 2.7. The 10 density modes for the shock region are shown in Figure 4.7.a. This figure shows a cross section of the mode taken along the stagnation line. The stagnation line contained the strongest portion of the shock




(a. Modes from TI) (b. Modes from Steady State)

Figure 4.7 Density Modes for Shock Region

and the greatest range of shock motion in index space. The x-axis is the cell center index number across the stagnation line from the nose (index number 1) to the free stream inlet (index number 34). Section II consisted of index 14 to 21. The overlap region extended section II down to index 12, but this portion is not reflected in the figure. For comparison, the 10 density modes which were unable to produce a solution for the previous case are shown in Figure 4.7.b. Notice that the modes from time integration differ in skewness and magnitude from those obtained by snapshots of steady state solutions. Both sets of modes did exhibit a similar increase in spatial frequency with mode number, but this was the only consistent feature between them. The physical motion of the shock contained in the snapshots from time integration was reflected in the modes, and a seemingly complete collection of steady state solutions for the same range of shock motion was unable to produce the same modes. Unlike results for the 1-D case, the 2-D blunt body problem involved a large change in shock strength in addition to shock motion. These additional dynamics make POD training more problematic. The steady state values of $C_P$ at Mach 2.7 for both the POD/ROM/DD with the reduced order shock region and the full order model are shown in Figure 4.8. Errors are on the order of 10%, which are slightly larger than errors observed for the full order shock region case.

The robustness of the POD/ROM/DD was studied by increasing the desired inlet Mach number from the training value of Mach 2.7. The POD/ROM/DD was able to obtain accurate steady state solutions for a range of inlet Mach numbers between 2.7 and 3.0. Mach numbers higher that 3.0 drove the solver unstable. At these greater Mach numbers the snapshots from the time integration of shock motion at Mach 2.7 did not contain the proper information to form a shock of the proper strength at Mach 3.0. This drove the solver unstable when searching for a steady state shock even though the modes were adequate to move a transient shock over this range of Mach numbers. Both shock motion and shock strength must be adequately represented in the modes for POD/ROM to work in the shock region.

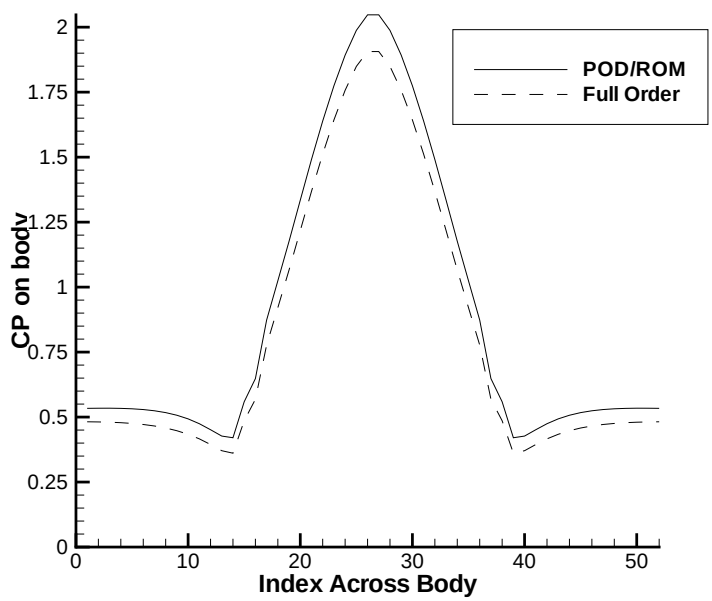



Figure 4.8 $C_P$ Error for POD/ROM/DD

The computational effort to produce the Mach 2.7 steady solution for the test case did not compare well with the full-order, explicit solver with local time stepping. The POD/ROM/DD for this case required the time accurate method to move the shock to its final position and the steady state chord method to obtain the steady state solution. A large increase in time step size was achieved with the time accurate chord method, requiring only

27 time steps to move the shock to its steady state location for an average CFL number of about 50. The full order solver required a CFL less than 1 for stability. However, aided by the supersonic flow conditions and the local time stepping, the full order solver converged to the steady state in only 1300 iterations. This required 1300 order $N$ function calls. In comparison, the POD/ROM/DD converged in 27 time steps plus 1 additional step to jump to steady state. But each of these steps required an update to the Jacobian for section II. Section II had 40 DOFs, and the numerical Jacobian calculation required 2 function calls per DOF. The total order $N$ function calls for the POD/ROM/DD to reach steady state was 2240. Wall clock time on the PC described in Table 4.1 was 118.42 seconds to obtain the POD/ROM/DD solution, while the wall clock time to obtain the full order solution was only 41.09 seconds. For comparison, the wall clock time to obtain the same solution from the POD/ROM/DD with the full order section II was 14.56 seconds.

### *4.6 Conclusions*

A very accurate and efficient reduced order model was generated for the 2-D blunt body problem using POD/ROM/DD to isolate the region of the flow field containing the shock motion. This region was solved at full order, while the remainder of the domain was treated with POD/ROM. The resulting ROM had a 4.3 to 1 reduction in DOFs and a computational improvement of 4.7 to 1. The POD/ROM/DD provided very accurate results over a parameter space of inlet Mach number and AoA that extended far beyond the parameter space used for POD training.

In contrast, the use of POD/ROM/DD with a reduced order shock region was not an advantageous approach for the 2D blunt body problem. POD training for the shock region was more problematic than initially expected. This was due to the great increase in shock strength accompanying motion of the bow shock off the blunt body. Although a set of modes to accommodate these harsh dynamics was identified, they did not produce a particulary accurate or robust reduced order model. The poor computational performance of the technique was not due to POD. Rather, the use of Newton's iterations in the presence of a moving discontinuity required many incremental solution steps to maintain stability. These incremental steps were more computationally expensive than the full or-

der integration. Full order explicit time integration for shock region yielded much better results than the implicit reduced order calculations for this problem.

However, POD/ROM/DD with a reduced order shock region could prove useful for unsteady cases. Use of implicit time marching (or even explicit integration) for the POD/ROM of the moving shock could provide a computational advantage in unsteady cases. For example, an unsteady transonic moving shock would occur in a largely subsonic solution domain. Subsonic cases would take longer to converge for the full order solver, and the time accurate integration would preclude use of local time stepping. Viscous cases requiring very fine numerical meshes would further reduce the full order solvers efficiency by decreasing the allowed time step for stability while increasing the problem order. This would magnify the CFL improvement seen in this POD/ROM/DD implementation. In addition, POD training for the shock region might be less problematic for periodic unsteady cases. Snapshots from a single cycle of oscillatory behavior could potentially apply to a wider parameter space. The unsteady transonic case is explored in the next chapter.

# V. *Analysis of a Transonic Aeroelastic Problem*

## 5.1 *Introduction*

In this chapter, the domain decomposition approach with POD/ROM is applied to an aeroelastic panel in cross-flow. A nonlinear coupling of the 2-D Euler equations and the von Kármán equation is used to simulate the dynamics of flow over the flexible panel, producing transonic LCO in the presence of certain panel parameters and free stream conditions. In the transonic regime, LCO involves a transonic shock that traverses the panel surface. Previous research (9) for this problem neglected reduced order modeling of the transonic case because of the difficulties that moving shocks pose to POD/ROM. The development of a POD/ROM to treat the transonic case is the subject of this chapter.

## 5.2 *Formulation*

In this section, the structural equations and fluid equations are described, and the domain decomposition is developed. Emphasis is given to the reduced order implementation of the fluid dynamics model, which includes a development of constrained optimization required to couple adjacent domains across internal boundaries.

*5.2.1 Structural Dynamics Equations.* Two-dimensional flow over a semi-infinite, pinned panel of length $L$ was considered. Panel dynamics were computed with von Kármán's large-deflection plate equation, which was placed in non-dimensional form using aerodynamic scales $L$ and $u_\infty$ (53) ($-\frac{1}{2} < x < \frac{1}{2}$):

$$\frac{\mu}{\lambda}\frac{\partial^4 w}{\partial x^4} - N_x \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial t^2} = \mu\left(\frac{1}{\gamma M_\infty^2} - P\right),$$

$$N_x = \alpha \int_{-\frac{1}{2}}^{\frac{1}{2}} \left(\frac{\partial w}{\partial x}\right)^2 dx.$$

Panel deflections $w(x)$, and velocities $s(x) = \dot{w}(x)$, were computed as a function of $x$ and $t$, given the values of the parameters $\mu$, $\lambda$, and $h/L$, where

$$\alpha = \left(\frac{6\mu}{\lambda}\right)\left(\frac{h}{L}\right)^{-2}(1-\nu^2).$$

Two boundary conditions were enforced at the panel endpoints ($x = \pm\frac{1}{2}$): $w = 0$ and $\frac{\partial^2 w}{\partial x^2} = 0$.

The structural dynamics equations were discretized and placed in first-order form to facilitate numerical integration. Spatial discretization was accomplished with second-order accurate, central differences, and the midpoint rule was used to compute the integral in the definition of $N_x$. A uniform distribution of grid points was assumed. The end conditions were explicitly accounted for by a modified central-difference formula. Excluding the endpoint states, the discrete structural variables were collocated into arrays $\underline{W}$ and $\underline{S}$, and then into the structural state, $\underline{Y}_s$. The values of $(\frac{1}{\gamma M_\infty^2} - P)$ at panel grid points were also collected into array $\underline{P}$. With these definitions, the structural equations take the first-order form

$$\dot{\underline{Y}}_s = \begin{bmatrix} 0 & L_N \\ I & 0 \end{bmatrix} \underline{Y}_s + \begin{bmatrix} \mu \underline{P} \\ 0 \end{bmatrix}, \tag{5.1}$$

where $I$ is the identity matrix, and $L_N$ is a nonlinear matrix operator given by

$$L_N \equiv -\frac{\mu}{\lambda}\frac{\delta^4}{\delta x^4} + N_x \frac{\delta^2}{\delta x^2}.$$

Here $\frac{\delta}{\delta x}$ represents central-difference, spatial discretization.

System (5.1) was integrated in time using an Euler implicit method which was first-order accurate in time. The Euler implicit method used the value for pressure provided by a coupled fluid model (9):

$$\left( I - \Delta_t \begin{bmatrix} 0 & L_N \\ I & 0 \end{bmatrix}^{n+1} \right) \underline{\mathsf{Y}} - \underline{Y}_s^n = \begin{bmatrix} \mu \underline{P} \\ 0 \end{bmatrix}, \tag{5.2}$$

$$F(\underline{\mathsf{Y}}) = 0,$$

where $\underline{\mathsf{Y}} = \underline{Y}_s^{n+1}$. The nonlinearity in (5.2) was accounted for through sub-iteration. $\underline{\mathsf{Y}}$ was iteratively computed using the chord method with $L_N$ evaluated about an undeflected

panel state, as represented by $L_N^0$:

$$\left( I - \Delta_t \begin{bmatrix} 0 & L_N^0 \\ I & 0 \end{bmatrix} \right) \underline{\mathsf{Y}}^{\nu+1} - \underline{\mathsf{Y}}^{\nu} = -F(\underline{\mathsf{Y}}^{\nu}). \tag{5.3}$$

The matrix $L_N^0$ was computed and $LU$-decomposed once at the start of the time-integration procedure to increase the efficiency with which (5.3) was solved. The chord method converged quickly for the transonic cases considered due to the small deflections of the panel ($< 2\%$ of the panel length). Four sub-iterates applied at each time step were generally sufficient to drive $F$ to near machine zero.

A fine structural grid of 101 interior points (103 points including the endpoints of the panel, where $w = s = 0$ was enforced) was used to discretize the panel. The large number of structural grid points was used to reduce the effects of translating fluid values to structural nodes.

*5.2.2 Fluid Equations.* Inviscid flow over an infinite flat plate was considered. The fluid dynamical system from equation (2.7) is reproduced below,

$$\frac{d\underline{U}(t)}{dt} = R(\underline{U}(t)) \ .$$

As described earlier, external boundaries were handled with ghost cells. The panel dynamics were enforced using a transpiration boundary condition for $(-\frac{1}{2} < x < \frac{1}{2}, y = 0)$ as described in section 2.1.

*5.2.3 Grid Generation.* The elastic panel model was inserted into the solid boundary at $-\frac{1}{2} < x < \frac{1}{2}$ and $y = 0$. A square domain of dimension $-23.668 < x < 23.668$ and $0 < y < 24.6$ was used for this research (the panel length was one non-dimensional length unit). The large domain was intended to contain all the flow dynamics, and the arbitrary choice of approximately 25 chord lengths in all directions mimics the domain used by other authors for the same problem (9, 54). Extending the domain to include all flow dynamics facilitates the use of characteristic boundary conditions for the outer portion of the domain. The results may not be general, since the domain size impacts the

panel response (see appendix F for an example). The structured grid is shown in Figure 5.1. It used 141 nodes along the solid surface and 116 nodes extending to the free stream. The spacing of the grid points increased geometrically from the solid wall in the normal direction. In the streamwise direction, the grid spacing was held constant at $\Delta_{wall} = 0.0125$ over the entire panel, and was increased geometrically upstream and downstream of the panel position.

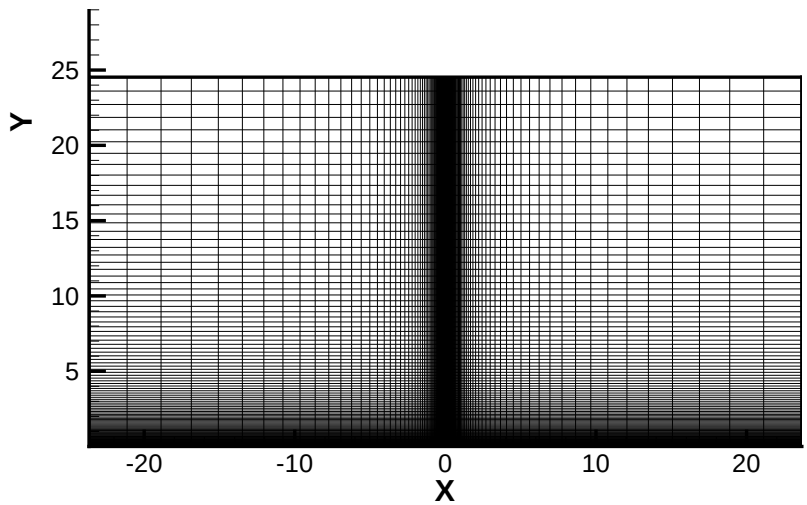



Figure 5.1 Grid for Flow over Panel

*5.2.4 Fluid-Structure Coupling.* The structural system from equation (5.1) was loosely coupled to the fluid system from equation (2.7) through the transpiration boundary condition. The structural solver was explicitly stepped forward in time by $\Delta_t$ from time level $t_n$ to time level $t_{n+1}$ using the pressure value on the panel from the fluid system. An extrapolated panel pressure value was used to avoid time lagging errors that can arise in this coupling scheme (54). The pressure value at time level $n$, $\underline{P}^n$, was used to estimate

$\underline{P}^{n+1}$ by

$$\begin{aligned}
\underline{P}_{ex}^{n+1} &= \underline{P}^{n} + \Delta\underline{P}^{n} \\
\Delta\underline{P}^{n} &= \underline{P}^{n} - \underline{P}^{n-1} \\
\underline{P}_{ex}^{n+1} &= 2\underline{P}^{n} - \underline{P}^{n-1}.
\end{aligned} \tag{5.4}$$

$\underline{P}_{ex}^{n+1}$ was used to update the panel deflection and velocity $\underline{Y}_{s}^{n+1}$, which were incorporated into the transpiration boundary condition. The fluid system was then time stepped to update the flow field to $\underline{U}^{n+1}$. This sequence was repeatedly performed to integrate the coupled system and produce time-accurate results.

The fluid system dominated the compute time required to produce solutions for the coupled system. Time step size was limited by the stability of the flow solver, and the number of degrees of freedom for the flow solver (64400) was far larger than the structural model (202). As such, the focus of this research was to couple a reduced order model of the flow field to the full-order structural model and produce an accurate aeroelastic panel response with less computational expense.

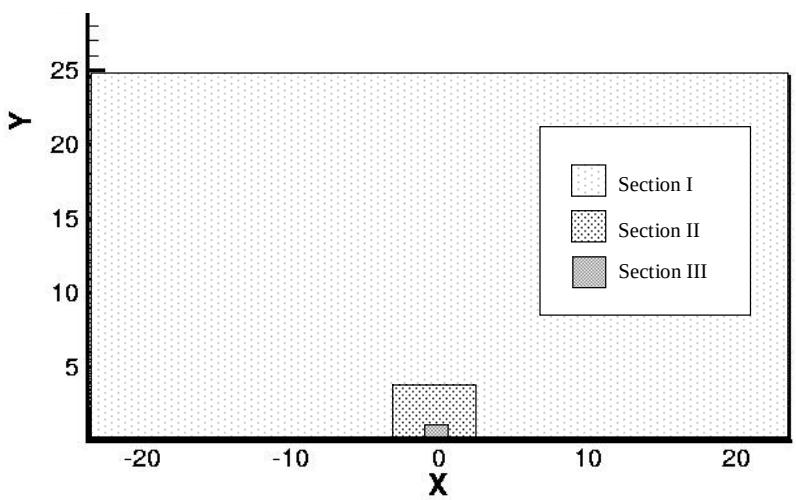



Figure 5.2 Domain Decomposition

*5.2.5 Domain Decomposition.* The solution domain was broken into sections to facilitate the use of POD with the moving transonic shock. Isolation of the transonic shock was the primary goal of the domain decomposition. Since the transonic shock was always attached to the panel, the region of the flow field directly above the panel was identified as the shock region. In addition, domain decomposition was used to improve the computational performance when obtaining unsteady solutions from the reduced order solver. Different time steps were employed for these different regions of the flow field to minimize the number of solver iterations (40, 55).

The solution domain was divided into three regions to form the POD/ROM/DD. The geometric shape of the region did not influence the solver performance, so regions formed by the intersection of rectangles were used for simplicity. The results presented were generated using the domain decomposition depicted in Figure 5.2. The large outer region is denoted section I and is called the "far-field." Section II is the middle region whose outer boundary is defined by three line segments. These line segments are the sides of a rectangle with corners at the $(x, y)$ pairs $(-3, 0)$, $(-3, 4)$, $(2.5, 4)$ and $(2.5, 0)$. Section II is called the "near-field," and it will be solved more frequently than the far-field to update the internal boundary shared with section III. Section III contains the flow over the panel. Its outer boundary is a rectangle with corners at the $(x, y)$ pairs $(-0.7, 0)$, $(-0.7, 0.65)$, $(0.7, 0.65)$ and $(0.7, 0)$, and is the previously mentioned shock region. A small shock forms on the upstream portion of the panel, and traverses over a large portion of the panel's length. This is the moving shock of interest for this analysis, and it is confined entirely to section III. No overlap is included in this decomposition. Domain overlap will be addressed later in the chapter. Section I contains 9300 cell centers, section II contains 4886 cell centers, and section III contains the remaining 1914 cell centers.

*5.2.6 Reduced Order Modeling.* Both implicit and explicit time accurate methods were used to recast the governing equations to solve for the reduced order variables. Implicit time stepping was an efficient way to handle the non-shocked regions of the domain decomposed flow field (sections I and II). Here the dynamics were benign enough that the reduced order Jacobian required no updates. Explicit time stepping was used for the shock

region since the moving discontinuity in section III was too strong a nonlinearity for the implicit scheme to capture efficiently.

*5.2.7 Implicit Formulation with Overlapping Domains.* Under certain conditions, constraints had to be introduced into the reduced order solver to enforce smoothness on the internal boundary between adjacent domains. Consider an implicit time integration step from $t_n$ to $t_{n+1}$ for a domain section whose full order and reduced order vector of flow variables are denoted $\underline{U}_S$ and $\widehat{\underline{U}}_S$ respectively. This domain section shares an internal boundary with an adjacent domain section, they overlap, and the adjacent section has already been time integrated from $t_n$ to $t_{n+1}$. When $\widehat{\underline{U}}_S$ is integrated to $t_{n+1}$, the goal of the boundary coupling is to match the overlapping region of $\underline{U}_S$ with the corresponding portion of the adjacent domain. The implicit, time-integration algorithm can be forced to do this using Lagrange constrained optimization. Both $C^1$ and $C^2$ constraint options from Chapter 2 are considered. The time-accurate implementation for both is described below.

*5.2.7.1 $C^1$ Constraints.* Referring to the function $F$ from equation (2.19) and the functional $\ell(\underline{U})$ from equation (2.25), a series of $C^1$ constraints were introduced to enforce equality within a subset of the overlap region. For some number of constraints $I$, a total of $I$ vectors $\underline{U}_i^{over}$ and $T_i$ were required with the same dimensions as $\underline{U}_S$. For the $i^{th}$ constraint, $\underline{U}_i^{over}$ contained flow values from a subset of the overlapping portion of the adjacent domain section. The goal of the $i^{th}$ constraint was to force $\underline{U}_S$ to match the flow values in $\underline{U}_i^{over}$. These flow values were collocated to the vector locations in $\underline{U}_i^{over}$ that corresponded to the identical locations within the overlapping region of $\underline{U}_S$. $T_i$ contained a 1 in each fluid variable location corresponding to the selected fluid variables from the adjacent domain, collocated into $\underline{U}_i^{over}$. Zeros were placed everywhere else in $T$. The dot product of $T$ with $(\underline{U}_S - \underline{U}_i^{over})$ tended to zero when the flow variables in $\underline{U}_S$ matched $\underline{U}_i^{over}$. Otherwise, the dot product produced a small scalar residual. This constraint approximated the $L^1$ norm, or average value, of the flow variable differences in this subset of the overlapping domain. Fluid values for $\underline{U}_i^{over}$ could be chosen as required to produce a reasonable solution. Each constraint could apply to any combination of the four fluid variables, over any portion of the overlapping region.

One Lagrange multiplier $\lambda_i$ was used for each constraint. Lagrange constrained optimization (51) modifies $\ell(\underline{U}_S)$ by adding each linear constraint as shown in equation (5.6) (the $n+1$ superscript on $\underline{U}_S$ is dropped for convenience). This new functional $Q$, called the Lagrangian, whose minimizing value was obtained by finding the values of $U_S$, and $\lambda_i$ such that the gradient of $Q$ was zero.

$$\underline{y} = \begin{bmatrix} \underline{U}_S \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_I \end{bmatrix} \tag{5.5}$$

$$Q(\underline{y}) \equiv \ell(\underline{U}_S) + \sum_{i=1}^{I} \lambda_i (\underline{U}_S - \underline{U}_i^{over})^T \; T_i \tag{5.6}$$

$$G(\underline{y}) \equiv \frac{dQ(\underline{y})}{d\underline{y}} = [0] \tag{5.7}$$

$$G(\underline{y}) = \begin{bmatrix} F(\underline{U}_S) + \sum_1^i \lambda_i T_i \\ (\underline{U}_S - \underline{U}_1^{over})^T \; T_1 \\ (\underline{U}_S - \underline{U}_2^{over})^T \; T_2 \\ \vdots \\ (\underline{U}_S - \underline{U}_I^{over})^T \; T_I \end{bmatrix} \tag{5.8}$$

The reduced order mapping was modified to include the Lagrange multipliers.

$$\underline{y} = \begin{bmatrix} \Psi_S & \begin{bmatrix} 0 \\ \vdots \end{bmatrix} & \begin{bmatrix} 0 \\ \vdots \end{bmatrix} & \begin{bmatrix} 0 \\ \vdots \end{bmatrix} & \begin{bmatrix} 0 \\ \vdots \end{bmatrix} \\ [0 \; \ldots] & 1 & 0 & 0 & 0 \\ [0 \; \ldots] & 0 & 1 & 0 & 0 \\ [0 \; \ldots] & 0 & 0 & \ddots & 0 \\ [0 \; \ldots] & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \widehat{\underline{U}}_S \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_I \end{bmatrix} \tag{5.9}$$

Equation (2.32) (repeated below) was used to identify the reduced order Jacobian from the full system as follows.

$$\frac{d\widehat{G}(\underline{\widehat{y}})}{d\underline{\widehat{y}}} = (\Psi_\lambda^T \Psi_\lambda)^{-1} \Psi_\lambda^T \frac{dG(\underline{y}))}{d\underline{\widehat{y}}}$$

Where $\frac{dG(\underline{y})}{d\underline{\widehat{y}}} = \frac{dG(\underline{y})}{d\underline{y}} \Psi_\lambda$, which was

$$\frac{dG(\underline{y}))}{d\underline{\widehat{y}}} = \begin{bmatrix} \frac{dF(\underline{U}_S)}{d\underline{\widehat{U}}_S} & T_1 & T_2 & \dots & T_I \\ T_1^T \Psi_S & 0 & 0 & \dots & 0 \\ T_2^T \Psi_S & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ T_I^T \Psi_S & 0 & 0 & \dots & 0 \end{bmatrix},$$

and $\frac{dF(\underline{U}_S)}{d\underline{\widehat{U}}_S}$ was obtained from $\frac{dR(\underline{U}_S)}{d\underline{\widehat{U}}_S}$ by using equation (2.16).

*5.2.7.2 $C^2$ Constraints.* The optimization algorithm was modified to use $C^2$ constraints. Again, a series of constraints were considered instead of using a single constraint for the entire overlap region as shown in equation (2.24). The $i^{th}$ constraint becomes

$$C_i^2 = (\underline{U}_S - \underline{U}_i^{over}) \cdot (\underline{U}_S - \underline{U}_i^{over}) = 0 \ , \tag{5.10}$$

and the Lagrangian was minimized as follows:

$$Q(\underline{y}) = \ell(\underline{U}_S) + \sum_{i=1}^{I} \lambda_i (\underline{U}_S - \underline{U}_i^{over})^T (\underline{U}_S - \underline{U}_i^{over}) \tag{5.11}$$

$$G(\underline{y}) = \frac{dQ(\underline{y})}{d\underline{y}} = [0] \tag{5.12}$$

$$G(\underline{y}) = \begin{bmatrix} F(\underline{U}_S) + 2\sum_1^i \lambda_i(\underline{U}_S - \underline{U}_i^{over}) \\ (\underline{U}_S - \underline{U}_1^{over})^T(\underline{U}_S - \underline{U}_1^{over}) \\ (\underline{U}_S - \underline{U}_2^{over})^T(\underline{U}_S - \underline{U}_2^{over}) \\ \vdots \\ (\underline{U}_S - \underline{U}_I^{over})^T(\underline{U}_S - \underline{U}_I^{over}) \end{bmatrix}. \tag{5.13}$$

The Jacobian for the Newton iterations comes from equation (2.32), where $\frac{dG(\underline{y})}{d\widehat{\underline{y}}}$ is found to be

$$\frac{dG(\underline{y})}{d\widehat{\underline{y}}} = \begin{bmatrix} \frac{dF(\underline{U}_S)}{d\widehat{\underline{U}}_S} + 2\sum_{i=1}^I \lambda_i \Psi_i^{over} & \dots & 2(\underline{U}_S - \underline{U}_i^{over}) & \dots \\ \vdots & \vdots & \vdots & \vdots \\ 2(\underline{U}_S - \underline{U}_i^{over})^T \Psi_S & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}. \tag{5.14}$$

The $\Psi_i^{over}$ term is simply the matrix $\Psi_S$ overwritten with zero for any mode value not pertaining to the subset of the overlap region and the fluid variables constrained by the $i^{th}$ constraint. Unfortunately, this implementation requires the Jacobian be reformed for each Newton iteration, since $\frac{dG(\underline{y})}{d\widehat{\underline{y}}}$ is now a function of the unknowns $\underline{U}_S$ and $\lambda_i$. Even though $\frac{dF(\underline{U}_S)}{d\widehat{\underline{U}}_S}$ does not need to be recomputed, the additional multiplies within the "$2(\underline{U}_S - \underline{U}_i^{over})^T\Psi_S$" and the "$2\sum_{i=1}^I \lambda_i \Psi_i^{over}$" terms result in this implementation being less efficient.

### *5.3 Results*

The full-order, transonic LCO behavior is described first, followed by a discussion of the POD/ROM/DD solver implementations. Observations are made on the effectiveness of overlap, boundary constraint types, and domain-specific time steps. The accuracy, order reduction, and computational savings for a variety of implementations are discussed. Finally, the robustness is explored by observing the change in solver accuracy as panel dynamic pressure is modified.

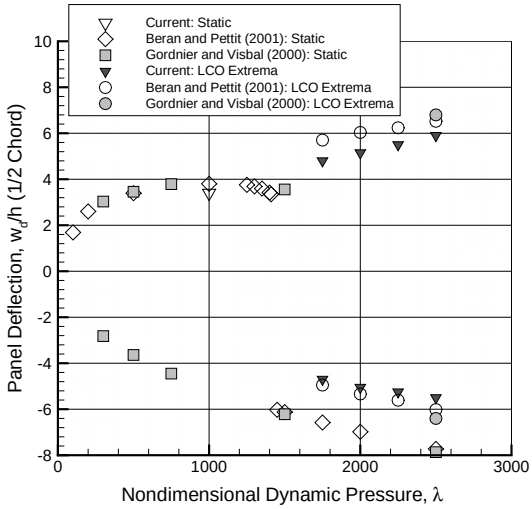



Figure 5.3 Panel Response with Changing Dynamic Pressure

*5.3.1 Full-System Transonic LCO.* Variations in panel dynamic pressure ($\lambda$) constituted the parameter space of interest for the transonic panel analysis. Increased values of $\lambda$ correspond to decreased panel stiffness. Panel response at Mach 0.95 across this parameter space is illustrated in Figure 5.3. For comparison, this figure contains data obtained from the literature for the same problem (9, 56). The maximum panel deflection amplitude at the $\frac{1}{2}$-chord point is shown for a range of $\lambda$. When $\lambda$ was less than 1750, panel stiffness was sufficient to prevent unsteady oscillatory behavior. Any initial panel deflection and velocity eventually damped to a static deflection state, with the panel deflected either upwards or downwards depending on the initial condition. In Figure 5.3, the two static deflection branches are evident for $\lambda < 1750$. Values of $\lambda$ above 1750 resulted in oscillatory panel behavior with the proper initial condition, otherwise the panel deflection settled at the downward deflected static solution. Figure 5.3 illustrates how the upper static branch evolves into LCO, while the lower branch remains static when $1750 > \lambda > 2500$. The mid-chord amplitude of the upward panel deflection during LCO is about 15% lower than other cases reported in the literature. The dissipation in the first order Roe solver provides more damping of the high frequency panel deflections than the higher order solvers used in the archived data. This additional damping is the primary cause of the muted panel deflections. The muting of the upward panel deflection is not as prominent in the static

case. Otherwise, the dynamic and static behavior of the aeroelastic model for both static and LCO solutions from the full-system simulation are shown to be in good agreement with results found in the literature for the same problem (56, 9). The muted panel deflections were not a concern since the transonic moving shock was the non-linear feature for analysis. The POD/ROM will be shown to match the full-system accurately, and a full-system model with higher fidelity could be inserted without loss of generality.

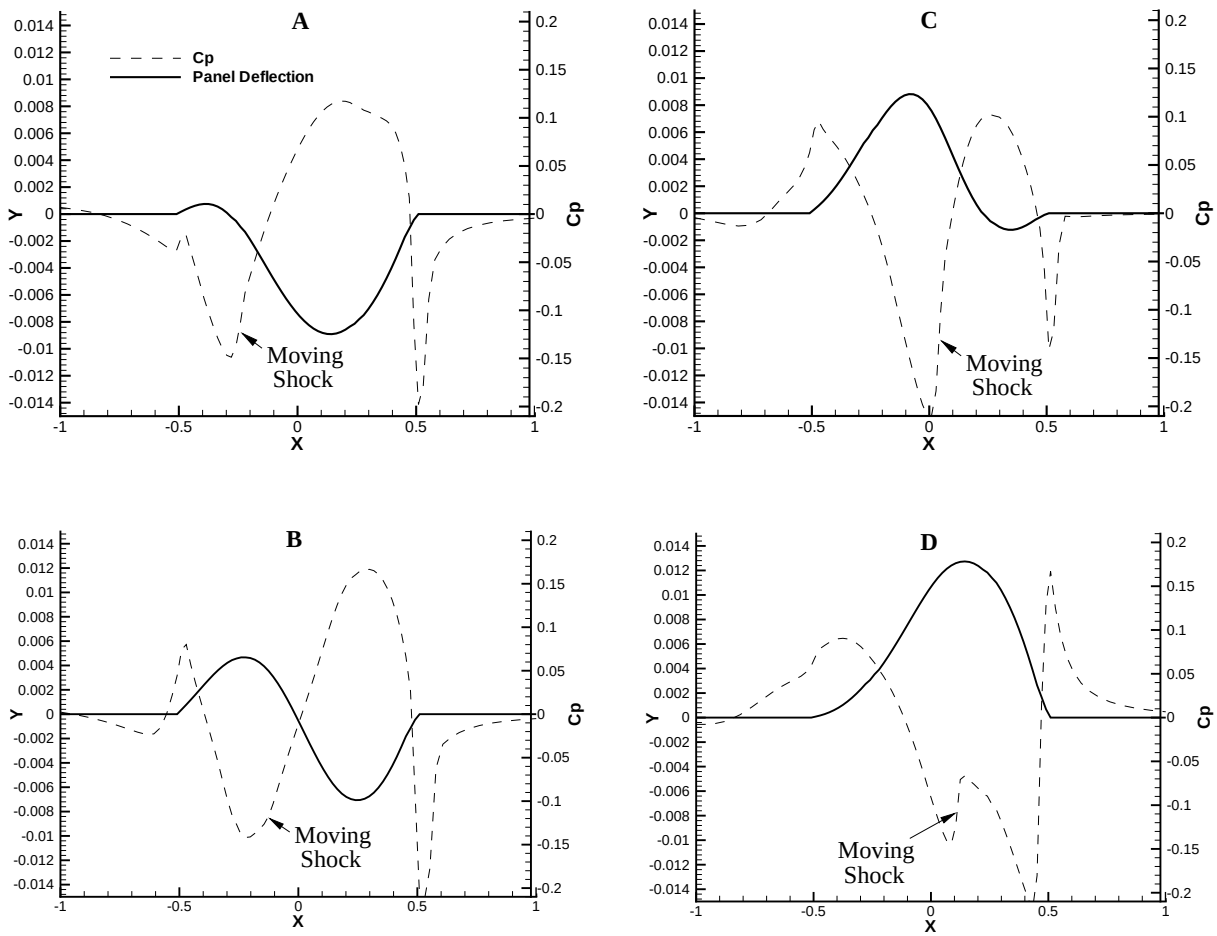



Figure 5.4 Moving Transonic Shock

With $\lambda = 2500$, the full-system simulation produced the transonic LCO shown in Figures 5.4 and 5.5. A portion of the LCO cycle exhibits a moving shock (57). The panel deflections corresponding to the portion of LCO containing a moving shock are shown in

Figure 5.4. The plot of $C_P$ shows the shock motion progressing from subplot A to D. The dissipation in the first-order Roe's solver slightly smears the moving shock. A stronger shock also develops at the end of the panel during this portion of the LCO. Because this was a stationary shock, it did not pose a significant problem for the POD/ROM/DD.

Figure 5.5 shows the panel time history at the $\frac{3}{4}$-chord point. Transonic LCO was established very quickly from a small panel deflection in free stream conditions. The initial panel deflection was a small, downward sinusoidal panel deflection of $1 \times 10^{-3}$. The

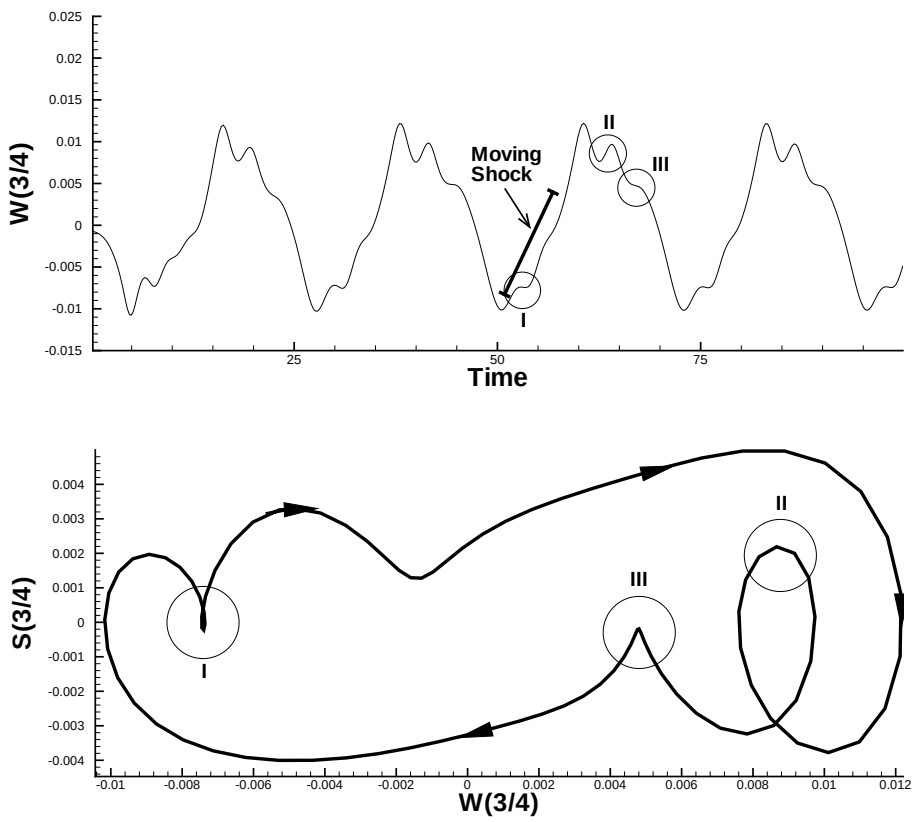



Figure 5.5 LCO Onset

LCO was well established by 15 seconds, which was about half the period of a single panel oscillation. This extremely rapid LCO onset makes time integration an efficient implementation for analysis of this case. The bottom half of Figure 5.5 is a phase plot of panel velocity and position at $\frac{3}{4}$ chord. Strong nonlinearities are evident in the upswing

of the panel (at point I), and again after the crest at points II and III on the downswing. These appear as loops or peaks in the phase map.

*5.3.2 Solver Implementation.* Two cases of the POD/ROM/DD were considered. The first case modeled the shock region at full order, the second case modeled it with POD/ROM. The far-field (section I) and the near-field (section II) used the same ROM for both cases. Both the far and near-fields used an implicit, time-accurate solver, and both were modeled with POD/ROMs trained by 100 snapshots taken at evenly spaced intervals over one complete cycle of the panel LCO. Five modes per fluid variable were adequate for both sections, resulting in 20 DOFs in the POD/ROM for each. The far and near-field domains were overlapped by three cells everywhere on the common boundary. Constraints were required when the time steps were different, otherwise the near and far-field were effectively modeled with a single domain using five DOFs per fluid variable.

Both $C^2$ and $C^1$ constraints were implemented on the shared boundary between the far and near-fields. A single $C^2$ constraint on density over the entire overlapping region was sufficient to produce accurate results. To use $C^1$ constraints, a careful selection of subsets within the overlap region was required. A series of 5 subsets with one $C^1$ constraint for each subset, were used to couple the near and far-field domains. Density was the constrained fluid variable. Each subset contained a vertical column of three cells, positioned on the top portion of the internal boundary as shown in Figure 5.6. Each vertical strip is referred to as a "staple." The $x$ locations for each staple were $-1$, $-0.25$, 0, 0.25 and 1. Each staple comprised a single cell in width, and 3 cells in height to traverse the overlap region in the vertical direction.

This research produced several insights regarding the effective use of $C^1$ constraints. First, the size of the subsets played a critical role in the effectiveness of the $C^1$ constraints. A large number of cells within the subset provided many DOFs for the optimization to manipulate. Many DOFs in the overlap region allowed the average of the flow field error to be zero without the overlapping flow fields matching very closely. Conversely, if the subset was a single cell within the overlapping domain, the error between fluid values in the overlapping cell was driven to machine zero (order $10^{-16}$).

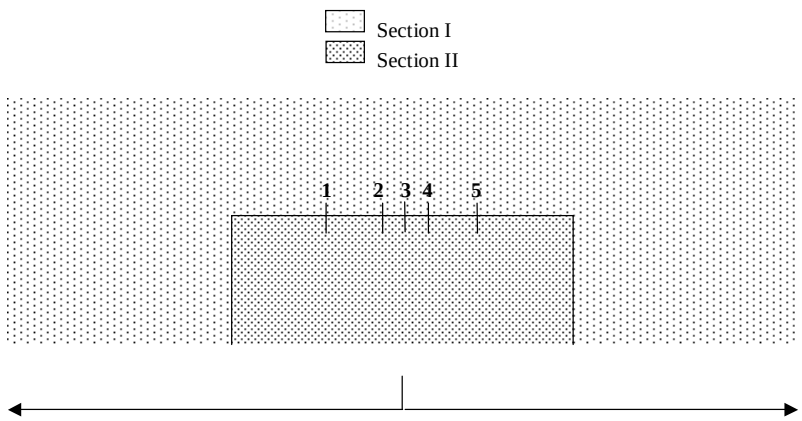



Figure 5.6 $C^1$ Constraints

In addition, choosing the size of the constrained subset was complicated by the projection into the reduced order space. The solution within the overlap region was a linear combination of 5 spatial modes. The solver would sacrifice the validity of the solution in the rest of the flow field to meet the constraint in the overlap region. For example, in driving the flow error to order $10^{-16}$ for a constraint on a single cell, the flow solution in the non-overlapping part of the domain would produce a non-physical result. The POD modes were an approximation of the flow of order $10^{-6}$, therefore they could not support both an order $10^{-16}$ constraint in the overlap region, and a physical flow field solution elsewhere.

Finally, the position of each constrained subset was determined by the flow dynamics. In general, regions of high dynamics on the boundary required some form of constraint while regions of lower dynamics required none. The careful placement of a few localized $C^1$ constraints could stabilize the entire boundary. The staple size and locations depicted in Figure 5.6 took all of these issues into consideration.

When the shock region was modeled with POD/ROM, 20 modes per fluid variable were obtained from the set of 100 snapshots described earlier. An explicit time accurate solver was used for the shock region. No overlap was required between the near-field and

the shock region. An attempt to introduce one cell of overlap everywhere on the common boundary produced an instability in the full system. The location of an internal boundary in a region of strong nonlinear flow behavior can cause solver instabilities (41). A wide variety of $C^1$ constraint combinations were tried. While some combinations stabilized the overlapping boundary between the near-field and shock region, none produced a particularly accurate flow field. The number of constraints was limited by the number of modes used per fluid variable. When more constraints were used, the reduced order Jacobian was not invertible.

*5.3.3 Accuracy, Order Reduction and Compute Time.* The domain sizes and the POD/ROMs for the near and far-field were not varied. As a result, efficiency was only affected by the number of domain updates for the near and far-field, the time-step size for the shock region, and the type of constraint. Each solver is referred to by these key values for convenience. For example, consider the POD/ROM/DD with a CFL of 0.9 in the full-order shock region, the near-field domain updated every 40 shock region time steps, and the far-field domain updated every 80 time steps. The CFL of 0.9 yielded a time step size of $4.6455^{-3}$ (non-dimensional time units), and the near and far-field were updated every 0.186 (which is $40 \times 4.6455^{-3}$) and 0.37 (which is $80 \times 4.6455^{-3}$) respectively. This was denoted as the full-case 0.9/0.186/0.37 model. When the full-order shock region is replaced by a POD/ROM with CFL of 2.5, the CFL of 2.5 yielded a time step size of $1.29^{-2}$, and the near and far-field were updated every 0.52 (which is $40 \times 1.29^{-2}$) and 1.04 (which is $80 \times 1.29^{-2}$) respectively. This POD/ROM/DD is denoted as the POD-case 2.5/0.52/1.04 model. Both the full-case 0.9/0.186/0.37 model and the POD-case 2.5/0.52/1.04 model use the same POD basis for the near and far-field domains.

The panel LCO from full-case 0.9/1.48/2.96 and POD-case 2.5/2.06/4.13 models are compared at the $\frac{1}{2}$-chord position and the $\frac{3}{4}$-chord position in Figure 5.7. The use of either $C^1$ or $C^2$ constraints provided similar accuracy. The moving shock was replicated accurately within the shock region for both models. Regional time-stepping over a range of near-field updates from $\frac{1}{50}$ to $\frac{1}{12}$ of one LCO period yielded similar accuracy. Models discussed in this section generally used larger time steps to improve computational

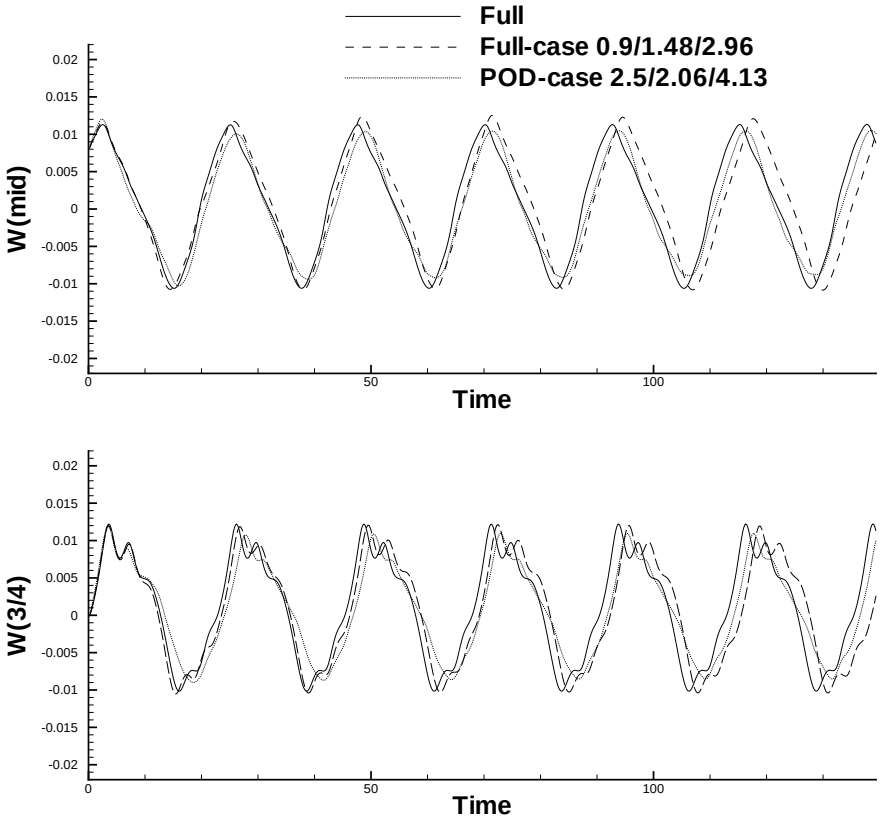



Figure 5.7 Modeled Panel Response

performance. The CFL improvement to 2.5 was the largest allowed for stability of the POD/ROM in the shock region. Much larger CFL increases with the sub-space projection method have been reported in the literature (9, 21), however these cases did not involve a moving shock. This limitation in CFL improvement is consistent with the difficulties seen for the blunt body application in the last chapter. Similar limitations were also encountered when time integrating the transient flow field for the 1-D quasi-nozzle flow in Chapter 3.

While both models produced panel responses with similar accuracy, the full-case implementation produced results with less computational cost. The results of several runs are shown in Table 5.1. The full system solver required 358.08 seconds (wall-clock time) to compute 25 non-dimensional seconds on the Compaq workstation described in Table 5.2. This modeled one cycle of the panel LCO. In contrast, the POD-case 2.5/2.06/4.13

Table 5.1 Computational Costs

| Case | Wall Clock Time | Constraint |
|---|---|---|
| Full System | 358.08 | N/A |
| Full-case 0.9/0.185/0.37 | 161.51 | $C^1$ |
| Full-case 0.9/0.185/0.37 | 196.10 | $C^2$ |
| Full-case 0.9/0.74/1.48 | 128.96 | $C^1$ |
| Full-case 0.9/1.48/2.96 | 123.03 | $C^1$ |
| Full-case 0.9/1.48/2.96 | 136.43 | $C^2$ |
| POD-case 2.5/0.515/1.03 | 142.46 | $C^1$ |
| POD-case 2.5/2.06/4.13 | 133.15 | $C^1$ |

required 133.15 seconds, while the full-case 0.9/1.48/2.96 required 128.96 seconds. The full-order solver was more efficient in the shock region than the explicit time integration of POD/ROM. The CFL increase to 2.5 was not large enough to compensate for this inefficiency.

Table 5.2 Work Station Specifications

| | |
|---|---|
| Processor | 500MHz Alpha |
| Cache | 4M |
| Operating System | Digital UNIX v4.0G |
| Memory | 512M |
| Fortran Compiler | Digital Fortran 90 Compiler v5.2 |
| Optimizations | -O |

The use of $C^1$ constraints approached the efficiency of $C^2$ constraints when the number of far-field updates was reduced. $C^2$ constraints required additional multiplies to update the Jacobian for every Newton iterate. For example, the full-case 0.9/0.185/0.37 model required 67 far-field updates. The $C^2$ constraints were 21.4% less efficient for this case. When the number of far-field updates was reduced to 8 for the full-case 0.9/1.48/2.96 model, the $C^2$ constraint case was only 5.79% less efficient. The additional multiplies with $C^2$ constraints were offset by a reduction in DOFs (5 $C^1$ constraints versus only one $C^2$

constraint), and the $C^2$ constraints only required two Newton iterations in comparison to four with $C^1$ constraints.

The full system had 64400 DOFs, the full-case models had 7701 DOFs (including 5 $C^1$ constraints), and the POD-case models had 125 DOFs. The 88% DOF reduction for the full-case models resulted in a 66% reduction in computational cost with this solver implementation. The repeated use of the full-order function call in the Newton iterations prevented the cost reduction from reaching 88%. The additional reduction in DOFs for the POD-case models did not result in additional cost savings for this solver implementation. While the explicit integration scheme demonstrated the existence and adequacy of the POD modes for capturing a moving shock, a Galerkin-type approach is motivated to realize a computational savings in proportion to the reduction in DOFs. Extrapolated to 3-D, an Euler solver of similar fidelity in all three dimensions would yield approximately $5,300,000$ DOFs. A similar domain decomposition would produce full-case models with approximately $335,000$ DOFs. This would be a 93% reduction in DOFs for the full-case models, slightly better than the 88% reduction in DOFs for the 2-D implementation. Similarly, a slightly better than 66% reduction in computational cost could be expected from the full-case implementation in 3-D.

*5.3.4 Robustness.* The panel phase plot in Figure 5.8 demonstrates the ability of POD/ROM/DD to capture nonlinear behavior at $\lambda = 2500$, which was the value used for POD training. When POD is used in the shock region, the flow field near the panel is constrained by the projection into the reduced order space. The flow field does not damp high-frequency structural responses in the same way as the full-order flow field, resulting in the differences between the full system and the POD-case responses on the phase plot. Reducing the number of DOFs used for the structural model could improve the POD-case panel response by eliminating high-frequency structural modes. At Mach 0.95, LCO states exist for any value of $\lambda$ greater than 1750 (9). Only static panel deflections have been identified for $\lambda < 1750$. POD/ROM/DD's trained with snapshots from LCO at Mach 0.95 and $\lambda = 2500$ were able to model the LCO states at Mach 0.95, however, neither POD/ROM/DD could produce a static solution for $\lambda < 1750$. The snapshots only

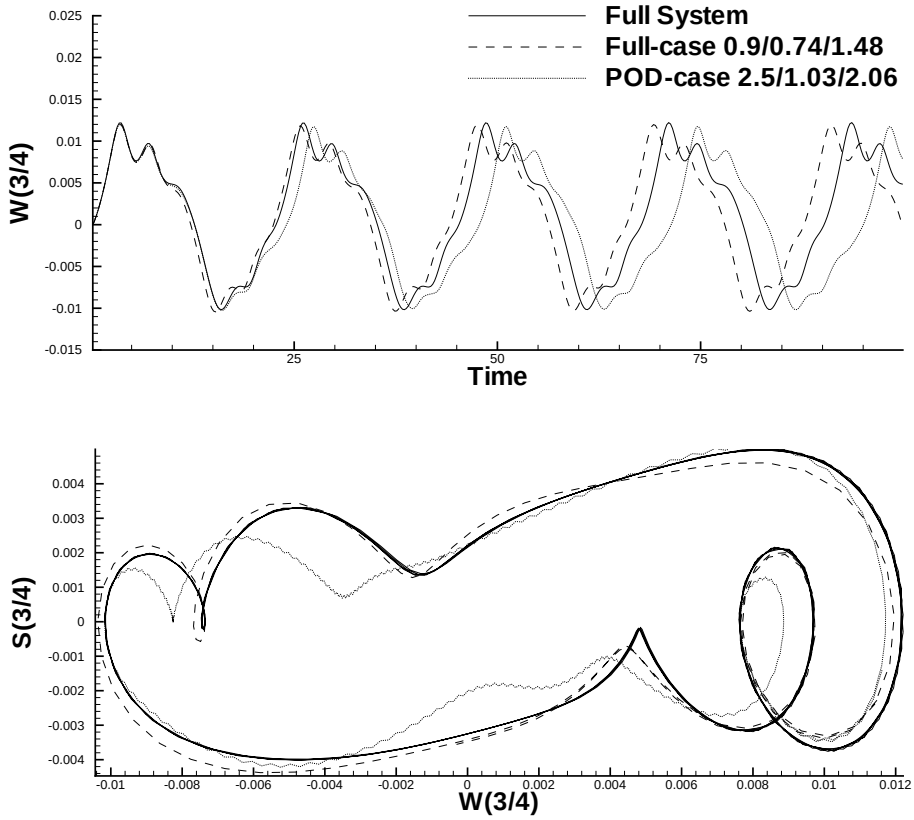



Figure 5.8 LCO for $\lambda = 2500$

contained data from an LCO flow field, and the resulting modes could not be linearly combined to produce a static solution.

Next, the extendability of the POD/ROM/DD was explored. The POD basis from training at $\lambda = 2500$ (the same model used for results shown in Figure 5.8) was used to model the panel response for $\lambda = 2000$. Referring to Figure 5.3, this parameter variation encompasses $\frac{2}{3}$ of the LCO parameter space. No effort was made to optimize POD training for accuracy or robustness over the parameter space. This example demonstrates the robustness of POD/ROM in the presence of moving shocks with sub-optimal training.

The panel response and phase plot for $\lambda = 2000$ is shown in Figure 5.9. The full system LCO response at $\lambda = 2500$ is shown with a thick dashed line. At $\lambda = 2000$ the panel response is reduced, and the phase plot fits within the phase plot for $\lambda = 2500$. Both POD/ROM/DD's track this trend, while preserving the nonlinear panel behavior

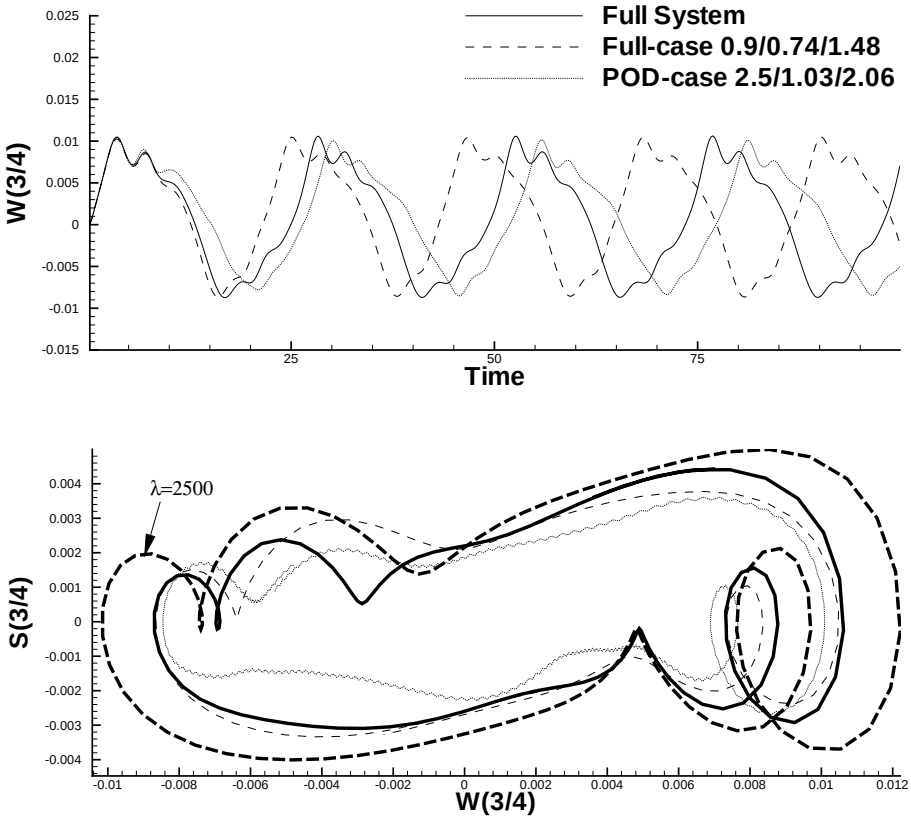



Figure 5.9 LCO for $\lambda = 2000$ (trained at $\lambda = 2500$)

at the new value of $\lambda$. The extendability of POD/ROM/DD across this large change in parameters using this simple, sub-optimal approach demonstrates the potential for future analysis of flows with moving shocks using this implementation.

### *5.4 Conclusions*

A reduced order modeling approach was successfully applied to a transonic aeroelastic problem with large motions of a strong shock. Domain decomposition enabled the use of POD/ROM for this case. The non-Galerkin solver used a constrained optimization approach to link internal boundaries. Time accurate results were obtained using different time steps for each domain, and the reduced order fluid model was coupled to the panel dynamics to provide an accurate reduced order aeroelastic system.

An 88% reduction in DOFs resulted a 66% computational saving for implementations using the full system model in the region of the flow field containing the shock. When POD/ROM was used in the shock region, there was almost three orders of magnitude reduction in DOFs (from 64400 to 125). Both cases yielded accurate panel LCO, and replicated correct motion of the transonic shock. In addition, both cases accurately modeled transonic LCO states corresponding to parameter values not used for POD training.

This robustness across a parameter space demonstrated the potential of POD/ROM for analysis of transonic LCO. The ultimate goal of such analysis is to fully characterize the LCO branches within a parameter space, especially trying to identify regions of supercritical LCO where LCO states can exist at parameter levels lower than those observed in the gradual onset of LCO. At supercritical parameter values, both static and LCO solutions are possible depending on the initial condition, with larger disturbances generally exciting the LCO state. While linear analysis can be used to predict LCO onset values, it cannot determine the existence of supercritical LCO states. Analysis with POD/ROM may eventually be able to identify these LCO states, and help engineers adjust parameters to ensure no LCO anywhere within the performance envelope of the aeroelastic system.

# VI. Summary

## 6.1 Conclusions

The ability of POD to produce reduced order models for inviscid fluid problems with moving strong shocks has been thoroughly explored. Since POD is a linear method, the use of POD/ROM is easily likened to other reduced order techniques that require linearization of the flow field about some known solution. The moving shock problem is untenable for these techniques because any motion of the shock requires re-linearization of the problem, mitigating any computational savings. POD/ROM is different because it uses the POD basis to identify a linear space that approximates the domain and range of the Euler equations. This linear space is very low-dimensional, and the reduced order advantage comes from recasting the full systems dynamics to solve for the coordinates of the flow field projected into this linear space. This projection is the linear operation, and if the linear space contains the proper discontinuous functions the shock motion is preserved.

While theoretically valid, the identification of a proper POD basis for the moving shock case is problematic in practice. Projection into a linear space approximates the flow field with a linear combination of the POD basis. When added together, it is impossible for a small number of basis functions to produce a discontinuity at some location unless one or more of the basis functions contain discontinuities at that location. For POD/ROM, this principal means that a POD basis must contain vectors with discontinuous jumps at every shock location to enable proper shock motion. Not only shock position, but shock strength must be accurately reflected in the POD basis. The POD basis is obtained using the method of snapshots, which takes samples of the full system dynamics and identifies a minimal set of basis vectors using singular value decomposition. When using the method of snapshots, samples must be taken at every shock location (and shock strength) of interest to produce a POD basis that tracks discontinuities correctly. Essentially, POD/ROM can be used to replicate shock behavior seen a priori in the full system, but it cannot introduce any new shock motion. This proves useful for some applications where periodic shock motion is not greatly affected across some parameter space of interest. For other applications where shock motion is greatly affected across the parameter space, this restriction is unsatisfactory. In

addition, a large region of shock motion requires a great many modes, while other portions of the flow field only require a few. This spatial disparity in the fluid dynamics introduces robustness problems that can lead to numerical failure unless the snapshots and modes are very carefully selected. These problems motivated a new methodology for employing POD/ROM with moving shocks.

*6.1.1 Domain Decomposition.* Domain decomposition was introduced to increase the robustness of POD/ROM in the presence of moving shocks. The spatial solution domain was divided into regions to isolate the region containing the moving shock. Snapshots of the entire solution domain were collected from the full system, but POD/ROMs for each domain section were determined independently using the portion of the snapshots applying to that section. The POD basis for the shock region consisted of many discontinuous modes to replicate the shock motion. In regions without shocks, the POD basis had fewer modes with structures to match the more benign dynamics. Consequently the reduced order model was more stable because the flow field was projected into several linear spaces (one for each domain section) specially designed for the local dynamics.

For problems where the shock motion could not be adequately sampled, the full system model was used in the shock region. The use of POD in non-shocked portions of the flow still provided a significant reduction in both DOFs and computational expense. The full system model in the shock region could produce any shock motion, regardless of the POD training used for other domain sections in the flow. This ROM was robust across a very large parameter space.

When the domain was decomposed, the fluid problem was divided into a number of smaller problems linked by internal boundaries. An optimization-based methodology was used to ensure smoothness between domains at the internal boundaries. This technique involved overlapping the domain sections and constraining either the $L^1$ or $L^2$ error norm in the overlapped regions. Lagrange constrained optimization was used to enforce this condition. $L^1$ constraints that adequately linked the internal boundaries were difficult to identify, but once identified the solver implementation was very efficient. Identification of $L^2$ constraints was straight forward, but the implementation resulted in a less efficient

solver. Overlapping the domains in a region of strong nonlinearity destabilized the full system model. However, both $L^1$ and $L^2$ constraints worked well everywhere else in the domain.

Domain decomposition also allowed regional time stepping to improve the efficiency of the time accurate ROM. Outlying domain sections generally contained benign dynamics, yet the time accurate solution was tied to a very small time step related to the dynamics in the shock region. An implicit solver was used for the outlying regions, allowing for a very large time step relative to either the full or reduced order solver used in the shock region. This provided an additional boost in computational performance for the time accurate case.

*6.1.2 Applications.* The POD basis was incorporated into both explicit and implicit solvers using the subspace projection method. This method relied on the full system dynamics to produce either time accurate, or steady state POD/ROMs for each domain section. The POD/ROMs for each domain section were linked with the constrained optimization technique to form a POD/ROM/DD for several applications. These application demonstrated how POD could be accurately applied to the moving shock problem.

*6.1.2.1 Quasi-Steady Shock Motion.* The accuracy and order reduction of the domain decomposition approach was first demonstrated for a quasi 1-D nozzle flow. These POD/ROMs were trained using $\gamma = 1.4$. The accuracy was examined for quasi-steady shock motion as $\gamma$ was varied from $1.4 \rightarrow 1.37$. Flow field errors were less than 2%, and the shock movement was tracked within one grid point of the true shock location. The full order solution had 750 DOFs, the POD/ROM/DD with a full order shock region had 58 DOFs, and the POD/ROM/DD with a POD/ROM for the shock region had 55 DOFs. The 1-D analysis is contained in Chapter 3.

Next, the approach was extended to 2-D. This analysis is contained in Chapter 4. A very accurate and efficient reduced order model was generated for the 2-D blunt body problem using POD/ROM/DD to isolate the region of the flow field containing the shock motion. This region was solved at full order, while the remainder of the domain was

treated with POD/ROM. The resulting ROM had a 4.3 to 1 reduction in DOFs and a computational improvement of 4.7 to 1. The POD/ROM/DD provided accuracy within 5% over a parameter space of inlet Mach number and AoA that extended far beyond the parameter space used for POD training. In contrast, the use of POD/ROM/DD with a reduced order shock region was not an advantageous approach for the 2-D blunt body problem. Full order explicit time integration for shock region was over 8 times faster than the implicit reduced order calculations for this problem.

*6.1.2.2 Unsteady Shock Motion.* The reduced order modeling approach was successfully applied to a transonic aeroelastic problem with large motions of a strong shock. This analysis is contained in Chapter 5. Time accurate results were obtained using different time steps for each domain, and the reduced order fluid model was coupled to the panel dynamics to provide an accurate reduced order aeroelastic system. POD/ROM/DD yielded accurate panel LCO, and replicated correct motion of the transonic shock. In addition, both full system and POD/ROMs for the shock region accurately modeled all the transonic LCO states, even though the POD/ROMs from all domain sections were only trained for a single state. This robustness across a parameter space demonstrated the potential of POD/ROM for transonic design in aeroelastic problems. An 88% reduction in DOFs resulted a 66% computational saving for implementations using the full system model in the region of the flow field containing the shock. When POD/ROM was used in the shock region, there was almost three orders of magnitude reduction in DOFs (from 64400 to 125).

## *6.2 Significant Advances*

The course of this research has provided several advances to the field of reduced order fluid modeling. They include the extension of POD/ROM to the moving shock case, the addition of domain decomposition to the use of POD/ROM, the extension of optimization-based solvers to the field of reduced order modeling, and the application of reduced order modeling to a transonic aeroelastic problem. These advances are delineated as follows:

*6.2.1 Shock Capturing with POD.* Prior to this research, all applications of POD to high-speed compressible flow problems involved stationary shock waves. The extension of POD application and theory to moving shocks is a useful advancement. The characteristics of an adequate POD basis for the moving shock problem, and the insights into POD training to obtain such a basis are a key contribution.

*6.2.2 Domain Decomposition.* Decomposing the flow field into sections, and applying POD/ROM independently to each section is a new innovation. This was a logical extension of domain decomposition currently in use for large full system models. In the later case, domain decomposition facilitates computational efficiency through parallel computing. While parallel computing could also be used with the POD/ROM/DD, this research focused on domain decomposition to improve solver robustness. Computational performance for time accurate POD/ROM was improved with domain decomposition because it enabled regional time stepping. The application of regional time stepping to the reduced order problem is an additional advancement.

*6.2.3 Optimization-Based Solvers.* Optimization solvers based on Lagrange constraints have emerged in the past two years for domain decomposition of full systems models. The application of this approach to domain decomposition of reduced order modeling is a new innovation. This research produced specific analysis, implementation, and quantified results for the use of both $L^1$ and $L^2$ constraints. This contribution is essential when using domain decomposition with POD/ROM. Explicit and implicit optimization-based solver implementations were developed for both time accurate and steady state applications.

*6.2.4 Transonic Aeroelastic Panel Response.* The innovations listed above enabled in the application of reduced order modeling to a transonic aeroelastic problem. Previous research with POD/ROM avoided the transonic regime due to the presence of strong moving shocks in the flow field. Until now, no reduced order modeling method has been successfully applied to this problem.

### *6.3 Summary and Future Work*

The thesis statement for this dissertation claimed that "*domain decomposition would enable the use of POD to generate reasonable low order approximations of* 2*-D high speed fluid flows with shock movement over confined regions.*" This thesis has been thoroughly validated by the development, analysis, implementation, and applications contained herein.

Future research is needed to realize the demonstrated potential for nearly three orders of magnitude reduction in compute time. This gain in computational performance was not realized in this research, because the non-Galerkin solver implementation was not faster than the full-system implementation in the reduced order shock region. Instead, computational gains of about one order of magnitude accompanied three orders of magnitude in DOF reduction. Although the explicit, non-Galerkin solver was not efficient, it did demonstrate the existence and adequacy of POD based fluid modes to model an unsteady shock.

An implementation methodology using Galerkin projection could provide the additional improvements in computational performance. Galerkin projection is a method to identify a set of nonlinear ODEs that model the dynamics of the reduced order variable. Instead of using the full system dynamics, this much smaller set of ODEs could be solved in its place to produce a truly reduced order system. To date, the Galerkin approach with POD has only addressed incompressible viscous problems with periodic boundaries. An extension to incompressible flow with nonlinear boundaries is necessary for use with the POD/ROM/DD. The viscous incompressible flow applications avoids nonlinear boundary conditions since the velocity component are zero along solid surfaces, and neither density nor energy are considered. The inclusion of density and energy, along with nonlinear, coupled internal boundaries introduce difficulties that must be solved before the Galerkin method can be used with POD/ROM/DD. Coupled nonlinear solid boundaries such as the elastic panel equations must also be addressed.

This research represents a significant advancement toward integrating POD/ROM into optimization algorithms with application to transonic aeroelastic design problems. Once a more efficient reduced order implementation is identified, analysis with POD/ROM

can go beyond demonstration. Transonic flutter analysis needs to be explored much more fully than the handful of cases reported in the literature. Nonlinear optimization with POD/ROM could identify key design parameters for aeroelastic structures.

Identification of an efficient method to couple nonlinear boundary conditions into a reduced order fluid model would be a great advancement for control applications with POD/ROM as well. The control surface deflections for many flow control problems come from solid boundaries, and a reduced order set of dynamics coupling the flow field and the control deflections could extend flow control applications beyond low-speed incompressible flow, to compressible high-speed flows. Once a POD/ROM implementation is identified with computing efficiency to match the order reduction seen in this research, design optimization and control for high-speed flows could become a practical endeavor.

*Bibliography*


1. Romanowski, M .C. and Dowell, E. H. "Reduced Order Euler Equations for Unsteady Aerodynamic Flows: Numerical Techniques." *AIAA 96-0528, 34th Aerospace Sciences Meeting and Exhibit, Reno NV*. January 1996.

2. Dowell, E. H. "Eigenmode Analysis in Unsteady Aerodynamics: Reduced-Order Analysis," *AIAA Journal*, *34, n8*:1578–1583 August 1996.

3. Beran, P. S., Huttsell, L. J., Buxton, B. J., Noll, C. and Osswald, G. "Computational Aeroelastic Techniques for Viscous Flow." *CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics, Williamsburg, VA*. June 22-25, 1999.

4. Hall, K. C., Thomas, J. P. and Dowell, E. H. "Reduced-Order Modeling of Unsteady Small-Disturbance Flows Using a Frequency-Domain Proper Orthogonal Decomposition Technique." *AIAA 99-0655, 37th Aerospace Sciences Meeting and Exhibit, Reno, NV*. January 11-15, 1999.

5. Dowell, E. H., Hall, K. C., Thomas, J.P., Florea, R., Epureanu, B.I. and Heeg, J. "Reduced Order Models in Unsteady Aerodynamics." *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, St. Louis, MO*. April 12-15, 1999.

6. Florea, R., Hall, K. C. and Dowell, E. H. "Eigenmode Analysis and Reduced Order Modeling of Unsteady Transonic Full Potential Flow Around Isolated Airfoils." *CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics, Williamsburg, VA*. June 22-25, 1999.

7. Pettit, C. L. and Beran, P. S. "Reduced-Order Modeling for Flutter Prediction." *AIAA 2000-1446-CP, CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics, Williamsburg, VA*. April 2000.

8. Mortara, S. A., Slater, J. C. and Beran, P. S. "An Optimal Proper Orthogonal Decomposition Technique for the Computation of Nonlinear Panel Flutter." *AIAA 2000-1936*. 2000.

9. Beran, P. and Pettit, C. "Prediction of Nonlinear Panel Response Using Proper Orthogonal Decomposition." *AIAA 2001-1292, 42nd Structural, Structural Dynamics, and Material Conference, Denver, CO*. April 2001.

10. LeGresley, P. A. and Alonso, J. J. "Airfoil Design Optimization Using Reduced Order Models Based on Proper Orthogonal Decomposition." *AIAA 2000-2545, Fluids 2000 Conference and Exhibit, Denver, CO*. June 2000.

11. Ito, K. and Ravindran, S. S. "A Reduced-Order Method for Simulation and Control of Fluid Flows," *Journal of Computational Physics*, *143*:403–425 1998.

12. Shvartsman, S. Y. and Kevrekidis, I. G. "Nonlinear Model Reduction for Control of Distributed Systems: a Computer-Assisted Study," *American Institute of Chemical Engineering*, *44*(7):1579–1595 1998.

13. Banerjee, S., Cole, J. V. and Jensen, K. F. "Nonlinear Model Reduction Strategies for Rapid Thermal Processing Systems," *IEEE Transactions on Semiconductor Manufacturing*, *11*(2):266–275 1998.

14. Kunisch, K. and Volkwein, S. "Control of the Burgers Equation by a Reduced-Order Approach Using Proper Orthogonal Decomposition," *Journal of Optimization Theory and Applications*, *102*(2):345–371 1999.

15. Rediniotis, O. K., Ko, J., Yue, X. and Kurdila, A. J. "Synthetic Jets, their Reduced Order Modeling and Applications to Flow Control." *AIAA 99-1000, 37th Aerospace Sciences Meeting and Exhibit, Reno, NV*. January 11-15, 1999.

16. Park, H. M. and Lee, M. W. "An Efficient Method of Solving the Navier-Stokes Equation for Flow Control," *International Journal for Numerical Methods in Engineering*, *v41*:1133–1151 1998.

17. Sirovich, L. "Turbulence and the Dynamics of Coherent Structures. Part 1: Coherent Structures," *Quarterly of Applied Mathematics*, *45, n3*:561–571 October 1987.

18. Stakgold, Ivar. *Green's Functions and Boundary Value Problems* (Second Edition). Washington DC: John Wiley and Sons, Inc., 1998.

19. Holmes, P., Lumley, J.L. and Berkooz, G. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, 1996.

20. Canuto, C., Hussaini, M.Y., Quarteroni, A. and Zang, T.A. *Spectral Methods in Fluid Dynamics*. Berlin: Springer-Verlag, 1988.

21. Beran, P. and Silva, W. "Reduced Order Modeling: New Approaches for Computational Physics." *AIAA 2001-0853, 39th Aerospace Sciences Meeting and Exhibit, Reno, NV*. January 2001.

22. Karhunen, K. *Zur Spektral Theorie Stochastischer Prozesse*. Ann. Acad. Sci. Fennicae, Ser, 1946.

23. Loeve, M. *Functions de Second Ordre*. Paris, France: C. R. Academie des Sciences, 1945.

24. Lumley, J. L. "The Structure of Inhomogeneous Turbulence." *Proceedings of the International Colloquium on the Fine Scale Structure of the Atmosphere and its Influence on Radio Wave Propogation*. 166–178. Moscow: Dokl. Akad. Nauk SSSR, 1967.

25. Moin, P. "Probing Turbulence via Large Eddy Simulation," *AIAA Paper 84-0174* 1984.

26. Sirovich, L. and Rodriguez, J. D. "Coherent Structures and Chaos: A Model Program," *Physics Letters*, *v120, n5*:211–214 February 1987.

27. Chambers, D. H., Adrian, R. J., Moin, P., Stewart, D. S. and Sung, H. J. "Karhunen-Loeve Expansion of Burger's Model of Turbulence," *Physics of Fluids*, *31, n9*:2573–2582 1988.

28. Rodriguez, J. D. and Sirovich, L. "Low-Dimensional Dynamics for the Complex Ginzburg-Landau Equation," *Physica*, *v43*:77–86 1990.

29. Webber, G. A., Handler, R. A. and Sirovich, L. "The Karhunen-Loeve Decomposition of Minimal Channel Flow," *Physics of Fluids*, *v9, n4*:1054–1066 April 1997.

30. Mahajan, A. J., Bakhle, M. A. and Dowell, E. H. "A New Method for Aeroelastic Stability Analysis of Cascades Using Nonlinear, Time Marching CFD Solvers," *5th Symposium on Multidisciplinary Analysis and Optimization (AIAA 94-4396)* 1994.

31. Kurdila, A.J., Carrol, B., Nishida, T. and Sheplak, M. "Reduced-Order Modeling for Low Reynolds Number Flow Control." *SPIE Conference on Mathematics and Control in Smart Structures, Newport Beach, CA3667*. 68–79. March 1999.

32. Hall, K .C. "Eigenanalysis of Unsteady Flows About Airfoils, Cascades, and Wings," *AIAA Journal*, *32, n12*:2426–2432 December 1994.

33. Glowinski, R. and Pironneau, O. "On a finite element approximation of the Stokes problem," *Numerical Mathematics*, *33*:397–424 1979.

34. Gunzburger, M. D. and Nicolaides, R. "Issues in the implementation of substructing algorithms for the Navier-Stokes equations," *Applied Numerical Mathematics*, *2*:243–256 1986.

35. Cahouet, J. "On some difficulties occuring in the simulation of incompressible fluid flows by domain decomposition methods." *Domain Decomposition in Scientific and Engineering Computing, SIAM, Philadelphia, PA*. 1988.

36. Fortin, M. and Aboulaich, R. "Schwarz's decomposition method for incompressible flow problems." *Proceedings of the First International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, PA*. 333–349. 1988.

37. Gunzburger, M. D. and Nicolaides, R. "Issues in the implementation of substructing algorithms for the Navier-Stokes equations." *Advances in Computer Methods for Partial Differential Equations, IMACS,New Brunswick, NJ*. 57–63. 1988.

38. Carlenzoli, C., Quarteroni, A. and Valli, A. "Spectral domain decomposition methods for compressible Navier-Stokes equations." *Proceedings of the Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, PA*. 441–450. 1992.

39. Azaiez, M. and Quateroni, A. "A special Stokes solver in domain decomposition methods." *Domain Decomposition in Scientific and Engineering Computing, AMS, Providence, RI*. 151–156. 1994.

40. J.Lions and Pironneau, O. "Non Overlapping Domain Decomposition for Evolution Operators," *Comptes rendus de l'Academie des sciences. Serie I. Mathematique*, *330*(10):943–951 April 10, 2000.

41. J.Lions and Pironneau, O. "Overlapping Domain Decomposition for Evolution Operators," *Comptes rendus de l'Academie des sciences. Serie I. Mathematique*, *330*(10):937–943 April 10, 2000.

42. Gunzburger, M. D. and Lee, H. K. "An Optimization-Based Domain Decomposition Method for the Navier-Stokes Equations," *Society for Industrial and Applied Mathematics (SIAM) Journal of Numerical Analysis*, *37*(5):1455–1480 May 4, 2000.

43. Du, Q. and Gunzburger, M. D. "A Gradent Approach to Optimization-Based Multidisciplinary Simulations and Nonoverlapping Domain Decomposition Algorithms," *Society for Industrial and Applied Mathematics (SIAM) Journal of Numerical Analysis*, *37*(5):1513–1541 May 4, 2000.

44. Farhat, C., Macedo, A., Lesoinne, M., Roux, F., Magoules, F. and de La Bourdonnaie, A. "Two-level domain decomposition methods with Lagrange multipliers for the fast iterative solution of acoustic scattering problems," *Computer Methods in Applied Mechanics and Engineering*, *184*(2):213–241 2000.

45. Tannehill, John C., Anderson, Dale A. and Pletcher, Richard H. *Computational Fluid Mechanics and Heat Transfer*. Washington DC: Hemishpere Publishing Company, 1997.

46. Janus, J.M. "The Development Of A Three-Dimensional Split Flux Vector Euler Solver With Dynamic Grid Applications," *Masters Thesis* 1984.

47. Sankar, L. N., Ruo, S. Y. and Malone, J. B. "Application of Surface Transpiration in Computational Aerodynamics." *AIAA 86-0511, AIAA 24th Aerospace Sciences Meeting, Reno, NV*. January 1986.

48. Antonnen, J. S. R. *Techniques for Reduced Order Modeling of Aeroelastic Structures*. PhD dissertation, Air Force Institute of Technology, School of Engineering and Management, 2001.

49. Naylor, Arch W. and Sell, George R. *Linear Operator Theory in Engineering and Science*. New York NY: Springer-Verlag, Inc., 1982.

50. Park, H. and Lee, M. "Control of Navier-Stokes Equations by Means of Mode Reduction," *International Journal Numerical Methods in Fluids*, *33*:535–557 2000.

51. Strang, Gilbert. *Introduction to Applied Mathematics*. Cambridge MA: Wellesley Cambridge Press, 1986.

52. Shubin, G.R., Stephens, A.B. and Glaz, H.M. "Steady Shock Tracking and Newton's Method Applied to One-Dimensional Duct Flow," *Journal of Computational Physics*, *39*:364–374 1981.

53. Selvam, R. P. and Morton, S. A. "Computation of Nonlinear Viscous Panel Flutter." *AIAA 2001-1292*. 1998.

54. Hurka, J. and Ballmann, J. "Elastic Panels in Transonic Flow." *AIAA 2001-2722, 19th AIAA Computational Fluid Dynamics Conference, Anaheim, CA*. June 2001.

55. Yu, H. "Solving parabolic problems with different time steps in different regions in space based domain decomposition methods," *Applied Numerical Mathematics: Transactions of IMACS*, *30*(4):475 1999.

56. Gordnier, R. E. and Visbal, M. R. "Development of a Three-Dimensional Viscous Aeroelastic Solver for Nonlinear Panel Flutter." *AIAA 2000-2337*. June 2000.

57. Bendiksen, O. O. and Davis, G. A. "Nonlinear Traveling Wave Flutter of Panels in Transonic Flow." *AIAA 1995-1486*. 1995.

58. Adams, Robert A. *Sobolev Spaces*. Orlando FL: Academic Press, Inc., 1975.

59. Maybeck, Peter S. *Stochastic Estimation and Control*, *One*. Arlington VA: Navtech Book and Software Store, 1994.

60. Joshi, S.S, Speyer, J.L. and Kim, J. "Finite Dimensional Optimal Control of Poiseuille Flow," *Journal or Guidance, Control and Dynamics*, *22*(2):340–348 1999.

61. Cortelezzi, L., Leonard, A. and Doyle, J.C. "An example of active circulation control of the unsteady separated flow past a semi-infinite plate," *Journal of Fluid Mechanics*, *260*:127–154 1994.

62. Coller, B. D., Holmes, P. and Lumley, J. L. "Control of bursting in boundary layer models," *Applied Mechanics Review*, *6*(2):S139–S143 1994.

63. Delville, J., Cordier, L. and Bennet, J. "Large-scale-structure identification and control in turbulent shear flows." *Flow Control: Fundamentals and Practices*. 199–273. Springer, 1998.

64. Prabhu, R. D., Collis, S. S. and Chang, Y. "The influence of control on proper orthogonal decomposition of wall-boudned turbulent flows," *Physics of Fluids*, *13*(2):520–537 2001.

65. Sutton, G.P. *Rocket Propulsion Elements*. John Wiley and Sons, 1992.

66. Anderson, J.D. *Fundamentals of Aerodynamics* (Second Edition). New York: McGraw-Hill, 1991.

67. Hill, P.G. and Peterson, C.R. *Mechanics and Thermodynamics of Propulsion*. Addison-Wesley Publishing Company, 1965.

68. Huzel, D.K. and Huang, D.H. *Modern Engineering for Design of Liquid-Propellant Rocket Engines*. AIAA, 1992.

69. Anderson, J.D. *Computational Fluid Dynamics*. McGraw-Hill, Inc, 1995.

70. Lucia, D. J., King, P. I., Beran, P. S. and Oxley, M. E. "Reduced Order Modeling for a One-Dimensional Nozzle Flow with Moving Shocks." *AIAA 2001-2602, 19th AIAA Computational Fluid Dynamics Conference, Anaheim, CA*. June 2001.

71. Antonnen, J. S. R., King, P. I. and Beran, P. S. "The Accuracy of POD-Bases Reduced-Order Models with Deforming Grids." *AIAA 2001-2541, 19th AIAA Computational Fluid Dynamics Conference, Anaheim, CA*. June 2001.

72. Lindsey, W. F. and Daley, B. N. "Effects of Compressibility on the Flow Past a Two-Dimensional Bump." *Technical Note 2484, National Advisory Committee for Aeronautics*. 1952.

73. Kaplan, C. "The Flow of a Compressible Fluid Past a Curved Surface." *Report 768, National Advisory Committee for Aeronautics*. 1944.

74. Anderson, J. D. *Hypersonic and High Temperature Gas Dynamics*. McGraw-Hill, 1989.

## *Appendix A. Numerical Analysis*

This appendix develops the theory for reduced order modeling of numerical fluid solvers used in this research. Early in the discussion, operations on an infinite dimensional function space are approximated by operations on a large, but finite dimensional vector space through spatial discretization of the domain. The governing equations for inviscid fluid flow are formulated as an evolutionary operator, and the method of snapshots is developed in this context. Linear operator theory and vector space analysis methods are used to derive the reduced order mapping that is a characteristic of POD/ROM. The spectral theory of linear operators is used to derive the optimal basis for the reduced order space identified by the method of snapshots. The subspace projection approach for obtaining the reduced order variables is derived.

### *A.1 Mathematical Foundations*

The governing equations for inviscid flow are known as the Euler equations. For unsteady two-dimensional flow, the Euler equations (in conservation form) are shown below.

$$\frac{\delta\rho}{\delta t} + \frac{\delta(\rho u)}{\delta x} + \frac{\delta(\rho v)}{\delta y} = 0 \tag{A.1}$$

$$\frac{\delta(\rho u)}{\delta t} + \frac{\delta(\rho u^2 + P)}{\delta x} + \frac{\delta(\rho u v)}{\delta y} = 0 \tag{A.2}$$

$$\frac{\delta(\rho u)}{\delta t} + \frac{\delta(\rho u v)}{\delta x} + \frac{\delta(\rho v^2 + P)}{\delta x} = 0 \tag{A.3}$$

$$\frac{\delta E_t}{\delta t} + \frac{\delta(E_t + P)u}{\delta x} + \frac{\delta(E_t + P)v}{\delta y} = 0 \tag{A.4}$$

Here, $\rho$ is density, $u$ is $x$-direction velocity, $v$ is $y$-direction velocity, $P$ is pressure, and $E_t$ is total energy. The fluid variables can be renamed as follows,

$$u_1(x, y, t) = \rho(x, y, t) \tag{A.5}$$

$$u_2(x, y, t) = \rho(x, y, t) \cdot u(x, y, t) \tag{A.6}$$

$$u_3(x, y, t) = \rho(x, y, t) \cdot v(x, y, t) \tag{A.7}$$

$$u_4(x, y, t) = E_t(x, y, t) \ . \tag{A.8}$$

The isentropic flow equations are used to recast (A.1) through (A.4) using (A.5) through (A.8). Here $\gamma$ is the ratio of specific heats for air, and $e$ is the internal energy per unit mass.

$$P = \gamma \rho e \tag{A.9}$$

$$e = \frac{E_t}{\rho} - \frac{1}{2}u^2 \tag{A.10}$$

Equations (A.1) through (A.4) can then be rewritten,

$$\dot{u_1} = R_1(u_1, u_2, u_3, u_4) \tag{A.11}$$

$$\dot{u_2} = R_2(u_1, u_2, u_3, u_4) \tag{A.12}$$

$$\dot{u_3} = R_3(u_1, u_2, u_3, u_4) \tag{A.13}$$

$$\dot{u_4} = R_4(u_1, u_2, u_3, u_4) \tag{A.14}$$

Equations (A.11) through (A.14) are a coupled set of four nonlinear partial differential equations (PDEs) in two spatial dimensions $(x, y)$, and in time $(t)$. The term $\dot{u}$ denotes the time derivatives which are segregated from the spatial derivatives. The spatial derivatives are nonlinear, and they are represented by the nonlinear differential operator $R$. Since the time derivative is a linear operator, this system can be recast as a time evolution of spatial functions. An evolution operator is derived by integrating in time from a given initial condition.

$$u_1(x, y, t + \Delta t) = u_1(x, y, t) + \int_t^{t+\Delta t} R_1(u_1, u_2, u_3, u_4)dt \tag{A.15}$$

$$u_2(x, y, t + \Delta t) = u_2(x, y, t) + \int_t^{t+\Delta t} R_2(u_1, u_2, u_3, u_4)dt \tag{A.16}$$

$$u_3(x, y, t + \Delta t) = u_3(x, y, t) + \int_t^{t+\Delta t} R_3(u_1, u_2, u_3, u_4)dt \tag{A.17}$$

$$u_4(x, y, t + \Delta t) = u_4(x, y, t) + \int_t^{t+\Delta t} R_4(u_1, u_2, u_3, u_4)dt \ , \tag{A.18}$$

which for small $\Delta t$, can be approximated in a number of ways including,

$$\begin{aligned}
u_1(x,y,t+\Delta t) &\cong u_1(x,y,t) + \Delta t R_1(u_1,u_2,u_3,u_4) \\
u_2(x,y,t+\Delta t) &\cong u_2(x,y,t) + \Delta t R_2(u_1,u_2,u_3,u_4) \\
u_3(x,y,t+\Delta t) &\cong u_3(x,y,t) + \Delta t R_3(u_1,u_2,u_3,u_4) \\
u_4(x,y,t+\Delta t) &\cong u_4(x,y,t) + \Delta t R_4(u_1,u_2,u_3,u_4)
\end{aligned} \tag{A.19}$$

$\Delta t$ is a fixed time step, and this approximate equation set can be formulated as a single state transition equation. A four-tuple set of initial spatial functions,

$$U(x,t) = (u_1(x,t), u_2(x,t), u_3(x,t), u_4(x,t)) \tag{A.20}$$

evolve over time by repeated application of a nonlinear differential operator $R$, which is shown in the form of a state transition function $K$ in equation (A.22),

$$U(x,t+\Delta t) = U(x,t) + \Delta t R(U(x,t)) \tag{A.21}$$

$$U(x,t+\Delta t) = K(U(x,t)) \ . \tag{A.22}$$

Notice that the spatial variables $x$ and $y$ have been generalized to a single spatial variable $x$ for notational convenience. The variable $x$ applies to the spatial component of either 1-D, 2-D or 3-D problems. $K$ can now be considered as an operator on functions of $x$ only, and the following notation change reflects this,

$$u_1(x,y,t_n) \Rightarrow \psi_n^1(x) \tag{A.23}$$

$$u_2(x,y,t_n) \Rightarrow \psi_n^2(x) \tag{A.24}$$

$$u_3(x,y,t_n) \Rightarrow \psi_n^3(x) \tag{A.25}$$

$$u_4(x,y,t_n) \Rightarrow \psi_n^4(x) \ . \tag{A.26}$$

The subscript for $\psi$ refers to the sequence of progression from input functions at $n$ to output functions at $n+1$ in regard to a single application of the nonlinear operator $K$. The superscript on $\psi$ denotes which of the fluid variables is represented. Consider a single

application of $K$, to the ordered four-tuple $\{\psi_n^1, \psi_n^2, \psi_n^3, \psi_n^4\}$. The domain and range of $K$ will come from the set of all functions of $x$ with sufficient smoothness such that the spatial derivatives in (A.11) through (A.14) are defined. In addition, these equations are ill-posed without sufficient spatial boundary conditions to produce a unique solution. The domain of $K$ is further restriction to only include functions meeting the specified boundary conditions for $U(x,t)$. In general, the domain and range of $K$ do not form a linear space (49, 58, 19). Instead, they form an operator manifold $X$.

Repeated application of $K$ will take any initial set of functions $(\psi_0^1, \psi_0^2, \psi_0^3, \psi_0^4)$ and propagate it along a path through the four-tuple operator manifold $(X_1 \times X_2 \times X_3 \times X_4)$. If there is a steady state solution, repeated applications of $K$ will ultimately converge to these functions and any additional applications of $K$ will not change the result.

In many cases, the Euler equations as depicted in (A.1) through (A.4) will yield a steady state solution. Consider Figure A.1 for the path of $\psi_n^1 \in X_1$ as $n : 0 \rightarrow n_{ss}$ where $n_{ss}$ denotes the increment in the sequence after which steady state is achieved. When the

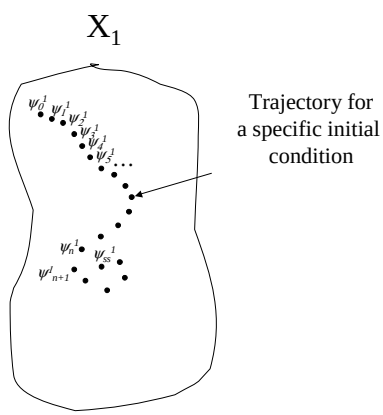



Figure A.1 Notional Trajectory within the Operator Manifold

boundary conditions are held fixed at values that result in steady state flow fields, the trajectories of $(\psi_n^1, \psi_n^2, \psi_n^3, \psi_n^4)$ will reach the same steady state from any initial condition that satisfies the boundary conditions. Any element of $(X_1 \times X_2 \times X_3 \times X_4)$ is a candidate outcome of the operator $K$ with few exceptions. Once the infinite function space $X$ is spatially discretized into a finite dimensional vector space $E_N$, these exceptions will no

longer be a concern, so both $R_K$ (the range of $K$) and $D_K$ (the domain of $K$) consist of all functions contained in the operational manifold $(X_1 \times X_2 \times X_3 \times X_4)$.

### *A.2 Foundations for Reduced Order Modeling*

*A.2.1 Overview.* The reduced order modeling goal is to recreate the flow field evolution and/or the steady state solution without the computational burden of the full system solver. For 2-D and 3-D viscous flow fields, the computational load for solutions can mandate days of computer time in a high performance computing environment to obtain a single solution. This section will develop a methodology to reduce the spatial degrees of freedom, and increase the time integration efficiency for obtaining approximate solutions to equations (A.11) through (A.14). This methodology exploits a data eigenvalue technique known as Proper Orthogonal Decomposition (POD) for spatial order reduction. POD identifies fluid solution modes and the nonlinear operator $K$ is recast to solve for modal coefficients which are functions of time. This Reduced Order Model (ROM) then replaces the full system model for the computation of flow variables.

The flow field solution is approximated by any of a number of numerical techniques. Numerical flow solvers discretize the spatial domain resulting in a vary large number of DOFs. Each DOF corresponds to a dimension in Euclidean "$N$-space." The main product of the POD process is a mapping between the low and high dimensional outcomes. Here "dimension" is interchanged with "degree of freedom," and the full system numerical flow solver yields the high dimensional outcomes. The mapping is used to transform the high dimensional problem into a low dimensional problem in Euclidean "$M$-space" where $M << N$. The low dimensional problem is then solved quickly, and transformed back to the high dimensional solution.

The first step in the POD process is to approximate the operator manifold $X$ with a linear space $\ell$. This problem requires four-tuples $(\ell_1 \times \ell_2 \times \ell_3 \times \ell_4)$ to approximate $(X_1 \times X_2 \times X_3 \times X_4)$. The linear manifold must contain every flow vector $(\psi_n^1, \psi_n^2, \psi_n^3, \psi_n^4)$ needed to approximate the flow solutions of interest. Errors are introduced to the flow solution vector when $\psi_n \notin \ell$. Once the linear manifold is identified, the isomorphism between the linear manifold $\ell$ and the space of column vectors in $E_Q$ is exploited, where

$Q$ is the dimension of $\ell$. The elements of the isomorphic spaces $(E_Q^1 \times E_Q^2 \times E_Q^3 \times E_Q^4)$ are the coordinates of each $N$-dimensional fluid variable in terms of an appropriate basis.

Coordinate generation for the elements of $\ell_i$ (here $i$ denotes one of the four fluid variables) produces a non-symmetric linear operator from $\ell_i$ to $E_Q^i$. The pseudo inverse $S$ is used to obtain an eigenvalue decomposition of the elements of $E_Q^i$ using Singular Value Decomposition (SVD). The eigenvectors from the SVD of $S$ form an orthonormal basis for $E_Q^i$, and transforming $E_Q^i$ to this basis enables an optimal truncation of the degrees of freedom in $E_Q^i$ that contribute the least to the $L_2$ norm of the elements. The truncated vector spaces for each fluid variable do not need to have the same dimension, and the Euclidean four-tuple $(E_{M(1)}^1 \times E_{M(2)}^2 \times E_{M(3)}^3 \times E_{M(4)}^4)$, where $M(i) < Q$, is the reduced order space for the POD/ROM. In summary, POD approximates the domain and range of $K$ with a linear space of four-tuples from Euclidean $Q$-space, and reduces this further to a four-tuple of $M(i)$ dimensional vector spaces via SVD and truncation.

*A.2.2 POD/ROM Applied to a Fluid Problem.* Now we specifically develop this methodology for the 2-D Euler equations. With a full system model spatially discretized to $N$ degrees of freedom for operation on $E_N$, the POD/ROM will operate on some reduced order space $E_M$ where $M <<< N$. If $K$ were a linear operator, a Rayleigh-Ritz procedure could be used to identify the operator eigenmodes, and the low energy modes could be truncated without affecting disproportionate errors upon the solution (18). However, in dealing with a nonlinear operator $K$, the solution modes are determined geometrically by POD, and modal truncation is an empirical process that begins by eliminating low energy degrees of freedom, but is not complete until the accuracy of the resulting reduced order model is evaluated.

The first step in the POD/ROM process is identifying an appropriate linear manifold $\ell$ using the method of snapshots. Snapshots are samples of the approximate flow field solution from the numerical flow solver. One way to collect snapshots is to choose some value of $\Delta t$ and collect outputs from the time evolution of the full system at each time step. Considering the trajectory diagram in Figure A.1, this procedure is essentially collecting

sets of $\psi_n^i$ which are $N$ dimensional vectors. The number of snapshots required ($Q$) is generally much smaller than $N$.

If there is any question about the required number of snapshots $Q$, it is better to take a large number of snapshots. Later SVD will be applied, and redundant data will be weeded out via truncation. So consider a set of $Q$ snapshots of the fluid variables $\psi_j^1$, $\psi_l^2$, $\psi_p^3$, $\psi_r^4$ shown below,

$$\{\psi^1_{j=1,2,\ldots Q}\} = \{\psi_1^1, \psi_2^1, \psi_3^1, \ldots, \psi_Q^1\} \tag{A.27}$$

$$\{\psi^2_{l=1,2,\ldots Q}\} = \{\psi_1^2, \psi_2^2, \psi_3^2, \ldots, \psi_Q^2\} \tag{A.28}$$

$$\{\psi^3_{p=1,2,\ldots Q}\} = \{\psi_1^3, \psi_2^3, \psi_3^3, \ldots, \psi_Q^3\} \tag{A.29}$$

$$\{\psi^4_{r=1,2,\ldots Q}\} = \{\psi_1^4, \psi_2^4, \psi_3^4, \ldots, \psi_Q^4\} \ . \tag{A.30}$$

For now, a further restriction is required that the sets $\{\psi_j^1\}$, $\{\psi_l^2\}$, $\{\psi_p^3\}$, and $\{\psi_r^4\}$ must all be linearly independent subsets of $E_N$, and define the four-tuple of linear spaces $\ell = (\ell_1 \times \ell_2 \times \ell_3 \times \ell_4)$ as follows.

$$\ell_1 = Span(\{\psi^1_{j=1,2,\ldots Q}\}) \tag{A.31}$$

$$\ell_2 = Span(\{\psi^2_{l=1,2,\ldots Q}\}) \tag{A.32}$$

$$\ell_3 = Span(\{\psi^3_{p=1,2,\ldots Q}\}) \tag{A.33}$$

$$\ell_4 = Span(\{\psi^4_{r=1,2,\ldots Q}\}) \tag{A.34}$$

Which implies that each $\{\psi_j^i\}$ is a basis for each $\ell_i$. With the linear space $\ell$ defined, the reduced order model can be constructed as follows. For the $i^{th}$ fluid variable consider $\mu \in \ell_i$, where $\mu$ is some linear combination of $\{\psi_n^i\}$ the basis vectors for $\ell_i$. When operated on by $K$, there are no guarantee that the result is in $\ell_i$. So consider the projection of $K\mu$ onto $\ell_i$, in the hope that the projection of $K\mu_i$ onto $\ell_i^\perp$ is very small. Projection onto $\ell_i$ is itself a linear operator $P$, and so define a new operator $\widetilde{K} = PK$ which takes the outcome of the operation $K$ on $\mu$ and projects this vector onto $\ell_i$ such that

$$\widetilde{K}\mu = PK\mu = w \in \ell_i; \ \forall \mu \in \ell_i \ . \tag{A.35}$$

Now consider the operator $\widetilde{K}$ acting on the fluid variable $\mu$. The reduced order space for this fluid variable is obtained via the operations depicted below in Figure A.2.

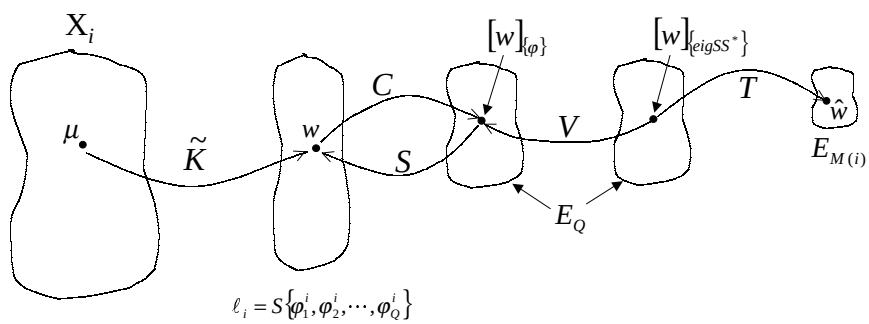



Figure A.2 Obtaining a Reduced Order Space

$C$ is the linear operation that produces the coordinates of $w$. The result is a vector of real numbers in $E_Q$. $S$ is the pseudo inverse of $C$, mapping vectors in $E_Q$ to their original counterparts in $\ell_i$. The operator $V$ is a transformation to the coordinates of $w$ from the basis formed by the orthonormal eigenvectors found via SVD on $S$. The operator $T$ is a truncation of $Q - M(i)$ coordinates of $w$ in this new basis. Before truncating low $L_2$ energy degrees of freedom one must check how the contribution of low energy modes to the solution under the operation of $\widetilde{K}$.

The choice of using the eigenvectors from the SVD of $S$ as a basis for the coordinates of $w$ provides the optimal set of coordinates for truncation (19). This is because it generates the basis that most efficiently aligns with the structure of the coordinates of all $w \in \ell_i$. As a result, there is no more efficient basis to maximize the projection of the coordinates of any $w \in \ell_i$ onto a subset of the first $M$ dimensions of the basis set. This allows identification of the dimensions of $E_Q$ that contribute little geometrically to the elements of $\ell_i$. This is discussed more fully later.

$C$ is the operator that provides the coordinates of $w$ in terms of the basis for $\ell_i$ which is the set of snapshots for the $i^{th}$ fluid variable $\{\psi_n^i\}$. This basis has $Q$ elements and the coordinates are a set of scalars $\{c_1, c_2, \ldots, c_Q\}$. For continuous functions in $x$, $C$ uses the

$L^2$ inner product and is shown to be the following.

$$c_1 = \frac{\langle w, \psi_1^i \rangle}{\| \psi_1^i \|} \tag{A.36}$$

$$c_2 = \frac{\langle w, \psi_2^i \rangle}{\| \psi_2^i \|} \tag{A.37}$$

$$\vdots \tag{A.38}$$

$$c_Q = \frac{\langle w, \psi_Q^i \rangle}{\| \psi_Q^i \|} \tag{A.39}$$

When spatially discretized to $N$ degrees of freedom, the basis set consist of vectors in $E_N$. The inner product becomes a dot product, and the operator of interest $S$ from $E_Q$ to $E_N$ is derived, for which $C$ is the pseudo inverse.

$$\frac{\langle w, \psi_j^i \rangle}{\| \psi_j^i \|} \Rightarrow \frac{(\psi_j^i)^T w}{(\psi_j^i)^T \psi_j^i} \tag{A.40}$$

$$c_j = \frac{(\psi_j^i)^T w}{(\psi_j^i)^T \psi_j^i} \tag{A.41}$$

$$(\psi_j^i)^T \psi_j^i c_j = (\psi_j^i)^T w \tag{A.42}$$

$$\begin{bmatrix} (\psi_1^i)^T \psi_1^i c_1 \\ (\psi_2^i)^T \psi_2^i c_2 \\ \vdots \\ (\psi_Q^i)^T \psi_Q^i c_Q \end{bmatrix} = \begin{bmatrix} (\psi_1^i)^T w \\ (\psi_2^i)^T w \\ \vdots \\ (\psi_Q^i)^T w \end{bmatrix} \tag{A.43}$$

$$\begin{bmatrix} (\psi_1^i)^T \psi_1^i & 0 & \dots & 0 \\ 0 & (\psi_2^i)^T \psi_2^i & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (\psi_Q^i)^T \psi_Q^i \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_Q \end{bmatrix} = \begin{bmatrix} \dots & \psi_1^i & \dots \\ \dots & \psi_2^i & \dots \\ & \vdots & \\ \dots & \psi_Q^i & \dots \end{bmatrix} w \tag{A.44}$$

$$\begin{bmatrix} \dots & \psi_1^i & \dots \\ \dots & \psi_2^i & \dots \\ & \vdots & \\ \dots & \psi_Q^i & \dots \end{bmatrix} \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ \psi_1^i & \psi_2^i & \dots & \psi_Q^i \\ \vdots & \vdots & \dots & \vdots \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_Q \end{bmatrix} = \begin{bmatrix} \dots & \psi_1^i & \dots \\ \dots & \psi_2^i & \dots \\ & \vdots & \\ \dots & \psi_Q^i & \dots \end{bmatrix} w \tag{A.45}$$

This is only true when,

$$\begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ \psi_1^i & \psi_2^i & \dots & \psi_Q^i \\ \vdots & \vdots & \dots & \vdots \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_Q \end{bmatrix} = w \ . \tag{A.46}$$

Therefore,

$$S\,[w]_{\{\psi\}} = w \tag{A.47}$$

$$Cw = [w]_{\{\psi\}} \ , \tag{A.48}$$

where

$$S = \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ \psi_1^i & \psi_2^i & \dots & \psi_Q^i \\ \vdots & \vdots & \dots & \vdots \end{bmatrix} \quad [w]_{\{\psi\}} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_Q \end{bmatrix} \quad C = (S^T S)^{-1} S^T \quad . \tag{A.49}$$

$S$ is a $N \times Q$ matrix whose columns are the basis vectors for the $i_{th}$ fluid element $\{\psi_n^i\}$. $C$ is seen to be the left inverse of $S$, generating the coordinates of $w$ in the basis $\{\psi_n^i\}$. Also consider the adjoint of $S$, which is $S^T$ mapping $E_N$ to $E_Q$ (of course $S^T \neq C$ even though they both map $E_N$ to $E_Q$). These operations are depicted in Figure A.3.

The set of $Q$ snapshots of each fluid variable $\{\psi_n^i\}$ forms a basis for each $\ell_i$, and by using the coordinates of elements of $\ell_i$ the order of the problem has been reduced from $E_N$ to $E_Q$. Even though $Q << N$, the number of degrees of freedom can be reduced further still. When $Q$ is too large, it is difficult to obtain linearly distinct sets. This causes numerical difficulties when inverting the nearly singular matrix later on in this process. $[w]_{\{\psi\}}$ is transformed into a new ordered basis $\{v_n\} = \{v_1, v_2, \dots, v_Q\}$ where the last $Q - (M(i) + 1)$ components of $[w]_{\{v_n\}}$ can be optimally truncated. This creates an $M(i)$ dimensional vector $\widehat{w} \in E_{M(i)}$ where $M(i) < Q$. This truncation procedure is unique for each fluid variable. The proper basis $\{v_n\}$ is selected in such a way as to maximize

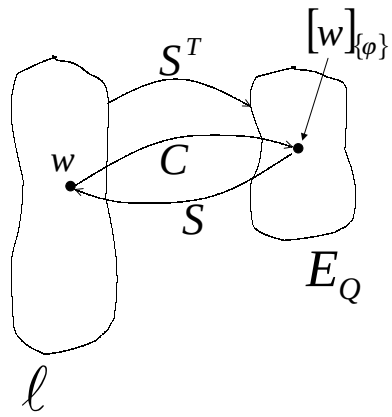



Figure A.3 Linear Operations between $E_Q$ and $E_N$

equation (A.50), where $\widehat{w}_{aug}$ is simply $\widehat{w}$ augmented with $Q-(M(i)+1)$ zeros in the last vector locations to give meaning to the inner product.

$$\max_{\{\mathrm{v_n}\}\subset\ell_\mathrm{i}} \left\langle \widehat{w}_{aug}, [w]_{\{v_n\}} \right\rangle ; \forall\ w \in \ell_i \tag{A.50}$$

where,

$$[w]_{\{v_n\}} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{M(i)} \\ c_{M(i)+1} \\ \vdots \\ c_Q \end{bmatrix}_{\{v_n\}} \qquad \widehat{w}_{aug} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{M(i)} \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{\{v_n\}} . \tag{A.51}$$

This is a geometric optimization on the elements of $E_Q$. It implies that the first $M(i)$ POD basis functions capture more energy on average that the first $M(i)$ functions of any other basis (19). Consider the operator $S$ from $E_Q$ to $\ell_i$. If $S$ were a symmetric operator, the expansion theorem for linear operators could be applied to obtain a set of eigenfunctions. A transition to this new basis would meet the optimality conditions from equation (A.50) above (19, 18). This same result can also be derived by variational princi-

ples (19, 18). However, $S$ is a non-symmetric linear operator from $E_Q$ to $E_N$. Fortunately the same optimal expansion can be obtained using the eigenvectors of either $S^T S$ or $SS^T$, which is known as the Singular Value Decomposition of $S$ (18). Since an orthonormal basis for $E_Q$ where $Q << N$ is desired, consider the eigenvectors of the $Q \times Q$ matrix $S^T S$. The operator $S^T S$ is non-negative definite (19), which guarantees the existence of $Q$ total eigenvalues, some positive and others zero valued. The eigenvectors associates with each eigenvalue are guaranteed to be mutually orthogonal vectors in Euclidean $Q$-space.

The operation of $S^T$ on the elements of $\ell_i$ is described first. The range of $S^T$ is $E_Q$, and SVD will optimally reproduce the elements of $E_Q$ using $\{u_n\}$; the set of orthonormal eigenvectors of $S^T S$.

$$\widehat{w}_\psi = S^T w = \sum_{n=1}^{Q} \langle S^T w, u_n \rangle u_n \tag{A.52}$$

The non-zero eigenvalues of $S^T S$ are identical to the eigenvalues of $SS^T$. If $S^T$ is a $Q \times N$ matrix of rank $Q$ ($Q < N$), then $S^T S$ will have $Q$ non-zero eigenvalues producing a set of $Q$ eigenvectors $\{u_n\} = \{u_1, u_2, \ldots, u_Q\}$. Likewise, $SS^T$ will have the same $Q$ non-zero eigenvalues along with $N - Q$ zero eigenvalues. The non-zero eigenvalues will yield a set of $Q$ eigenvectors $\{v_n\} = \{v_1, v_2, \ldots, v_Q\}$. The sets of eigenvectors are related (18),

$$v_n = \frac{S u_n}{\lambda_n} \ , \tag{A.53}$$

which leads to the following result,

$$S^T w = \sum_{n=1}^{Q} \langle S^T w, u_n \rangle u_n \tag{A.54}$$

$$S^T w = \sum_{n=1}^{Q} \langle w, S u_n \rangle u_n \tag{A.55}$$

$$S^T w = \sum_{n=1}^{Q} \langle w, \lambda_n v_n \rangle u_n \tag{A.56}$$

$$S^T w = \lambda_1 \langle w, v_1 \rangle u_1 + \lambda_2 \langle w, v_2 \rangle u_2 + \ldots \lambda_Q \langle w, v_Q \rangle u_Q \ , \tag{A.57}$$

$$S^T w = \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ u_1 & u_2 & \dots & u_Q \\ \vdots & \vdots & \dots & \vdots \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_Q \end{bmatrix} \begin{bmatrix} \langle w, v_1 \rangle \\ \langle w, v_2 \rangle \\ \vdots \\ \langle w, v_Q \rangle \end{bmatrix} . \tag{A.58}$$

Now define the following matrices,

$$V_Q = \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ u_1 & u_2 & \dots & u_Q \\ \vdots & \vdots & \dots & \vdots \end{bmatrix} \quad \Lambda_Q = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_Q \end{bmatrix} \quad \widehat{w}_Q = \begin{bmatrix} \langle w, v_1 \rangle \\ \langle w, v_2 \rangle \\ \vdots \\ \langle w, v_Q \rangle \end{bmatrix} . \tag{A.59}$$

Notice that $V_Q$ is the solution to the eigen equation $S^T S V_Q = V_Q \Lambda_Q$. This yields the reduced order linear mapping $\Psi_Q$,

$$S^T w = V_Q \Lambda_Q \widehat{w}_Q \tag{A.60}$$

$$S^T w = S^T S V_Q \widehat{w}_Q \tag{A.61}$$

$$w = S V_Q \widehat{w}_Q \tag{A.62}$$

$$\Psi_Q = S V_Q \tag{A.63}$$

$$w = \Psi_Q \widehat{w}_Q . \tag{A.64}$$

The linear operation $\Psi_Q$ transforms $w \in \ell_i$ into $E_Q$. The elements of $E_Q$ are the coordinates of $w$ in the basis $\{v_n\}$ which are the set of eigenvectors of $SS^T$ corresponding to non-zero eigenvalues. The elements of the basis $\{v_n\} = \{v_1, v_2, \dots, v_Q\}$ are ordered by eigenvalue $\lambda_1 > \lambda_2 > \dots > \lambda_Q$. If $\lambda_{M+1} <<< \lambda_1$ for the $i^{th}$ fluid variable, $\lambda_{M+1} > \lambda_{M+2} > \dots > \lambda_Q$ and their associated eigenvectors can be eliminated with little geometric affect to the elements of $E_Q$. This truncation produces a vector $\widehat{w} \in E_{M(i)}$, and the reduced order space for $K$ is the linear space $(E_{M(1)} \times E_{M(2)} \times E_{M(3)} \times E_{M(4)})$. Since the eigenvectors associated with the truncated eigenvalues are excluded, the coordinate transformation matrix of eigenvectors $V$ becomes a $Q \times M(i)$ matrix.

The final step involves transforming the operator $K$ for operation in $(E_{M(1)} \times E_{M(2)} \times E_{M(3)} \times E_{M(4)})$ instead of operation on $(X_1 \times X_2 \times X_3 \times X_4)$. This is most simply done using subspace projection (3). A unique reduced order mapping can be identified for each of the fluid variables,

$$w_1 = \Psi_1 \widehat{w}_1 = S_1 V_1 \widehat{w}_1 \tag{A.65}$$

$$w_2 = \Psi_2 \widehat{w}_2 = S_2 V_2 \widehat{w}_2 \tag{A.66}$$

$$w_3 = \Psi_3 \widehat{w}_3 = S_3 V_3 \widehat{w}_3 \tag{A.67}$$

$$w_4 = \Psi_4 \widehat{w}_4 = S_4 V_4 \widehat{w}_4 \ , \tag{A.68}$$

where $(w_1, w_2, w_3, w_4) \in (X_1 \times X_2 \times X_3 \times X_4) \subset (E_N \times E_N \times E_N \times E_N)$; also $(\widehat{w}_1, \widehat{w}_2, \widehat{w}_3, \widehat{w}_4) \in (E_{M(1)} \times E_{M(2)} \times E_{M(3)} \times E_{M(4)})$. The full and reduced order fluid variables are collocated into a single column vector $\underline{U}$.

$$\underline{U} = \begin{bmatrix} \uparrow \\ \vdots \\ w_1 \\ \vdots \\ \downarrow \\ \vdots \\ \uparrow \\ \vdots \\ w_4 \\ \vdots \\ \downarrow \end{bmatrix} \in E_{4N}; \quad \widehat{\underline{U}} = \begin{bmatrix} \uparrow \\ \widehat{w}_1 \\ \downarrow \\ \vdots \\ \uparrow \\ \widehat{w}_4 \\ \downarrow \end{bmatrix} \in E_{M(1)+M(2)+M(3)+M(4)} \tag{A.69}$$

The linear mapping between $\underline{U}$ and $\widehat{\underline{U}}$ is,

$$\underline{U} = \begin{bmatrix} \Psi_1 & 0 & 0 & 0 \\ 0 & \Psi_2 & 0 & 0 \\ 0 & 0 & \Psi_3 & 0 \\ 0 & 0 & 0 & \Psi_4 \end{bmatrix} \widehat{\underline{U}} \tag{A.70}$$

$$\underline{U} = \begin{bmatrix} S_1V_1 & 0 & 0 & 0 \\ 0 & S_2V_2 & 0 & 0 \\ 0 & 0 & S_3V_3 & 0 \\ 0 & 0 & 0 & S_4V_4 \end{bmatrix} \widehat{\underline{U}} \tag{A.71}$$

$$\underline{U} = \begin{bmatrix} S_1 & 0 & 0 & 0 \\ 0 & S_2 & 0 & 0 \\ 0 & 0 & S_3 & 0 \\ 0 & 0 & 0 & S_4 \end{bmatrix} \begin{bmatrix} V_1 & 0 & 0 & 0 \\ 0 & V_2 & 0 & 0 \\ 0 & 0 & V_3 & 0 \\ 0 & 0 & 0 & V_3 \end{bmatrix} \widehat{\underline{U}} \ , \tag{A.72}$$

which is denoted as follows,

$$\begin{aligned} \underline{U} &= \Psi\widehat{\underline{U}} \\ \underline{U} &= SV\widehat{\underline{U}} \ , \end{aligned} \tag{A.73}$$

with the following definitions,

$$\Psi = \begin{bmatrix} \Psi_1 & 0 & 0 & 0 \\ 0 & \Psi_2 & 0 & 0 \\ 0 & 0 & \Psi_3 & 0 \\ 0 & 0 & 0 & \Psi_4 \end{bmatrix} \quad S = \begin{bmatrix} S_1 & 0 & 0 & 0 \\ 0 & S_2 & 0 & 0 \\ 0 & 0 & S_3 & 0 \\ 0 & 0 & 0 & S_4 \end{bmatrix} \quad V = \begin{bmatrix} V_1 & 0 & 0 & 0 \\ 0 & V_2 & 0 & 0 \\ 0 & 0 & V_3 & 0 \\ 0 & 0 & 0 & V_4 \end{bmatrix} .$$

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 & 0 & 0 \\ 0 & \Lambda_2 & 0 & 0 \\ 0 & 0 & \Lambda_3 & 0 \\ 0 & 0 & 0 & \Lambda_4 \end{bmatrix} \tag{A.74}$$

Here $\Psi$ is a $4N \times (M(1) + M(2) + M(3) + M(4))$ matrix, $S$ is a $4N \times 4Q$ matrix, $V$ is a $4Q \times (M(1) + M(2) + M(3) + M(4))$ matrix, and $\Lambda$ is a $(M(1) + M(2) + M(3) + M(4)) \times (M(1) + M(2) + M(3) + M(4))$ matrix. To reduce the order of equation (A.21), (A.73) needs to be rearranged. Remembering that $V$ is a non-symmetric $Q \times N$ matrix with linearly independent columns (they are orthogonal), the pseudo inverse of $V$ exists, and it is used to invert the reduced order mapping. Also, the zero valued eigenvalues have been removed from $\Lambda$ which becomes a diagonal matrix with positive values on the diagonals. Thus the invertibility of $\Lambda$ is also guaranteed.

$$\underline{U} = S \cdot V \cdot \underline{\widehat{U}} \tag{A.75}$$

$$(SV)^T \cdot \underline{U} = (SV)^T \cdot SV \cdot \underline{\widehat{U}} \tag{A.76}$$

$$(SV)^T \cdot \underline{U} = V^T S^T SV \cdot \underline{\widehat{U}} \tag{A.77}$$

$$V^T S^T \cdot \underline{U} = V^T V \Lambda \cdot \underline{\widehat{U}} \tag{A.78}$$

$$\underline{\widehat{U}} = \Lambda^{-1} \cdot (V^T V)^{-1} \cdot V^T \cdot S^T \cdot \underline{U} \tag{A.79}$$

When $V$ is symmetric, which will occur if no modal truncation is needed to eliminate zero values eigenvalues, the pseudo inverse can be eliminated and this equations reduces to the following.

$$\underline{\widehat{U}} = \Lambda^{-1} \cdot V^{-1} \cdot S \cdot \underline{U} \tag{A.80}$$

Multiplying both sides of equation (A.21) by $\Lambda^{-1}(V^T V)^{-1} V^T S^T$ yields the following reduced order flow solver.

$$\underline{\widehat{U}}^{n+1} = \underline{\widehat{U}}^{n} + \Delta t \cdot \Lambda^{-1}(V^T V)^{-1} V^T S^T \cdot R(\underline{U}^n) \tag{A.81}$$

So the POD/ROM reduces each flow variable from $4N$ to $M = M(1) + M(2) + M(3) + M(4)$, where $M$ is the combined number of snapshots for each fluid variable, truncated to some extent. Each reduced order degree of freedom modifies a spatial solution mode, whose contribution to the full order solution is of order with the magnitude of the corresponding eigenvalue. The method of order reduction in equation (A.81) relies on the full order function evaluation at each flow field integration step. As such, the order of each

integration step is not actually reduced. However, this reduction technique can greatly increase the time step size allowed for stability.

### *A.3 Summary of POD/ROM*

To summarize the reduced order modeling procedure, refer to the boxology of equation (A.81) in Figure A.4.

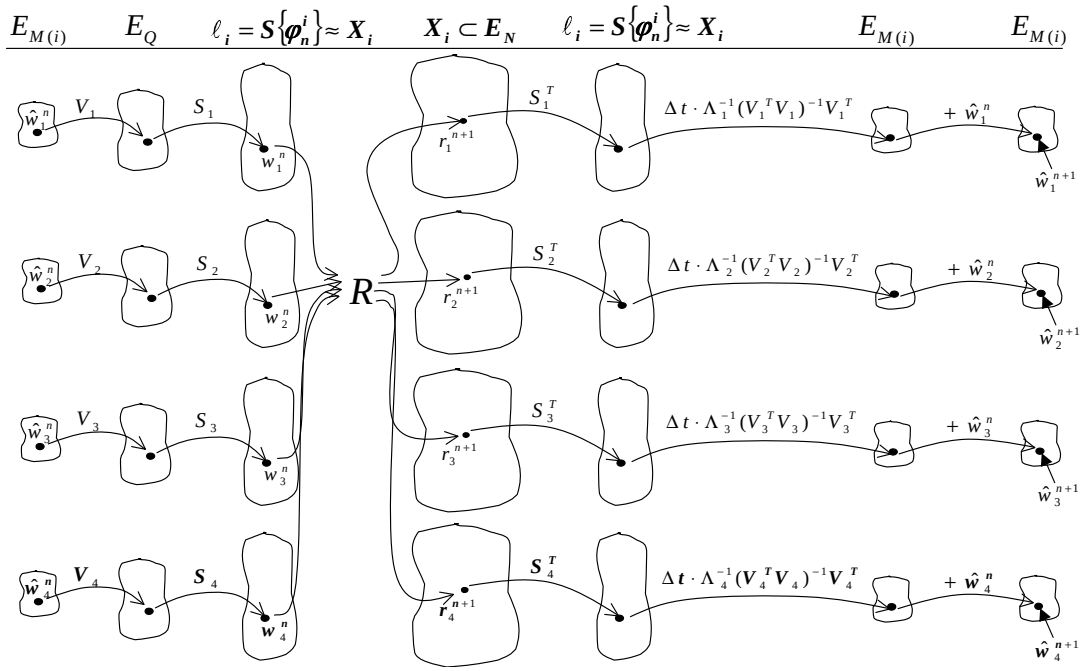



Figure A.4 Operation on the Reduced Order Space

Considering each fluid variable separately at time interval $n$, the process takes the reduced order vector $\widehat{w}_i^n$ maps it into the coordinates of $w_i^n$ in the basis of $\ell_i$ which is the set of snapshots $\{\psi_n^i\}$. The result is a vector in $E_Q$, whose entries are the modal coefficients used to reconstruct the full order flow variable $w_i^n$ as a function of $x$ (or an element of $E_N$) via operation $S_i$.

This function ($w_i^n$) is then operated on by the nonlinear operator $R$, along with the other fluid variables. This operation produces a residual $r_i^{n+1} \in X_i$ at full system order,

which must be augmented with the last solution to produce the completed state update. This full order residual is projected into $\ell_i$ via the linear operator $S^T$. The combination of $S^T \circ R(\underline{w})$ is a projection operator $\widetilde{R}$ which casts the full order residual onto $\ell_i$ after every state update. The projection of the full order residual into $\ell_i$ is next projected into the reduced order space $E_{M(i)}$ via the operation $\Delta t \cdot \Lambda_i^{-1}(V_i^T V_i)^{-1} V_i^T$. The results is a reduced order state update which is added to $\widehat{w}_i^n$ to obtain $\widehat{w}_i^{n+1}$.

The reduced order model propagates the modal coefficients $\underline{\widehat{U}}$ for all time of interest. Equation $(A.73)$ maps the reduced order results for all time back into the full order operator manifold $(X_1 \times X_2 \times X_3 \times X_4)$ where the accuracy can be analyzed.

## *Appendix B. Additional Results for Quasi 1-D Nozzle*

Additional results and analysis are presented herein for the quasi 1-D nozzle problem. The full order solution was generated via Roe's scheme with 63 spatial grid points, and the reduced order models were generated via methods described in Appendix A. Two transient flow evolutions are considered; one that includes a large range of dynamics, and a low dynamic transient case that never deviates greatly from its initial condition. These two cases provided insight into accurate reduced order modeling of shock waves with POD.

Shannon sampling theory (59) suggests that many snapshots of a highly dynamic flow evolution are necessary to recreate the flow evolution with a reduced order model. In addition, the reduced order model for the highly dynamical case should require more modes relative to a low dynamic case. These issues are investigated herein. Also, observations are made regarding the effect of discontinuities on the number of modes. Spectral methods applied to fluid dynamics indicate an inordinate number of modes (relative to "smooth" flow fields) are required to recreate shock like discontinuities. Much like many terms in a Fourier series are required to reproduce a step function (20). The necessity of many modes in the POD/ROM to replicate discontinuous spatial functions is investigated. Finally, although the reduced order model considers three fluid variables ($\rho \in X_1$, $\rho u \in X_2$, $E_t \in X_3$), the discussion will generally center on the behavior of density ($\rho$). This avoids excessive discussion when all the fluid variables behave in a similar manner.

### *B.1 Replicating a 100 Second High Dynamic Flow Evolution*

The high dynamic case was devised to produce a wide variety of flow dynamics during the evolution of the flow. Since the flow converges to steady state, the evolution was a Cauchy sequence of vectors (18). This is depicted notionally in Figure B.1. The initial conditions and boundary data described in Chapter 3 were used, so the steady state solution for $K$ (that is $\psi^1_{100}$ in Figure B.1) was the curve for $\gamma = 1.4$ shown in Figure 3.5. Notice that the steady state solution contained a discontinuity. Shocks occur when the defining boundary conditions force $K$ to display hyperbolic PDE behavior for a region of the flow field. The resulting flow evolution swept from smooth linear functions to discontinuous

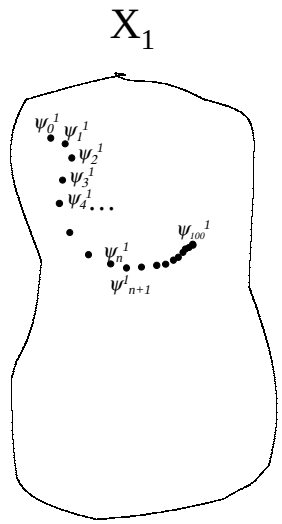



Figure B.1 100$s$ Density Function Flow Evolution

ones. The 100$s$ case started as a linear function, and formed a discontinuity. Next, the location of the discontinuity underwent a harmonic oscillation about the steady state shock location. The steady state solution was very nearly reached by 100$s$.

A reduced order model was created for this case by taking snapshots of each fluid variable at fixed time increments. These sets of $N = 63$ vectors formed the basis functions for the three linear manifolds $\ell_1 = S\{\psi_n^1\} \subset E_N$, $\ell_2 = S\{\psi_n^2\} \subset E_N$, and $\ell_3 = S\{\psi_n^3\} \subset E_N$. Three metrics were considered to record the accuracy of the reduced order models. The first was the maximum error norm $L_{max}$, which is evaluated in equation (B.1) (shown for error in density). Notice that $L_{max}$ is reported as a percentage of the value of the fluid variable at that $x$ location and time.

$$\begin{aligned} L_{max}(t) &= \frac{\overset{\max}{x_{max} \in \mathbb{R}} |err(x,t)|}{\rho(x_{max},t)_{cfd}} \times 100\% \\ err(x,t) &= \rho(x,t)_{cfd} - \rho(x,t)_{pod} \end{aligned} \tag{B.1}$$

The $L^2$ error norm was also used as shown in equation (B.2). For comparative reference, the spatially averaged value of each fluid variable was also computed at each corresponding time (shown for density $\overline{\rho}$). At any given instant in time, the $L^2$ error norm represented a standard deviation about the mean value of the fluid variable. Using density as an

example, if $L^2_{err}$ was a small percentage of $\overline{\rho}$, the POD/ROM did a good job of recreating density over the great majority of the spatial domain (at that time step). If a small part of the spatial domain was characterized by large errors, the $L_2$ error norm would average this out, but the $L_{max}$ norm would be large to capture this event. However, if the POD/ROM kept to within a certain threshold of the true solution, but never accurately replicated any significant spatial portion of the function, then the $L^2$ error norm would be large, when the $L_{max}$ norm might be small.

$$\begin{aligned} L^2_{err}(t) &= \sqrt{\int_0^{x_f} err(x,t)^2 dx} \\ \overline{\rho}(t) &= \frac{\int_0^{x_f} \rho(x,t)dx}{x_f} \end{aligned} \tag{B.2}$$

Both $L_{max}$ and $L^2_{err}$ evaluate accuracy for a given value of time. The time integrated $L_2$ norm described in equation (B.3) was used to consider the performance of the POD/ROM for the flow evolution. The spatially averaged value of the fluid variable was time integrated to provided a metric for how well the POD/ROM tracked the entire flow evolution.

$$\begin{aligned} L^t_{err} &= \int_0^{t_f} L^2_{err}(t)dt \\ \overline{\rho}^t &= \int_0^{t_f} \overline{\rho}(t)dt \end{aligned} \tag{B.3}$$

With these metrics in place, the accuracy of several POD/ROMs was evaluated for this case. The POD/ROMs used both 50 and 100 snapshots, taken at $2s$ and $1s$ intervals respectively during the $100s$ evolution of the full system model.

Since the test case was relatively low order (spatial discretization of $N = 63$), the set of 100 snapshots had linearly dependent data. Even though the simple 1-D problem was relatively low order, a large number of snapshots was required to build the POD/ROM. Use of a very few snapshots required exceptional insight into the structure of the flow field. Since it was hard to know a priori where in time to take snapshots to capture the key solution structures, the safest method was to take too many snapshots and truncate unnecessary data. In working with 100 snapshots with $Q = 100 > N = 63$, the earlier restriction that the snapshots be linearly independent vectors was neglected. This restric-

tion was not strictly necessary as long as truncation is employed to eliminate redundant data. This permited an extremely large number of snapshots.

The performance of each POD/ROM is recorded in the tables at the end of this section. Performance in terms of the $L_{max}$ error norm is presented in Table B.1, performance in terms of the $L^2_{err}$ norm is presented in Table B.2, and performance in terms of the $L^2_{err}$ norm is presented in Table B.3. These results are summarized below.

An exceptionally good representation of the full order trajectory was obtained using 58 total modes (20 modes for $\rho$, 19 modes for $\rho u$, and 19 modes for $E_t$). Every metric indicates a close adherence to the true spatial solution for each fluid variable at each point in time. The largest $L_{max}$ for any fluid variable was only 1.15%. This indicates that there is no place or time that the reduced order model does not closely match the true solution. The largest $L^2_{err}$ for any fluid variable at any time was only 0.404%, indicating that spatial features of the solution are also closely tracked for all time. The small value of $L^t_{err}$ (about 0.1%) supports this conclusion as well. For this high dynamical flow evolution, an exceptional solution accuracy was maintained while reducing the order from 189 degrees of freedom to 58.

Depending on the application, errors of 5% on the average might be acceptable. This was achieved with about 45 modes (15 modes for $\rho$, 15 modes for $\rho u$, and 15 modes for $E_t$). From 36 to 28 modes, the reduced order model generated a reasonable approximation of the full order solution for most of the trajectory. Reduced order models below 25 modes were of questionable use. When using 50 snapshots, an exceptionally accurate solution could not be obtained. The timing of the 50 snapshots did not capture the key structures for a highly accurate solution. However, the 47 mode solution was comparable to the 48 mode solution generated from 100 snapshots. Finally, notice that the POD/ROM got worse when too many modes were included. The modes corresponding to very low eigenvalues captured dynamics due to machine numerics, not flow structures. When these modes were included, the solution was polluted with modal errors. Instabilities arose that prevented the solution from reaching steady state.

*B.2 Replicating a 30 Second Low Dynamic Flow Evolution*

To illustrate that spatial discontinuities are of no consequence to the number of modes, a POD/ROM was generated to replicate a low dynamical trajectory of shocked flow solutions. Specifically, the last $30s$ of the $100s$ flow evoution was replicated. This comprised a sequence of functions that was nearly converged to the steady state value. The low dynamical trajectory is depicted notionally in Figure B.2. For this case there was

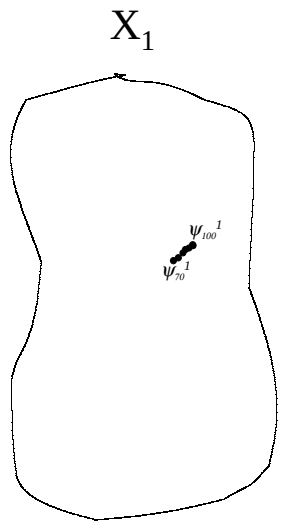



Figure B.2 $100s$ Density Function Flow Evolution

no formation of a shock from a smooth initial condition. Also, the movement of the shock was limited to small motions in the back of the shock. The range of solutions for density were obtained from the full order solution and are shown in Figure B.3.

Several POD/ROMs were generated from a set of 30 evenly spaced snapshots ($1s$ intervals of the full system model). The results are presented in terms of the three error norms defined in equations (B.1), (B.2) and (B.3). Performance in terms of the $L_{max}$ error norm is presented in Table B.4, performance in terms of the $L^2_{err}$ norm is presented in Table B.5, and performance in terms of the $L^2_{err}$ norm is presented in Table B.6.

An exceptionally good representation of the full order trajectory was obtained using only 9 total modes (3 modes for $\rho$, 3 modes for $\rho u$, and 3 modes for $E_t$). The largest $L_{max}$ for any fluid variable was only 1.4% indicating that there was no place or time that the reduced order model did not closely match the true solution. The largest $L^2_{err}$ was only

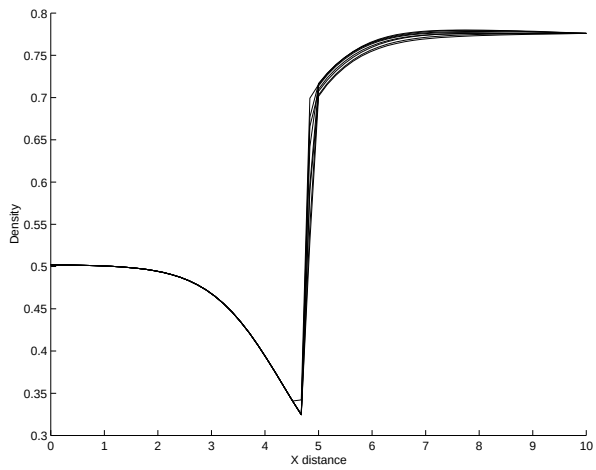



Figure B.3 Density Function $t = 0 \rightarrow 30s$

0.707%, indicating that spatial features of the solution were also closely tracked for all time. The small value of $L^t_{err}$ (about 0.44%) supports this conclusion as well. Also, a very useful reduced order model (average errors $< 5\%$) was generated from even fewer modes (2 modes for $\rho$, 3 modes for $\rho u$, and 2 modes for $E_t$ for a total of 7 degrees of freedom).

This is in stark contrast to the 58 modes required for exceptionally accurate reproduction of the high dynamical trajectory. The fact that both cases involved flow fields with discontinuities proves conclusively that the number of required modes is not intrinsically increased for shocked flows. The presence of the discontinuity was readily recreated with only the first three modes. The smooth portions of the flow field were recreated with three modes as well. Unlike sine or cosine series, the POD/ROM identified discontinuous modes to represent discontinuous flow fields. A very few of these modes were adequate in representing the full order solution. The spectral methods attempt to recreate discontinuous flow fields with smooth basis functions (complex exponentials, Chebyshev polynomials, and Legendre polynomials) (20). A great many modes are required to accurately recreate a discontinuous function from a set of smooth functions.

### *B.3 Summary of Key Insights*

The ability of POD/ROM to replicate two shocked flow evolutions was presented; one characterized by high dynamics, and another with lower dynamics. POD almost perfectly recreated the low dynamic case using only 3 degrees of freedom per fluid variable. Unlike traditional spectral methods, the existence of a discontinuity in the flow solution did not increase the number of modes required for accuracy relative to smooth flow field solutions.

In addition, the POD/ROM could very accurately reproduce a flow evolution involving a transition from smooth to shocked flow, including a shock with significant movement. Recreating this high dynamical case required more modes for accurate representation (15 modes per fluid variable). This was roughly 5 times greater than the low dynamical case.

The spatial features of a particular flow field did not determine the number of modes for an accurate POD/ROM. Rather, the variety of vectors making up the flow evolution determined the required number of modes. The high dynamical cases required more modes regardless of the presence or absence of shocks.

Table B.1 Maximum Error Norm for 100$s$ Trajectory

| Evenly Spaced Snapshots | Modes Retained $\rho/\rho u/E_t$ | Highest $L_{max}$ $\rho$ (time occurred) | Highest $L_{max}$ $\rho u$ (time occurred) | Highest $L_{max}$ $E_t$ (time occurred) |
|---|---|---|---|---|
| 100 | 22/21/22 | 2.7584% (24$s$) | 0.5519% (29$s$) | 2.2318% (24$s$) |
| 100 | 20/19/19 | 1.1553% (24$s$) | 0.3522% (13$s$) | 0.9562% (24$s$) |
| 100 | 15/15/15 | 8.6998% (29$s$) | 2.0238% (29$s$) | 7.1013% (30$s$) |
| 100 | 12/15/12 | 13.203% (11$s$) | 3.8913% (59$s$) | 10.729% (11$s$) |
| 100 | 12/12/15 | 31.397% (20$s$) | 6.5045% (22$s$) | 26.507% (20$s$) |
| 100 | 12/12/12 | 43.099% (23$s$) | 7.2527% (22$s$) | 38.347% (23$s$) |
| 100 | 10/12/10 | 26.81% (18$s$) | 14.677% (20$s$) | 21.609% (18$s$) |
| 100 | 8/12/8 | 56.644% (81$s$) | 19.376% (14$s$) | 51.236% (79$s$) |
| 100 | 6/9/6 | 60.234% (29$s$) | 14.259% (23$s$) | 55.408% (29$s$) |
| 100 | 5/11/5 | 73.717% (100$s$) | 29.075% (47$s$) | 67.137% (82$s$) |
| 100 | 4/9/4 | 72.542% (36$s$) | 30.341% (37$s$) | 65.095% (36$s$) |
| 50 | 20/19/19 | 63.577% (25$s$) | 34.673% (2$s$) | 59.638% (25$s$) |
| 50 | 16/16/15 | 16.22% (41$s$) | 4.5572% (2$s$) | 12.6% (41$s$) |
| 50 | 12/15/12 | 27.035% (17$s$) | 5.8349% (16$s$) | 23.261% (17$s$) |
| 50 | 10/12/10 | 21.7% (5$s$) | 11.303% (12$s$) | 17.359% (5$s$) |
| 50 | 8/12/8 | 57.314% (84$s$) | 25.913% (23$s$) | 51.756% (83$s$) |

Table B.2 $L^2$ Error Norm for $100s$ Trajectory

| Evenly Spaced Snapshots | Modes Retained $\rho/\rho u/E_t$ | Highest $L^2_{err}$ $\rho$ (%/$t$ occurred) | Highest $L^2_{err}$ $\rho u$ (%/$t$ occurred) | Highest $L^2_{err}$ $E_t$ (%/$t$ occurred) |
|---|---|---|---|---|
| 100 | 22/21/22 | $4.524 \times 10^{-3}$ (0.73036%/24$s$) | $2.1744 \times 10^{-3}$ (0.37295%/1$s$) | 0.010065 (0.60986%/24$s$) |
| 100 | 20/19/19 | $2.0452 \times 10^{-3}$ (0.33018%/24$s$) | $2.3548 \times 10^{-3}$ (0.40389%/1$s$) | $4.8592 \times 10^{-3}$ (0.30319%/1$s$) |
| 100 | 15/15/15 | 0.017119 (2.6776%/30$s$) | $6.8873 \times 10^{-3}$ (1.2064%/2$s$) | 0.038778 (2.2926%/30$s$) |
| 100 | 12/15/12 | 0.054925 (9.022%/53$s$) | 0.018608 (3.624%/11$s$) | 0.1166 (7.3357%/53$s$) |
| 100 | 12/12/15 | 0.10109 (16.579%/63$s$) | 028505 (5.6149%/13$s$) | 0.22828 (14.248%/63$s$) |
| 100 | 12/12/12 | 0.11507 (18.722%/23$s$) | 0.035238 (6.9411%/13$s$) | 0.2512 (15.332%/23$s$) |
| 100 | 10/12/10 | 0.12285 (19.559%/26$s$) | 0.06426 (13.011%/21$s$) | 0.30474 (18.237%/26$s$) |
| 100 | 8/12/8 | 0.1669 (29.068%/15$s$) | 0.092861 (18.54%/16$s$) | 0.38962 (23.885%/79$s$) |
| 100 | 6/9/6 | 0.24601 (40.527%/56$s$) | 0.10474 (20.255%/10$s$ | 0.60358 (37.992%/54$s$) |
| 100 | 5/11/5 | 0.55191 (85.903%/33$s$) | 0.21771 (41.623%/45$s$) | 1.3459 (79.559%/32$s$) |
| 100 | 4/9/4 | 0.50537 (78.857%/31$s$) | 0.22961 (44.147%/38$s$) | 1.2282 (72.736%/29$s$) |
| 50 | 20/19/19 | 0.44107 (77.022%/8$s$) | 0.30661 (53.706%/2$s$) | 1.1197 (73.689%/3$s$) |
| 50 | 16/16/15 | 0.029168 (4.5938%/40$s$) | 0.026102 (4.7239%/4$s$) | 0.064676 (3.9095%/40$s$) |
| 50 | 12/15/12 | 0.055187 (9.4692%/17$s$) | 0.033961 (5.8249%/1$s$) | 0.12376 (7.9841%/17$s$) |
| 50 | 10/12/10 | 0.095253 (15.378%/24$s$) | 0.050962 (10.281%/19$s$) | 0.2379 (14.414%/24$s$) |
| 50 | 8/12/8 | 0.029179 (0.73036%/39$s$) | 0.015646 (0.73036%/23$s$) | 0.15092 (0.73036%/83$s$) |

Table B.3 Time Integrated $L^2$ Error Norm for $100s$ Trajectory

| Evenly Spaced Snapshots | Modes Retained $\rho/\rho u/E_t$ | $\int_0^{100s} L^2_{err}(t)dt$ $\rho$ | $\int_0^{100s} L^2_{err}(t)dt$ $\rho u$ | $\int_0^{100s} L^2_{err}(t)dt$ $E_t$ |
|---|---|---|---|---|
| 100 | 22/21/22 | 0.17638 (0.28764%) | 0.076261 (0.14796%) | 0.44535 (0.27714%) |
| 100 | 20/19/19 | 0.079291 (0.12931%) | 0.051073 (0.099092%) | 0.18187 (0.11318%) |
| 100 | 15/15/15 | 0.81332 (1.3264%) | 0.30912 (0.59975%) | 1.7115 (1.0651%) |
| 100 | 12/15/12 | 3.0654 (4.999%) | 0.8674 (1.6829%) | 4.7198 (2.9372%) |
| 100 | 12/12/15 | 4.4937 (7.3284%) | 1.5951 (3.0948%) | 10.176 (6.3324%) |
| 100 | 12/12/12 | 3.7507 (6.1167%) | 1.1607 (2.252%) | 7.7914 (4.8486%) |
| 100 | 10/12/10 | 3.8755 (6.3201%) | 1.9798 (3.8413%) | 10.105 (6.2884%) |
| 100 | 8/12/8 | 9.0159 (14.703%) | 3.1343 (6.0812%) | 21.409 (13.323%) |
| 100 | 6/9/6 | 15.703 (25.608%) | 4.031 (7.821%) | 38.065 (23.688%) |
| 100 | 5/11/5 | 38.633 (63.002%) | 15.087 (29.272%) | 93.079 (57.924%) |
| 100 | 4/9/4 | 22.67 (36.969%) | 7.0984 (13.772%) | 54.925 (34.18%) |
| 50 | 20/19/19 | 17.437 (28.436%) | 5.8231 (11.298%) | 43.06 (26.797%) |
| 50 | 16/16/15 | 1.2304 (2.0064%) | 0.52634 (1.0212%) | 2.8118 (1.7498%) |
| 50 | 12/15/12 | 2.1439 (3.4962%) | 0.91791 (1.7809%) | 4.8452 (3.0152%) |
| 50 | 10/12/10 | 3.1254 (5.0969%) | 1.4876 (2.8863%) | 8.1207 (5.0536%) |
| 50 | 8/12/8 | 10.291 (16.783%) | 3.7 (7.1787%) | 24.153 (15.03%) |

Table B.4 Maximum Error Norm for $30s$ Trajectory

| Evenly Spaced Snapshots | Modes Retained $\rho/\rho u/E_t$ | Highest $L_{max}$ $\rho$ (time occurred) | Highest $L_{max}$ $\rho u$ (time occurred) | Highest $L_{max}$ $E_t$ (time occurred) |
|---|---|---|---|---|
| 30 | 12/12/12 | 0.034373% ($17s$) | 0.0094979% ($17s$) | 0.025235% ($17s$) |
| 30 | 10/10/10 | 0.04179% ($10s$) | 0.012389% ($10s$) | 0.031264% ($10s$) |
| 30 | 8/8/8 | 0.029745% ($23s$) | 0.023077% ($2s$) | 0.028837% ($23s$) |
| 30 | 6/6/6 | 5.9281% ($30s$) | 1.2139% ($30s$) | 1.711% ($30s$) |
| 30 | 4/4/4 | 3.383% ($15s$) | 0.92973% ($15s$) | 2.5574% ($15s$) |
| 30 | 3/3/3 | 1.4009% ($6s$) | 0.60551% ($20s$) | 1.3872% ($6s$) |
| 30 | 2/3/2 | 6.6826% ($14s$) | 8.098% ($30s$) | 5.1885% ($13s$) |

Table B.5 $L^2$ Error Norm for $30s$ Trajectory

| Evenly Spaced Snapshots | Modes Retained $\rho/\rho u/E_t$ | Highest $L^2_{err}$ $\rho$ (%/$t$ occurred) | Highest $L^2_{err}$ $\rho u$ (%/$t$ occurred) | Highest $L^2_{err}$ $E_t$ (%/$t$ occurred) |
|---|---|---|---|---|
| 30 | 12/12/12 | $4.7262 \times 10^{-5}$ (0.007618%/$17s$) | $2.1675 \times 10^{-5}$ (0.004204%/$17s$) | $1.1173 \times 10^{-4}$ (0.006931%/$29s$) |
| 30 | 10/10/10 | $6.8512 \times 10^{-5}$ (0.011015%/$10s$) | $3.485 \times 10^{-5}$ (0.0067868%/$10s$) | $1.5837 \times 10^{-4}$ (0.0098241%/$29s$) |
| 30 | 8/8/8 | $3.3076 \times 10^{-4}$ (0.053195%/$13s$) | $1.2532 \times 10^{-4}$ (0.024352%/$13s$) | $7.0777 \times 10^{-4}$ (0.043444%/$13s$) |
| 30 | 6/6/6 | 0.059031 (9.5929%/$30s$) | 0.014547 (2.8251%/$30s$) | 0.062934 (3.9047%/$30s$) |
| 30 | 4/4/4 | $6.472 \times 10^{-3}$ (1.0405%/$10s$) | $3.1211 \times 10^{-3}$ (0.60821%/$9s$) | 0.01293 (0.7928%/$10s$) |
| 30 | 3/3/3 | $4.3664 \times 10^{-3}$ (0.70669%/$23s$) | $1.9671 \times 10^{-3}$ (0.38153%/$17s$) | 0.010925 (0.67518%/$22s$) |
| 30 | 2/3/2 | 0.056079 (9.1131%/$30s$) | 0.021202 (4.1177%/$30s$) | 0.16509 (10.243%/$30s$) |

Table B.6 Time Integrated $L^2$ Error Norm for $30s$ Trajectory

| Evenly Spaced Snapshots | Modes Retained $\rho/\rho u/E_t$ | $\int_0^{30s} L^2_{err}(t)dt$ $\rho$ | $\int_0^{30s} L^2_{err}(t)dt$ $\rho u$ | $\int_0^{30s} L^2_{err}(t)dt$ $E_t$ |
|---|---|---|---|---|
| 30 | 12/12/12 | $8.1218 \times 10^{-4}$ (0.0043714%) | $4.1998 \times 10^{-4}$ (0.0027224%) | $2.1268 \times 10^{-3}$ (0.0043674%) |
| 30 | 10/10/10 | $1.1708 \times 10^{-3}$ (0.0063015%) | $6.5283 \times 10^{-4}$ (0.0042317%) | $2.9818 \times 10^{-3}$ (0.0061233%) |
| 30 | 8/8/8 | $3.5855 \times 10^{-3}$ (0.019298%) | $1.7391 \times 10^{-3}$ (0.011273%) | $8.0705 \times 10^{-3}$ (0.016573%) |
| 30 | 6/6/6 | 0.34564 (1.8603%) | 0.087781 (0.56901%) | 0.37201 (0.76395%) |
| 30 | 4/4/4 | 0.091443 (0.49217%) | 0.046124 (0.29898%) | 0.22802 (0.46825%) |
| 30 | 3/3/3 | 0.083169 (0.44764%) | 0.036043 (0.23364%) | 0.20119 (0.41315%) |
| 30 | 2/3/2 | 0.49607 (2.67%) | 0.23757 (1.54%) | 1.5136 (3.1083%) |

## *Appendix C. Rocket Nozzle Flow Control Using a Reduced Order Fluid Dynamics Model*

This appendix documents an additional application that was investigated during the course of this dissertation research. The manuscript comes from an accepted conference paper that is currently in review with the AIAA Journal of Guidance, Navigation and Control.

*Nomenclature*

| | |
|---|---|
| $A(x)$ | Nozzle cross sectional area, $m^2$ |
| $A$ | Jacobian of flux terms $E$ in Euler equations |
| $E$ | Vector of X-axis fluxes |
| $E_T$ | Total energy, Joules |
| $\underline{G}(t)$ | System dynamic response function |
| $H$ | Numerical flux function from Roe's scheme |
| $\dot{m}$ | Mass flow rate, kg/sec |
| $N, M$ | Dimensions of reduced order mapping matrix |
| $nx$ | Total number of grid points in the nozzle |
| $P$ | Pressure, Pa |
| $R_g$ | Ideal gas constant, J/kg $\cdot$ K |
| $R$ | Transformation matrix |
| $R()$ | Right-Hand-Side or nonlinear portion of Euler equations |
| $t$ | Time, sec |
| $S$ | Matrix of flow field data, or snapshots |
| $T$ | Temperature, K |
| $T(t)$ | Thrust, Newtons |
| $\underline{u}(t)$ | Vector of control deflections |
| $u$ | Fluid velocity at a single spatial location in the flow, m/s |
| $U$ | Vector of conserved flow variables |
| $V$ | Matrix of singular values of $S$ |

| | |
|---|---|
| $\underline{w}$ | Represents any single fluid variable |
| $x$ | Spatial coordinate along axis of nozzle, m |
| $\underline{x}(t)$ | Spatial state vector of fluid variables |
| $\underline{X}$ | Vector of X-axis positions $x$ in nozzle, m |
| $Z$ | Vector of forcing terms for quasi 1-D nozzle Euler equations |
| $\gamma$ | Ratio of Specific Heats |
| $\Lambda$ | Matrix of eigenvalues |
| $\lambda$ | Ratio of time and spatial step size |
| $\rho$ | Density, kg/m$^3$ |
| $\Psi$ | Reduced order mapping matrix |

*Subscripts*

| | |
|---|---|
| 0 | Initial value |
| *amb* | Ambient |
| $c$ | Reference signal for control |
| $i$ | Index to increment discrete time steps for controller time scale |
| $j$ | Index to increment discrete spatial steps |
| $OL$ | Open loop |
| *opt* | Optimal |
| *stag* | Stagnation |
| $t$ | Time derivative |
| $x$ | Spatial derivative |

*Superscripts*

| | |
|---|---|
| $*$ | Value at nozzle throat |
| $n$ | Index to increment discrete time steps for fluid time scale |

## *C.1 Introduction*

Fluid dynamics are governed by a set of Partial Differential Equations (PDEs) known as the Navier-Stokes (NS) equations (45). Since current control synthesis methods almost exclusively tailored for Ordinary Differential Equations (ODEs) in time as plant models, fluids/controls research currently being reported in the literature approximates the NS equations with ODEs. These ODEs are carefully designed to capture the critical dynamics in the flow field, yet facilitate a tractable control synthesis via conventional methods (60, 61). While this approach has yielded results for specific flows (60, 61), few control applications have been demonstrated.

Fluid dynamics modeling is problematic because no general solution to the NS equations is known to date. While there are closed form solutions to a handful of flow fields, most of these solutions require assumptions that are not valid for problems of interest. In the past, aerodynamicists used these closed form solutions to educate themselves on the behavior of the NS equations, while actual aerodynamic design relied on wind tunnel testing. More recently, the wind tunnel has been augmented by the use of Computational Fluid Dynamics (CFD). CFD creates a computational wind tunnel where the complete NS equations are solved via computer based numerical approximation. This enables the prediction of stability and control derivatives.

This paper pursues the synthesis of controllers for high-speed, compressible fluid flow problems using on-line CFD based solvers to model the fluid dynamics, and on-line calculations/optimization algorithms for on-line control design. Most flow problems require very large (high order) CFD models to capture the dynamics of the flow. These CFD models are so large that design optimization via iterative search algorithms is infeasible in real time. The use of Proper Orthogonal Decomposition (POD) for order reduction of CFD flow field solutions was introduced in the mid 1990s. This method has been successfully applied to low-speed, non-linear aeroelastic problems (3). POD is a promising tool for use in model based control, and a few low-speed flow control applications with POD have been recently reported in the literature (62, 11, 63, 64).

This paper develops the controller synthesis strategy for a one-dimensional model problem involving high-speed compressible fluid flow. A CFD code is used as the system dynamics model. The dynamics model is integrated into an optimization algorithm, and an open-loop optimal controller is synthesized for the full order system. Next the reduced order fluid model is developed using POD, and a sub-optimal, model-based, open-loop controller is constructed via the POD Reduced Order Model (ROM) of the plant. The performance of the full order and POD ROM-based controllers is compared.

### *C.2 Problem Statement*

In this paper, a thrust controller for a liquid fuelled rocket engine with a variable throat area and adjustable propellant flow rate is designed. The rocket engine dynamics are modeled with a quasi 1-D nozzle flow, which is a compressible, 1-D, inviscid flow field. The 1-D nozzle assumption provides accuracy within about 6 percent of a higher order analysis (65). Flow control is accomplished by changing the nozzle geometry, which entails changing the nozzle throat area and thereby the fuel mass flow rate. This novel control concept has been tested on experimental rockets (65).

This problem is treated with a numerical method applicable to complex problems with no known analytical solution, even though the analytical solution for quasi 1-D nozzle flows is well known and provided in gas dynamics textbooks (66). This general numerical method is intentionally demonstrated on a problem with a known solution to allow for verification of the solution.

The relevant parameters for the chemical rocket engine were taken from the open literature (65). The rocket engine was designed for an optimal thrust of 5000 Newtons at $25,000$ meters of altitude. This yielded the following specifications.

$$\begin{aligned} A^* &= 13.87cm^2 \\ A_{exit} &= 612.5cm^2 \\ \dot{m} &= 1.919\frac{kg}{sec} \\ u_{exit} &= 2605\frac{m}{sec} \end{aligned}$$

The chamber conditions are as follows.

$$\begin{aligned} P_{stag} &= 2.068MPa \\ T_{stag} &= 2800K \\ \gamma &= 1.3 \\ R_g &= 355.4\frac{J}{kg \cdot K} \end{aligned}$$

The rocket engine operates on a simulated ascent trajectory between $10,000$ meters and $25,000$ meters of altitude. The simplified trajectory entails a linear decrease in ambient pressure over a 120 second flight time as in equation (C.1).

$$P_{amb} = Altitude \cdot \left\{ \frac{(P_{25,000} - P_{10,000})}{(25,0000 - 10,000)} \right\} \tag{C.1}$$

The rocket's thrust is given by equation (C.2) (67),

$$\text{Thrust} = \dot{m}u_{exit} + (P_{exit} - P_{amb})A_{exit} \ . \tag{C.2}$$

Without any control input, the nominal thrust along this ascent trajectory is shown in Fig. C.1.

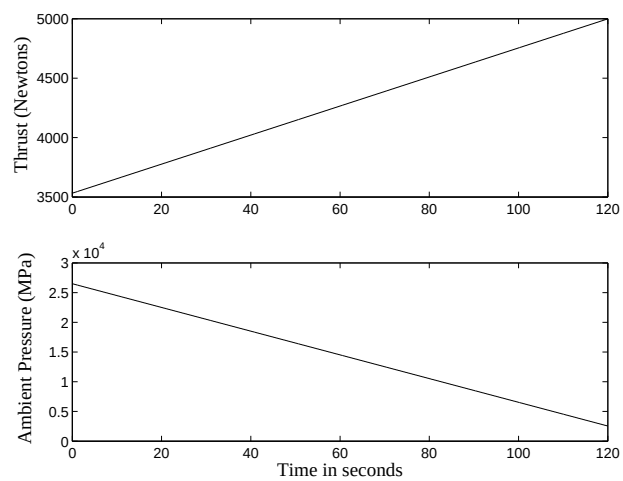



Figure C.1 Thrust From Unmodified Rocket

A schematic of the rocket nozzle is shown in Fig. C.2. The control only affected the nozzle geometry near the throat (shaded region in Fig. C.2), while most of the nozzle geometry (especially the nozzle exit area) remained fixed. No shock was allowed to stand in the nozzle, resulting in isentropic flow throughout the nozzle. Hence the isentropic flow relations yield the analytical solution for this flow field (66). The flow in the nozzle was

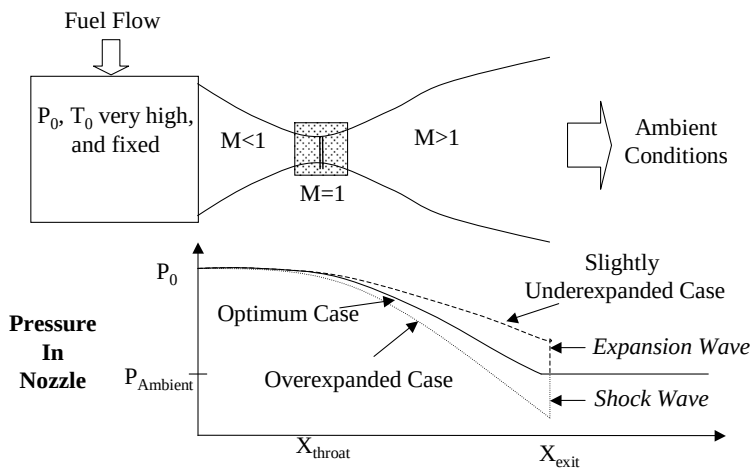



Figure C.2 Rocket Engine Schematic

constant in the uncontrolled rocket, and the increase in thrust throughout the trajectory (Fig. C.1) was due only to the drop in ambient pressure relative to the fixed exit exhaust pressure. Certain launch vehicles need to restrict this increase in thrust with altitude. Such would be the case when launching a g-sensitive payload into orbit. For example, the space shuttle uses fuel mixture ratio control in the combustion chamber of the main engine to restrict launch dynamics to less than 3g (68). For this model problem, an arbitrary thrust profile was selected to be tracked by the controller. The thrust level directly determines the rockets acceleration. Digital control was envisioned and the discrete time variable is $t_i, i = 1, 2, \ldots$ The control problem is described as follows.

The reference command signal is $T_c$:

$$\text{Thrust Desired} = T_c(t_i)$$

The control variable is $A_c^*$:

$$\text{Nozzle Throat Area} = A_c^*(t_i)$$

The nozzle throat area and mass flow rate are related by an equality constraint which enforces the choked flow condition at the nozzle throat. This relationship is given by the 1-D quasi-stationary isentropic flow relation (shown below in equation (C.3)) (67). Therefore, nozzle throat area is the only control variable.

$$\dot{m}_c(t_i) = A_c^*(t_i) \left\{ \frac{P_0}{\sqrt{RT_0}} \sqrt{\gamma} \left( \frac{2}{\gamma+1} \right)^{(\gamma+1)/2(\gamma-1)} \right\} \tag{C.3}$$

The states $\underline{x}(t_i)$ at time $t_i$ are a function of the x-axis position in the nozzle $\underline{X}$ at $nx$ discrete locations,

$$\begin{aligned} \underline{x}(t_i) &= \begin{bmatrix} \rho(\underline{X}, t_i) \\ \rho(\underline{X}, t_i) \cdot u(\underline{X}, t_i) \\ E_T(\underline{X}, t_i) \end{bmatrix} \\ \underline{X} &= \{x_1, x_2, \ldots, x_{nx}\} \ . \end{aligned}$$

Here $\rho$ is density, $\rho u$ is x-direction momentum, $P$ is pressure, and $E_T$ is total energy per unit mass. The output variable of interest (assuming no shock wave exists in the nozzle) is,

$$\text{Thrust} = T(t_i) \ ,$$

and the disturbance signal is,

$$\text{Ambient Pressure} = P_{amb}(t_i) \ .$$

For a pre-specified nozzle geometry, the system dynamics are given by,

$$\begin{aligned} \underline{x}(t_i + \Delta t) &= f\{\underline{x}(t_i); A_c^*(t_i), \dot{m}(t_i)\} \\ \underline{x}(t_0) &= \underline{x_0} \ . \end{aligned} \tag{C.4}$$

The state transition function $f$ is obtained from the inviscid flow equations (also known as the Euler equations). Each time step requires a function call to a CFD algorithm to propagate the state from one time to the next. The states for this system are arranged into a single column vector as shown in Fig. C.3. Notice that the disturbance did not affect the system dynamics since the flow at the nozzle exit was always supersonic. Changes in ambient pressure outside the nozzle cannot propagate upstream in a supersonic flow. The disturbance affects the overall thrust accordingly. Thus, the output variable is

$$T(t_i + \Delta t) = \dot{m} u_{exit}(\underline{x}(t_i + \Delta t)) + (P_{exit}(\underline{x}(t_i + \Delta t)) - P_{amb}(t_i + \Delta t)) A_{exit} \ , \qquad \text{(C.5)}$$

where $u_{exit}$ and $P_{exit}$ are functions of the state vector $x$.

$$\underline{x}(t_i) = \begin{bmatrix} \rho(1,t_i) \\ \vdots \\ \rho(j,t_i) \\ \vdots \\ \rho(nx,t_i) \\ \rho(1,t_i)\cdot u(1,t_i) \\ \vdots \\ \rho(j,t_i)\cdot u(j,t_i) \\ \vdots \\ \rho(nx,t_i)\cdot u(nx,t_i) \\ E_t(1,t_i) \\ \vdots \\ E_t(j,t_i) \\ \vdots \\ E_t(nx,t_i) \end{bmatrix} \begin{array}{l} j = 1 \\ \\ \text{X position} \\ \text{across nozzle} \\ \\ j = \text{nx} \\ j = 1 \\ \\ \\ \\ j = \text{nx} \\ j = 1 \\ \\ \\ \\ j = \text{nx} \end{array}$$

Figure C.3 State Vector of Flow Variables

The CFD fluid model used an explicit Roe scheme solver. The CFD flow model spatially discretized the 1-D nozzle into a grid along its length. For $nx$ grid points, there were $nx$ values of each flow variable computed in the model. The number of grid points was determined by the accuracy and stability of the CFD algorithm. For this problem, the scheme yielded a very nice solution for the convergence divergent nozzle problem with 250 grid points. Therefore, the state vector for the full order system had dimension 750.

*C.3 Controller Synthesis*

Perfect a priori knowledge of the "disturbance" (e.g. the ambient pressure profile) ia assumed along the trajectory of the rocket, and an open-loop optimal controller ia pursued for this problem. The open-loop control system is shown in Fig. C.4. Equation C.6 shows the open-loop controller signal $\underline{u}_{OL}(t_i)$ which is a function of the commanded thrust, the ambient pressure, and time (shown as $\underline{G}_{OL}(t_i)$).

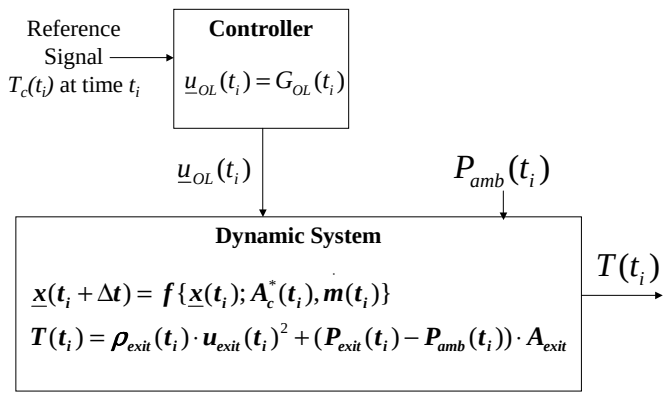



Figure C.4 Open Loop Control Diagram

$$\begin{aligned} \underline{u}_{OL}(t_i) &= \begin{bmatrix} A_c^*(t_i) \\ \dot{m}(t_i) \end{bmatrix} \\ \underline{G}_{OL}(t_i) &= g\{t_i, T_c(t_i), P_{amb}(t_i)\} \end{aligned} \tag{C.6}$$

Finally, the flow field prediction and the myopic control evaluation can be separated in time. The considerable time scale separation between the flow field and actuator response times mean that the actuator dynamics will not affect the stability of the solution, so they are left out for simplicity. The flow field reaches steady state between each time step for the control signal calculation/optimization. Strictly speaking, an otherwise steady flow field (time independent flow field) is disturbed by a step change in geometry ($A^*$ and the portion of the nozzle very near the throat) at each discrete value of $t_i$ in the open-loop simulation. The frequency content of the command signal $T_c(t_i)$ and the disturbance signal $P_{amb}(t_i)$ is low compared to the speed at which disturbances propagate through the flow

field, $\frac{l}{c}$, where $l$ is the length of the nozzle and $c$ is the reference speed of sound. The CFD code is used to perform integration in time and a new stationary (steady state) one-dimensional flow field is quickly established. This new flow field is used to predict the thrust produced with the new $A^*$ geometry. The polarity is obvious: An increase in $A^*$ produces an increase in thrust, while a decrease in $A^*$ produces a decrease in thrust. This physical insight offers an easy interpolation process for matching the predicted (calculated) thrust to the commanded thrust $T_c$. The nonlinear eq. (C.5) is solved when $\dot{m}(A_c^*)$ is found such that $T$ matches $T_c$. This procedure encapsulates the myopic optimization algorithm used to synthesize the optimal controller. The algorithm is outlined in Fig. C.5.

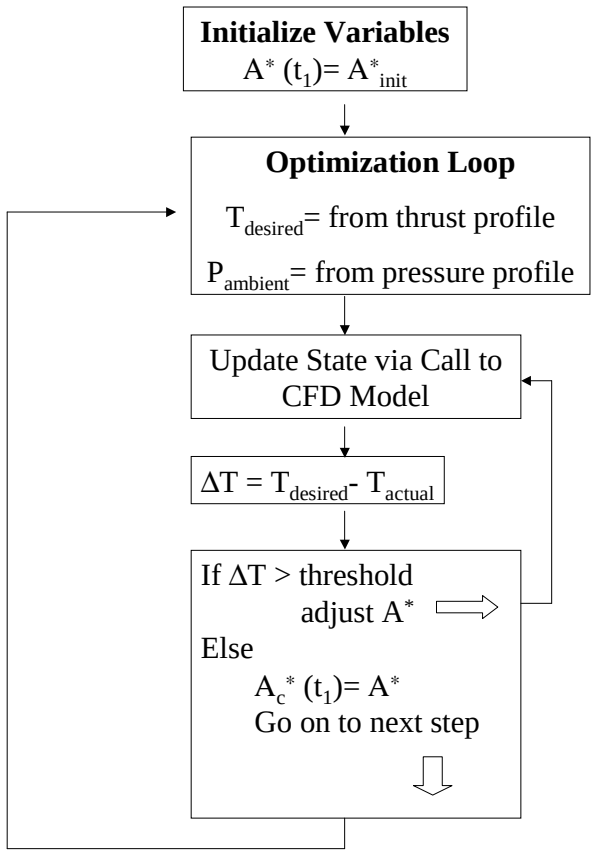



Figure C.5 Solution Process

### *C.4 Fluid Model*

The Euler equations are used to approximate inviscid flow fields. For unsteady one-dimensional flow in a duct of variable (but known) cross-sectional area, the Euler equations (in conservation form) reduce to the equation set given below (52).

$$U_t + E_x = Z$$

$$U = \begin{bmatrix} \rho A \\ \rho u A \\ \rho E_T A \end{bmatrix} \tag{C.7}$$

$$E = \begin{bmatrix} \rho u A \\ (\rho u^2 + P)A \\ (\rho E_T + P)uA \end{bmatrix} \tag{C.8}$$

$$Z = \begin{bmatrix} 0 \\ -P\frac{dA}{dx} \\ 0 \end{bmatrix} \tag{C.9}$$

The nozzle geometry for the unmodified part of the nozzle was selected to be a quadratic area profile (69) where $A^*_{Design}$ was the throat area for the original nozzle design ($13.87cm^2$), and $x^*$ was the fixed $x$ location of the minimum nozzle area, ($5.5456cm$).

$$A(x) = A^*_{Design} + 2.2(x - x^*)^2$$

The quadratic area profile was modified to include a variable throat area as shown in Fig. C.6. This was accomplished by inserting a circular ball of radius $r$, centered on the fixed $x^*$, into the flow field. The ball was lowered into the flow field so that the lower most edge of the ball results in a nozzle throat radius whose area ($A = \pi R^2$) was the throat area commanded by the controller ($A^*_c$). A region near the nozzle throat was created (denoted by $xb$ and $xf$) where the original nozzle configuration was linked to the ball via a line segment that connects the nozzle wall to the nearest tangent point on the ball. This

ensured a smooth surface in the vicinity of the nozzle throat. The nozzle entrance and exit were also extended with a constant area section to ensure no large flow gradients were passed through the boundary. This aided in convergence from poor initial conditions.

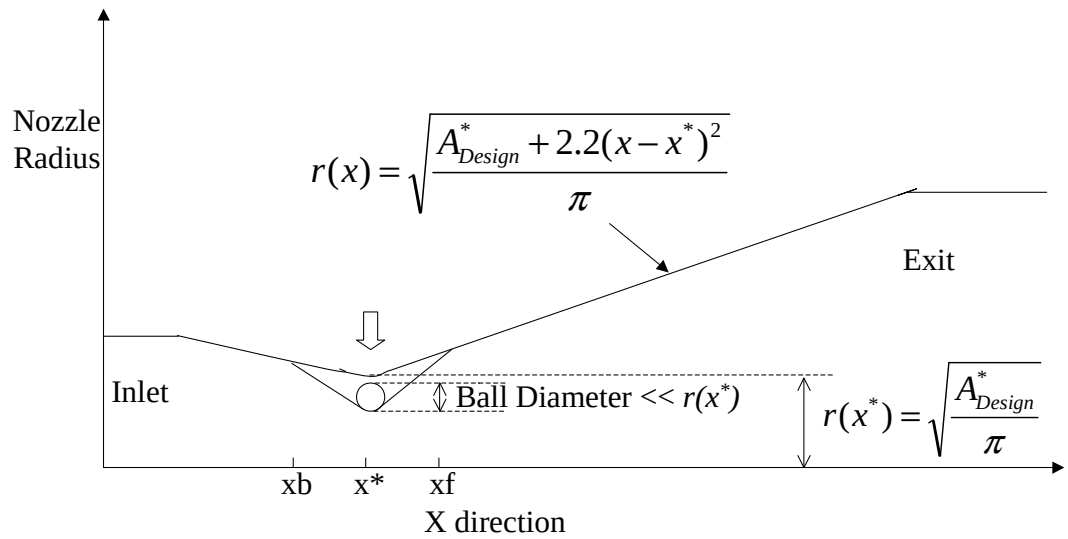



Figure C.6 Variable Throat Nozzle Geometry

Notice that equation (C.9) requires spatial differentiation of the nozzle area, but the nozzle area function lacks smoothness at $xb$, $xf$, and at the transition to the inlet and exit constant area sections. This is accommodated by numerically approximating the spatial derivative of the cross-sectional area via a second order central difference. This smoothed the area function and provided good results.

Roe's scheme was used to solve numerically this system of PDEs. Roe's scheme for the Euler equations is a finite-difference scheme that approximates the flux terms at each discrete grid point with an averaging technique using information at the surrounding points in the flow. See Tannehill et al (45) for a detailed formulation of the Roe's scheme. This finite-difference scheme was used to explicitly solve the entire flow field by marching a known initial condition in time via small time steps. A steady state flow solution was achieved when the maximum norm of the difference between flow solutions at adjacent time steps was less than a small error threshold (point wise convergence).

For the quasi 1-D nozzle problem, the boundary conditions had to be set at the nozzle's inlet and exit. By analyzing the characteristics of the governing flow equations, the appropriate number of terms to specify were determined. The inlet represents propellant flow out of the combustor, and was subsonic. It was appropriate to specify two of the three solution variables for this part of the flow field, which consisted of subsonic flow. Momentum and total energy were specified, letting the density term at the inlet be adjusted by the flow solution. Since a shock was not allowed to form in the nozzle, the exit flow was always supersonic and no information from the ambient environment affected the flow field. The exit condition was simply an outflow of the last grid point's flow solution. It was convenient to extrapolate extra grid points beyond the nozzle inlet and exit. The flow values at the extrapolated inlet and exit grid points were computed using a one-sided second derivative approximation for the velocity variable which was set equal to zero to enforce smoothness in the extrapolated boundary points.

Finally, Roe's scheme loses its dissipation when the eigenvalues of the Roe averaged matrix go through zero. As a result, a non-physical expansion shock can result in the solution, which is indeed the case for the convergent-divergent nozzle solved via Roe's scheme. An expansion shock will form at the nozzle throat unless an entropy fix is introduced. To accomplish this, the eigenvalues in $|\widehat{\Lambda}|$ were checked, and replaced with a finite value $\epsilon$, when they were smaller than a certain threshold (45). The value of this threshold was sensitive to grid size.

### *C.5 Reduced Ordered Modeling*

Reduced order modeling of the flow field was accomplished via Proper Orthogonal Decomposition. A non-Galerkin approach was used (3). The development of POD ROM for inviscid flow through a quasi 1-D nozzle, along with a performance analysis has been recently completed (70). Consider the column vector of flow variables shown in Fig. C.3. This vector represents the state variable in time, and was governed by equation (C.4). For order reduction of the flow field, the state vector was decomposed into the conserved flow variables, which yielded three vectors (one for density, one for momentum and one for total energy). Let the vector $\underline{w}(t)$ represent any one of these three flow variable vectors, then

the Roe's scheme flow solver can be considered as a nonlinear state transition function R acting on $\underline{w}(t)$. Notice that for $\underline{w}(t)$, $t$ now represents time at the fluid time scale.

$$\frac{d\underline{w}}{dt} = R(\underline{w}) \tag{C.10}$$

A linear transformation was sought between the full order state and a reduced order state $\underline{\widehat{w}}$ as follows.

$$\underline{w}(t) = \Psi \cdot \underline{\widehat{w}}(t)$$

$\Psi$ was constructed by collecting observations of the full order state vector at different time intervals throughout the integration of the full order Roe scheme solution. These observations or "snapshots" were collected prior to reaching steady state, since linearly independent snapshot vectors were required. For simplicity, a specific throat area configuration was selected. With this configuration fixed, snapshots were generated that were used to model every flow field of interest to the optimization algorithm (including those generated via different throat area geometries). Collecting snapshots from a variety of throat area geometries proved to be unnecessary for this model problem. The parameter space of interest was the small throat area changes required to affect control for this problem. The variations in the nozzle flow field across this parameter space were small enough to be reasonably represented by the flow structures from a single set of snapshots.

M total snapshots (usually $O(10)$ or less) of the full order state vector length $N$ were collected. These "snapshots" were compiled into a $N \times M$ matrix $S$, known as the snapshot matrix. POD guaranteed that each of the three reduced order flow field variables would yield an optimally convergent representation of the full order variable if the mapping function $\Psi$ was developed as follows.

$$\begin{aligned} S^T S \cdot V &= V \cdot \Lambda \\ \Psi &= S \cdot V \end{aligned}$$

Here $V$ is the matrix of eigenvectors of $S^T S$, and $\Lambda$ is the corresponding diagonal matrix of eigenvalues. Which was used to yield the following reduced order mapping.

$$\begin{aligned}
\underline{w}(t) &= S \cdot V \cdot \underline{\widehat{w}}(t) \\
S^T \cdot \underline{w}(t) &= S^T \cdot S \cdot V \cdot \underline{\widehat{w}}(t) \\
S^T \cdot \underline{w}(t) &= V \cdot \Lambda \cdot \underline{\widehat{w}}(t) \\
\underline{\widehat{w}}(t) &= \Lambda^{-1} \cdot V^{-1} \cdot S^T \cdot \underline{w}(t)
\end{aligned}$$

So inserting this relationship into equation (C.10) and applying a forward difference approximation, the flow model produced the following reduced order flow solver,

$$\frac{d\underline{w}}{dt} = R(\underline{w}) \Rightarrow \underline{w}^{n+1} = \underline{w}^n + \Delta t \cdot R(\underline{w}) \ ,$$

which becomes

$$\underline{\widehat{w}}^{n+1} = \underline{\widehat{w}}^n + \Delta t \cdot \Lambda^{-1} \cdot V^{-1} \cdot S^T \cdot R(S \cdot V \cdot \underline{\widehat{w}}^n) \ . \tag{C.11}$$

Thus, the POD ROM reduced each flow variable from $N$ to $M$, where $M$ was the number of snapshots. Each reduced order variable represented a solution mode, whose contribution to the full order solution was of order with the magnitude of the corresponding eigenvalue. No modal truncation was employed in this implementation. The method of order reduction in equation (C.11) relied on the full order function evaluation at each flow field integration step. As such, the order of each integration step was not actually reduced. However, this reduction technique can greatly increase the time step size allowed for stability. Therefore, the total number of time steps required for the explicit time accurate solver to reach steady state was significantly reduced.

For implicit schemes, where time step size does not affect stability, the governing equations must be projected into the reduced order space (say via a Galerkin projection) to realize a computational benefit from POD ROM. For a complete derivation and discussion of POD as it pertains to fluids, see the text by Lumley et al (19).

*C.6 Results*

For the model problem, a controller was designed to maintain a constant $5000N$ thrust as the rocket flies through the simplified trajectory shown in Fig. C.1. The unmodified rocket was designed for optimal performance of $5000N$ at $25,000m$. The controller was designed to maintain $5000N$ thrust throughout the entire trajectory (the target thrust was actually $5047N$). The nozzle required a larger mass flow rate and nozzle throat area at the lower altitudes to track the thrust profile. In exercising the optimization algorithm, errors were allowed to go uncorrected within $\pm 5N$, which was the convergence tolerance.

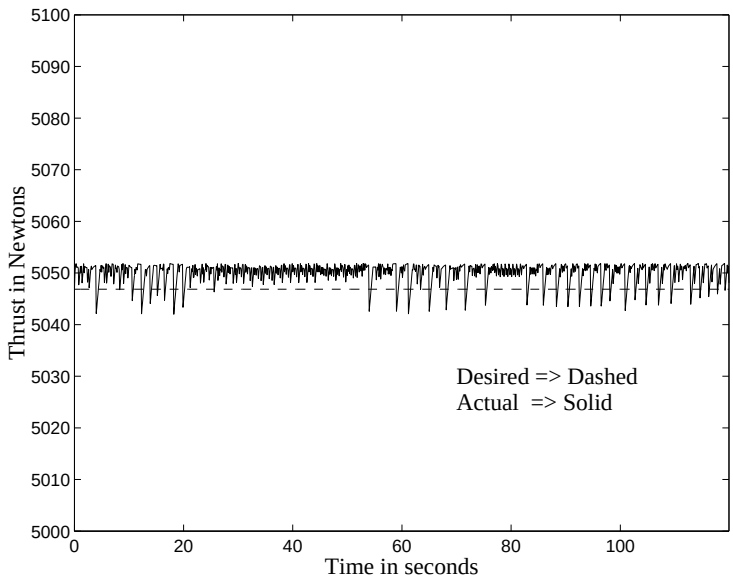



Figure C.7 Optimal Controller Thrust Performance

First consider the controller synthesized from the full order CFD plant model: the controller synthesized using the full order "plant" produced the results shown in Fig. C.7. Clearly the optimal controller had no problem maintaining thrust within the $5N$ threshold. As the rocket ascended and the pressure dropped, the tendency was for the thrust to increase. Once the increase broke the $5N$ threshold, the controller commanded some increment in throat area that brought the thrust down well within limits. This process of a naturally creeping increase in thrust, with a periodic controller initiated decrease in thrust produced the variations shown in Fig. C.7. These dynamics resulted in a slightly biased thrust error, which is shown in Fig. C.8. The bias and standard deviation were

small (less than 1%), and deemed insignificant for the model problem. Of course the solution algorithm could be tuned to provide smoother results, or use a smaller threshold to reduce the bias. The controller actuator commands are shown in Fig. C.9. These show

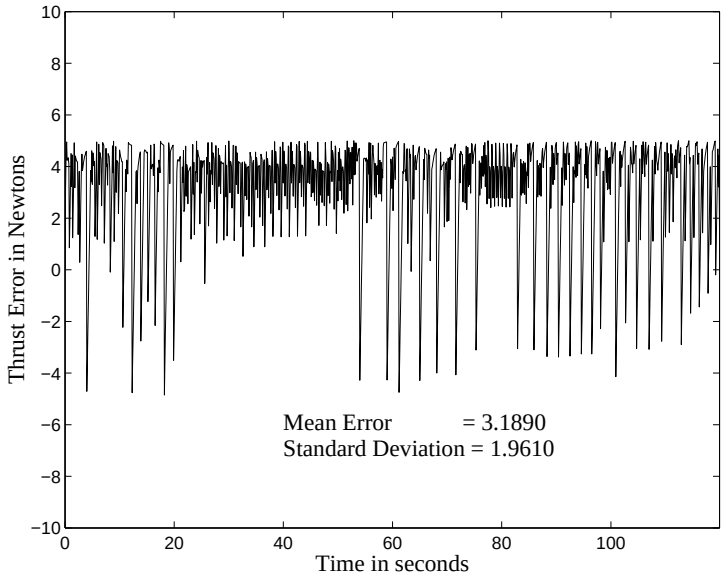



Figure C.8 Optimal Controller Thrust Error

the commanded throat area ($A_c^*$), the resulting nozzle throat radius, and the commanded propellant flow rate. The low bandwidth of the control signal is apparent. Thus, "spill over" is avoided.

The controller started with a larger throat area and mass flow rate, and as the flight progresses to higher altitudes these were reduced via actuator commands. At the terminal point of the trajectory, the controller commanded the throat area and flow rate for the optimum $5000N$ nozzle at $25,000m$ altitude (with a fixed divergent section and exit area). Since the disturbance and desired thrust were linear functions in time, the resulting controller commands were also linear. Note that the commanded actuator dynamics only required about a 7% change in throat geometry, and a 14% change in propellant flow rate to achieve the desired performance.

For problems where the full order model is too large to iterate through thousands of function calls in real time, the performance of a controller synthesized from a reduced order model must be considered. A reduced order model was generated from four snapshots of

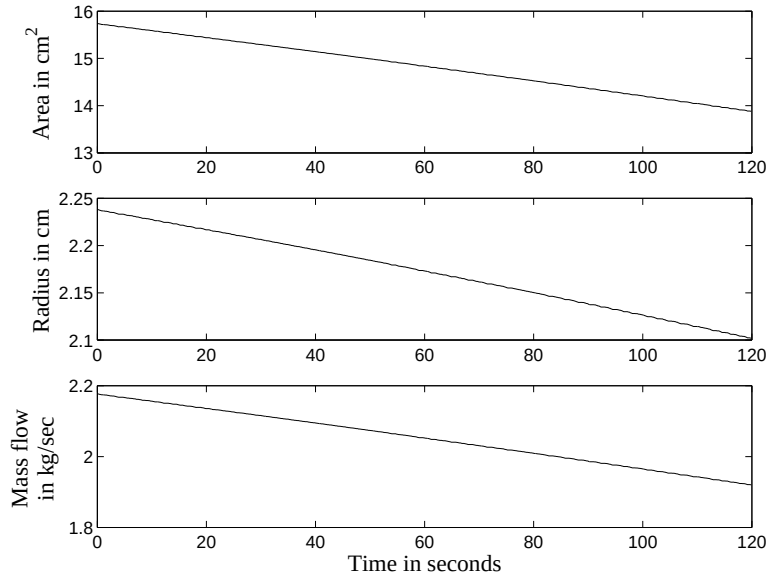



Figure C.9 Optimal Controller Actuator Commands

the full order system state vector, taken at even intervals as the flow solver progressed from initial condition to steady state. The snapshots were obtained from the full order flow solver explicit time integration with the nozzle geometry required at 20 seconds into the flight trajectory (approximate). These four snapshots produced a reduced order model with four modes for each of the three conserved flow variables. This resulted in a total state dimension of twelve. The twelve modes were applied to the unsteady equations as described previously.

The reduced order plant was inserted into the optimization algorithm to replace the full order function call, and the algorithm was rerun to generate a sub-optimal controller. The sub-optimal controller thrust performance, based on the full order plant, is shown in Fig. C.10. The optimization algorithm, based on the reduced order plant, introduced a slowly growing error in thrust. The error is shown in Fig. C.11. The mean error for the sub-optimal controller was $21.8N$ which is a 0.43% error.

Reduced order model accuracy can be adjusted to achieve desired performance goals by varying the number of modes and dispersion of snapshots (3). These improvements were not pursued since the performance shown in Fig. C.11 was adequate to demonstrate the controller strategy. For future implementations involving quasi-steady flow dynamics,

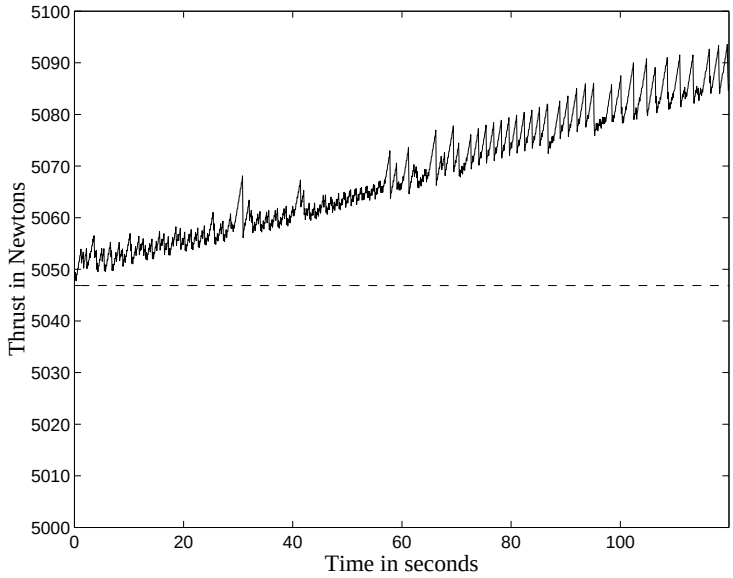



Figure C.10 Sub-Optimal Thrust Performance

a variety of steady state analysis approaches could be used instead of time accurate integration. It has been shown that steady analysis using POD ROM is both efficient and accurate (21). For a steady analysis, steady state flow solutions across the parameter space could be blended (48, 71) into a larger set of snapshots. This approach to data collection would cover a wider parameter space than simply using time integration at one parameter value. In such an implementation, lower energy modes should be truncated to keep the dimensionality small.

### *C.7 Conclusions*

A sub-optimal controller for a high-speed, compressible fluid control problem, albeit without shocks, has been demonstrated using proper orthogonal decomposition based reduced order computational fluid dynamics models. Both an optimal and sub-optimal controller were generated for a quasi one-dimensional supersonic convergent-divergent nozzle with varying backpressure. The optimal controller, generated from a full order plant, tracked the desired thrust profile within an arbitrarily small threshold. The reduced order model yielded a sub-optimal controller that replicated the optimal controller with 0.43% mean error. The reduced order plant had a state dimensionality of 12, versus the full order

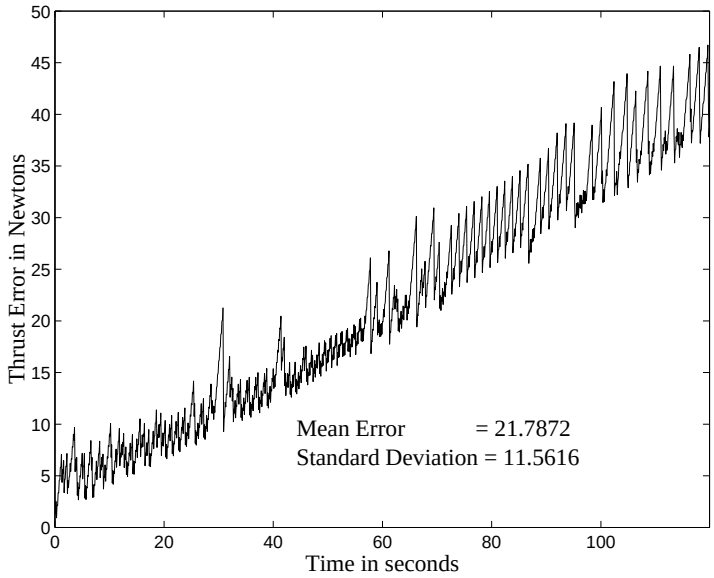



Figure C.11 Sub-Optimal Thrust Error

state dimensionality of 750, which is about a 62 to 1 order reduction. Proper orthogonal decomposition based reduced order models could realize much greater order reductions for problems with full order models requiring higher dimensionality. For example, order reductions for two-dimensional inviscid problems are typically 1000 to 1, and order reductions would be much greater still in three-dimensions.

## *Appendix D. Validation of Two-Dimensional Euler Code*

The steady and unsteady performance of the 2-D, inviscid, flow solver used as the full-system model for this research was validated through comparison of solver results with wind tunnel data, theory, and results in the literature. Wind tunnel data for subsonic channel flow over a five degree bump was used to validate the solvers subsonic and transonic performance. Additional results from a widely used production code, called COBALT, were used to identify differences between viscous and inviscid results. The steady state supersonic flow solution over a 15 degree wedge was compared with the exact inviscid solution to validate the solver's supersonic performance. Next, validation of the transpiration boundary condition is addressed. The aeroelastic panel response is validated by comparison with results from the literature for transonic free stream conditions.

### *D.1 Subsonic and Transonic Validation Cases*

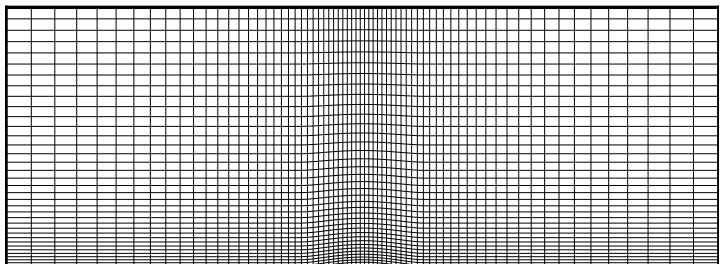
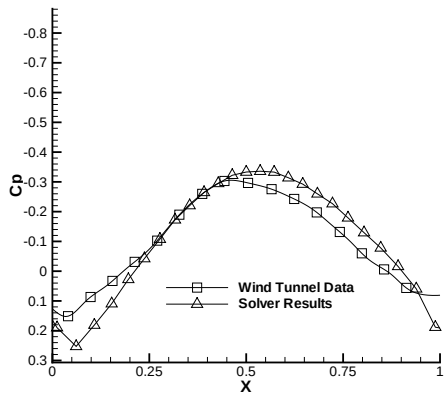


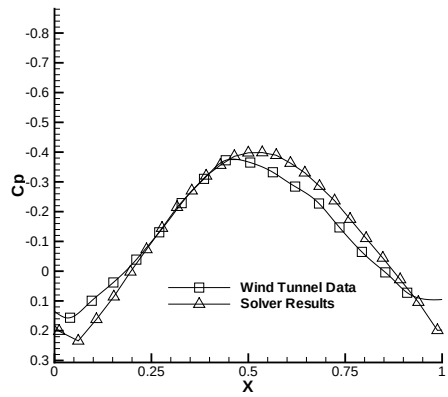
Figure D.1 Grid for Subsonic and Transonic Validation

Subsonic and transonic wind tunnel data was obtained from the literature, consisting of pressure coefficient across the surface of a bump in a channel at varying free stream Mach numbers (72). The bump was a circular arc profile with a height to chord length ratio of 0.05. The channel and bump geometry were taken from (73) and replicated in the 2−D Euler solver. The solver was run to steady state at the Mach numbers corresponding to the wind tunnel data. The computational grid for this problem is shown in Figure D.1.

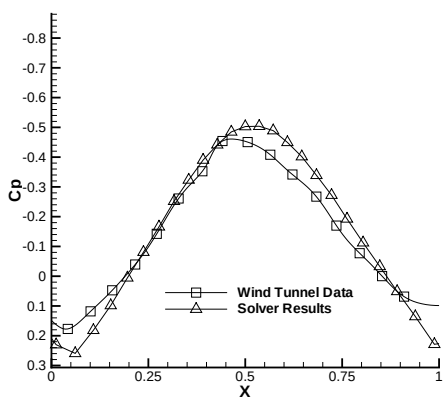
The upper and lower surfaces of the grid represented solid boundaries. The left vertical boundary was subsonic free stream inflow, while the right vertical boundary was subsonic outflow. Three subsonic cases were compared, using free stream Mach numbers




( a. Mach Number 0.226)




( b. Mach Number 0.519)

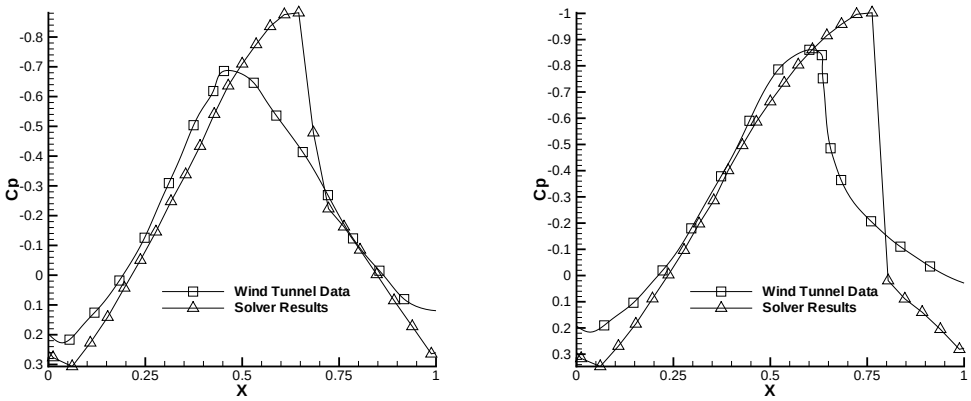



( c. Mach Number 0.680)

Figure D.2 Subsonic Surface Pressure

of 0.226, 0.519, and 0.680. Pressure coefficient comparisons across the bump surface for these three cases are shown in Figures D.2a, b, and c. Reasonably close adherence to wind tunnel data is shown for the subsonic case. Differences between solver and wind tunnel $C_p$ were attributed to viscous effects not modeled by the inviscid Euler code. Specifically, the wind tunnel flow field experienced a small amount of separation on the aft portion of the bump, disrupting otherwise symmetric streamlines in the flow field (72).

In addition, two transonic cases were considered, using free stream Mach numbers of 0.789 and 0.829. Pressure coefficient comparisons across the bump surface for these cases are shown in Figure D.3a, and b. The Mach 0.789 results show that the shock forms at




( a. Mach Number 0.789) ( b. Mach Number 0.829)

Figure D.3 Transonic Surface Pressure

lower Mach numbers for the inviscid Euler solver than for the wind tunnel data. As a result the shock location for the Mach 0.829 from the Euler solver is downstream of the shock location from the wind tunnel data. Notice that the pressure data is in good agreement upstream of the shock. These differences in transonic shock location are attributed to viscous effects in the wind tunnel data not modeled in the inviscid solver. Specifically, the wind tunnel flow field experiences separation on the aft section of the bump at transonic Mach numbers, resulting in a large turbulent wake affecting shock location and strength (72).

Such shock location differences are typical for inviscid CFD codes relative to viscous data. A comparison between the inviscid research code and data from a validated Euler code called COBALT is shown in Figure D.4 for Mach 0.789. The small upward spur at x= 0.38 in the COBALT data is due to a grid abnormality, it is a non-physical anomaly. The COBALT results were provided by the Graduate School of Engineering at Wright State University, via personal correspondence. Good agreement is shown between this research code and the validated Euler code for the transonic case. This demonstrates that the inviscid research code performs appropriately within the limitations of the inviscid assumption.

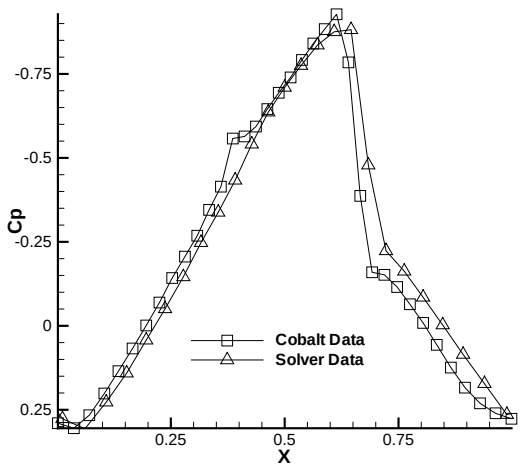
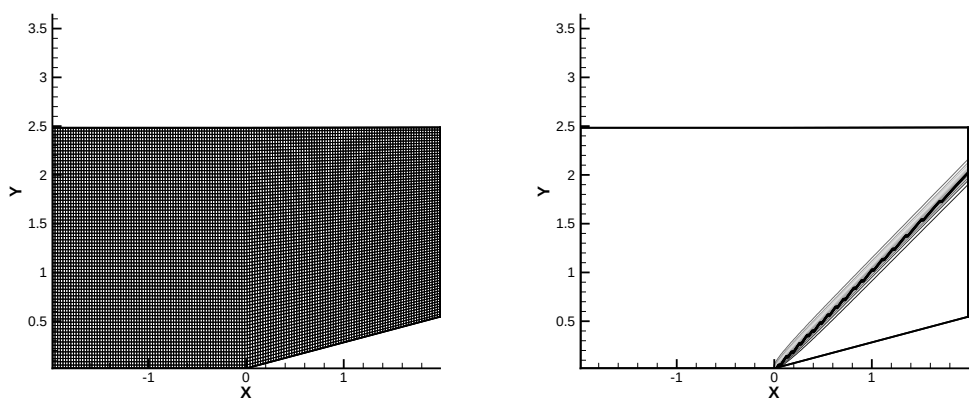



Figure D.4 COBALT Surface Pressure Comparison for Mach 0.789

*D.2 Supersonic Validation Case*

Steady state supersonic flow over a 15 degree wedge at Mach 2 was the test case for supersonic code validation. The Euler code was run to steady state with Mach 2 free stream conditions using the computational mesh shown in Figure D.5.a. The resulting steady state solution was compared to the exact inviscid solution from oblique shock theory found in (74). The results are shown in Figure D.5.b. The dark line represents the exact solution,




( a. Grid) (b. Mach Contour)

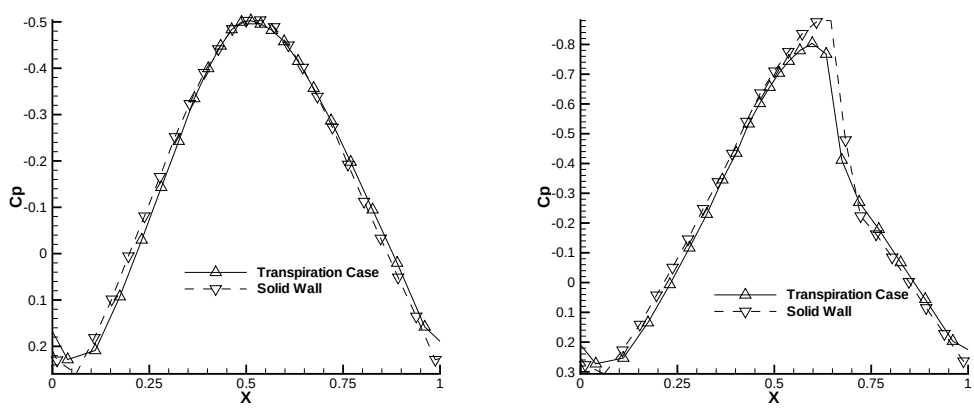
Figure D.5 Supersonic Validation at Mach 2

which is an oblique shock wave attached to the nose of the wedge at an angel of 45.344 degrees. The line is jagged because the exact solution was translated to the nearest grid point in the computational mesh. The contour lines are the Euler code solution superimposed on the exact solution. Notice that the contour lines are centered on the exact solution. This indicates proper shock placement by the Euler code. The first-order Roe solver spreads the shock over about three cells. This, combined with the angle at which the shock crosses cell faces, causes the Mach contour lines from the Euler code to be spread as is shown in Figure D.5.b. A comparison of the solver and exact flow field solutions after the shock are shown in Table D.1. These are in good agreement.

Table D.1 Flow Values Behind Shock at Mach 2: Exact Vs. Euler Code

| | Exact | Solver |
|---|---|---|
| U | 0.786609 | 0.785441 |
| V | 0.210787 | 0.210474 |
| Pressure | 0.391902 | 0.39207 |
| Density | 1.72892 | 1.72812 |
| Mach | 1.44572 | 1.44282 |

### *D.3 Transpiration Boundary Condition Validation*




( a. Mach Number 0.680) ( b. Mach Number 0.789)

Figure D.6 Transpiration Boundary Condition Comparison

The transpiration boundary condition was validated with results for the bump problem at Mach 0.680 and 0.789. The grid from Figure D.1 was replaced with a rectangular

grid, and the bump geometry was enforced with a transpiration boundary condition (47). Surface pressure results from the Euler code using the transpiration boundary condition (TBC) compared well with the pressure coefficients from the Euler code with the bump geometry reflected in the grid. Pressure coefficient comparisons across the bump surface for these cases are shown in Figure D.3a, and b.

### *D.4 Aeroelastic Panel Response*

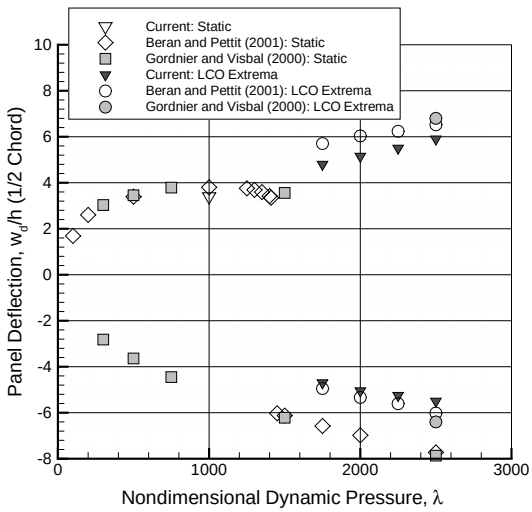



Figure D.7 Unsteady Validation Data

The aeroelastic panel response was validated by comparison with results from the literature using Mach 0.95 free stream conditions for air at sea level ($\mu = 0.1$, $\nu = 0.3$, $h/L = 0.002$, $L = 1$). Panel response as a function of panel dynamic pressure is illustrated in Figure D.7. The maximum panel deflection amplitude at the $\frac{1}{2}$-chord point is shown for a range of $\lambda$. When $\lambda$ was less than 1750, panel stiffness was sufficient to prevent unsteady oscillatory behavior. Any initial panel deflection and velocity eventually damped to a static deflection state, with the panel deflected either upwards or downwards depending on the initial condition. For $\lambda = 1000$, the midpoint deflection for the aeroelastic panel at steady state was 3.4 ($w_d/h$) (for the case of upward deflection). This compared well with the value of 3.8 ($w_d/h$) found by Beran and Petit (9) for this condition.

Values of $\lambda$ above 1750 resulted in oscillatory panel behavior with the proper initial condition, otherwise the panel deflection settled at the downward deflected static solution. Figure D.7 illustrates how the upper static branch evolves into LCO, while the lower branch remains static when $1750 > \lambda > 2500$. At LCO conditions, panel peak deflections at midpoint show a slight deviation with the archived results. With $\lambda = 2500$, Beran and Petit (9) found $(w_d/h)_{max}$ of 6.5, and $(w_d/h)_{min}$ of $-6.0$. The results from Gordnier and Visbal (56) were $(w_d/h)_{max}$ of 6.8, and $(w_d/h)_{min}$ of $-6.4$ for the same conditions. The aeroelastic model used for this research found $(w_d/h)_{max}$ of 5.90, and $(w_d/h)_{min}$ of $-5.5$. Similarly, at $\lambda = 2000$, Beran and Petit (9) found $(w_d/h)_{max}$ of 6.0, and $(w_d/h)_{min}$ of $-5.4$, and the aeroelastic model used for this research produced $(w_d/h)_{max}$ of 5.15, and $(w_d/h)_{min}$ of $-5.05$.

The mid-chord amplitude of the upward panel deflection during LCO is about 15% lower than other cases reported in the literature. The variety of results in the literature varied about 5% by comparison. The dissipation in the first order Roe solver provides more damping of the high frequency panel deflections than the higher order solvers used in the archived data. This additional damping is the primary cause of the muted panel deflections. The muting of the upward panel deflection is not as prominent in the static case.

Another, smaller contribution to the muted panel response came from the solver implementation. Pressure values from the fluid solver were attributed to the centers of each cell. In order to calculate the proper value of pressure at each panel node, the pressures were extrapolated to each node by averaging the adjacent cell centers. Similarly, panel deflections were extrapolated to each cell center by averaging the deflections at the adjacent nodes. These averaging operations smoothed the panel response, introducing an artificial stiffness that also contributed to the muted panel amplitudes.

Similar results were observed when compared with LCO results in the literature for Mach 0.95 with $E = 7.1 \times 10^{10}$, and $\rho_s = 2700$. These conditions (not shown in Figure D.7) translate to $\mu = 0.226851$, and $\lambda = 2392$ when $\nu = 0.3$. Hurka and Ballmann (54) found $(w_d/h)_{max}$ of 6.5, and $(w_d/h)_{min}$ of $-5.81$. The results from Bendiksen and Davis (57)

were $(w_d/h)_{max}$ of 6.2, and $(w_d/h)_{min}$ of $-5.65$ for the same conditions. The aeroelastic model used for this research found $(w_d/h)_{max}$ of 5.5, and $(w_d/h)_{min}$ of $-5.1$.

In terms of the onset of LCO, general panel behavior, and the existence of a transonic moving shock during LCO, the behavior of the aeroelastic model for both static and LCO solutions from the full-system simulation are shown to be in good agreement with results found in the literature for the same problem.

## *Appendix E. Data and Software Archives*

The location and nomenclature of the archived software, data, and results are described in this appendix. Archived files have been provided to the research sponsor (AFRL/VASD, Bldg. 146, 2210 Eighth Street, Wright-Patterson AFB OH 45433-7531) for each of the model problems: quasi 1-D nozzle flow, 2-D blunt body flow, and the transonic aeroelastic panel response. An example run of the aeroelastic panel model is provided to demonstrate proper running of the archived software.

### *E.1 Description*

*E.1.1 Quasi 1-D Nozzle Archives.* The quasi 1-D nozzle archives are located in the zipped file named "1D_nozzle.zip". The executable program is in the "Debug" subdirectory and is called "1D_shock_fit.exe". The source code is provided in files with the extension ".f90". Files with the extension".dat" are either read in by the executable program, or represent results written in a format that can be used to restart the program, feed other parts of the program (such as snapshot collection), or are results for plotting in Matlab. Restart files are denoted "initialize.dat" or "init_vec.dat". The later represents the last flow field output from the previous run. It is overwritten every time the model is executed. The former is a stagnant restart file (typically some desirable initial flow field from an earlier run whose file "init_vec.dat" was renamed), that can be used repeatedly to initialize a run with the same initial data. The number of grid points in the nozzle discretization, and the allocation of grid points to sections within the domain decomposition are set within the source files, and cannot be modified at run time.

*E.1.2 2-D Blunt Body Archives.* The 2-D blunt body archives are located in the zipped file named "2D_Blunt_Body.zip". The executable program is in the "Debug" subdirectory and is called "2D_Blunt_Body.exe". The source code is provided in files with the extension ".f90". Cell centers are allocated to three domain decomposed sections using the Matlab routine "domainMap3.m". This program reads in the ascii files "steadystate.AOA_0.M2.5.dat" and "steadystate.AOA_0.M5.dat" found in the archive, and uses this data to allocate cell centers to sections. The output of the Matlab code is several

files named "MapS1.dat", "MapS2.dat", and "MapS3.dat". These are provided in the archive, so the Matlab routine does not need to be rerun unless a change in the domain decomposition is required.

The program can accommodate up to four arbitrary sub-domains. If the domain decomposition is changed, then the global parameter specifications within "BluntBD_DD_lib" need to be updated to match the number of domains and the values in the first two rows of each "MapS#.dat" file. Snapshot files ("snapshot.dat") consist of flow field data for the entire domain. The creation of reduced order mappings "R_S#.dat" and "Phi_S#.dat" are computed for each section using "snapshot.dat". These files can be read into the program by user command at run time, or they can be recreated as desired.

As for the previous model, restart files are denoted "initialize.dat" or "init_vec.dat". The later represents the last flow field output from the previous run. It is overwritten every time the model is executed. The former is a stagnant restart file (typically some desirable initial flow field from an earlier run whose file "init_vec.dat" was renamed), that can be used repeatedly to initialize a run with the same initial data.

Several subdirectories are included containing the results of various runs. An archive of different grids and domain decomposition mappings are also provided. The executable code and the source code are configured to run the grid and domain decomposition provided in the main directory. Data output from the executable code with the extension ".plt" is configured for the Tecplot graphics package. Tecplot was used to extract pressure coefficients on the surface of the blunt body. These were saved into ascii files which were read into Matlab using the mfiles "CLcalc.m" and "CLCD_data.m". These routines integrate pressure to provide lift and drag coefficients.

*E.1.3 Aeroelastic Panel Archives.* The aeroelastic panel archives are located in the zipped file named "Panel.zip". The executable program is in the "Debug" subdirectory and is called "Panel.exe". The source code is provided in files with the extension ".f90" or ".f". Cell centers are allocated to three domain decomposed sections using the Matlab routine "domain_Pan3.m". This program reads in the ascii file "grid.dat" found in the archive, and uses this data to allocate cell centers to sections. The output of the Matlab

code is several files named “MapS1.dat”, “MapS2.dat”, and “MapS3.dat”. These are provided in the archive, so the Matlab routine does not need to be rerun unless a change in the domain decomposition is required.

The fluid model is contained in “Panel_flow_lib.f90” The program can accommodate up to four arbitrary sub-domains. If the domain decomposition is changed, then the global parameter specifications within “Panel_flow_lib.f90” need to be updated to match the number of domains and the values in the first two rows of each “MapS#.dat” file. Snapshot files (“snapshot.dat”) consist of flow field data for the entire domain. The creation of reduced order mappings “R_S#.dat” and “Phi_S#.dat” are computed for each section using data from “snapshot.dat”. The files provided can be read into the program by user command at run time, or they can be recreated as desired. For the flow model, restart files are denoted “initialize.dat” or “init_vec.dat”. The later represents the last flow field output from the previous run. It is overwritten every time the model is executed. The former is a stagnant restart file (typically some desirable initial flow field from an earlier run whose file “init_vec.dat” was renamed), that can be used repeatedly to initialize the flow field with the same initial data. The flow field can also be initialized with slug flow at a specified inlet mach number.

The structural model is contained in “Panel_structure_lib.f90”. The panel discretization is set in the source code, and cannot be updated at run time. The panel deflection can be initialized as a sine wave. The amplitude is operator selectable at run time. Otherwise, panel restart files can be used to initialize the panel deflection. For the panel model, restart files are denoted “panel.init.dat” or “panel.restart.dat”. The later represents the last panel deflection state from the previous run. It is overwritten every time the model is executed. The former is a stagnant restart file typically represent some repeatedly used initial panel deflection. Panel dynamic pressure ($\lambda$) is operator selectable at run time.

Several subdirectories are included containing the results of various runs. An archive of different grids and domain decomposition mappings are also provided. The executable code and the source code are configured to run the fluid grid and domain decomposition provided in the main directory. The executable code contains a panel model discretized

with 101 equally spaced nodes. Data output from the executable code with the extension ".plt" is configured for the Tecplot graphics package.

### *E.2 Example Run*

To run the aeroelastic panel software, first extract the files from the zip file "Panel.zip" onto a local directory of a computer using Windows 2000. If the software is to be run on another operating system, the source code will have to be recompiled using a Fortran 90 compiler for that system. This program requires access to the IMSL subroutine library. Once the files are unzipped, access the "Panel" directory and execute the program "Panel.exe". The program will prompt the user to enter the grid data file. The default file name is "grid.dat". The source code and ".exe" file are configured for the "grid.dat" provided in the archive. Using a new grid file for any other purpose than running the full order model (options 0 and 1) will require a new domain decomposition, and updates to the source code. Select the default grid file, and the program presents the user with some data about the run, and a sequence of thirteen options.

The full order model can be run using either option 0 to obtain a steady state flow field about a static panel deflection, or option 1 to obtain a time accurate solution, and snapshots of the flow field evolution at evenly spaced intervals. Option 0 is useful for obtaining a preconditioned initial flow filed for the time accurate integration.

Use option 1 to obtain the full order panel response, and collect snapshots for later use. Under option 1, the user is prompted to provide the frequency that data will be recorded during the run. This is the number of explicit time steps skipped between I/O cycles. The user then chooses whether or not the entire flow field or panel deflection state is to be written to file at every I/O cycle. These data files are formatted for Tecplot and can be used for animations. However, the slow down the execution and create very large data files. Regardless of the user response, the panel deflection history at the $\frac{1}{2}$ and $\frac{3}{4}$ chord points are written to "thist.dat" for analysis in Tecplot. The user is then prompted to enter Mach number, dynamic pressure, non-dimensional time for the duration of the time integration, the number of snapshots evenly spaced across the time integration, and CFL number (less than one for stability of the full order solver). The flow field is initialized

next, if slug flow is not desired, then the user has the option to specify the flow field from that last time integration, or the flow field contained in "initialize.dat". Next, the panel deflection state is initialized in a similar manner. The final prompt commands the snapshot file be written to memory automatically at the conclusion of the run. After any option is completed, the initial list of thirteen options is redisplayed for the user to execute another option, or exit the program.

Choosing option 2 will write the snapshots to the file "snapshot.dat". This is not necessary if the automatic write feature was selected for the option 1 run. This will overwrite the old snapshot file, so be sure to rename it if overwriting is undesirable.

Option 3 can be used to read in the existing snapshot file. Two choices of filename are provided, "snapshot.dat" or "snapshot.sect1.dat". This feature allows the use of archived snapshot files.

Options 4, 5, and 6 generate the reduced order mappings. These options require a set of snapshots be previously provided via options 1 or 3. The reduced order mappings are made for each domain section independently. The user is prompted whether or not to view each eigenvalue one at a time (from largest to smallest). Choose this option to customize the number of modes based on eigenvalue size. Otherwise, direct the program to use all the modes without truncation, or select an eigenvalue size below which all modes will be truncated. This suite of options is provided for each of the four fluid variables in each section.

Options 4, 5, and 6 are time consuming, so option 55 allows the user to save the reduced order mappings for future use. The mappings for sections I and II are saved automatically once this option is selected, while the mapping for section III (the shock region) is saved after a user prompt. Typically a different set of snapshots is used for the non-shocked regions of the domain. Old files are over-written by this process, so be sure to rename old files if this is undesirable. Option 56 reads in archived reduced order mapping files produced using option 55.

Once the reduced order mappings are loaded, the reduced order model options 7, 8, and 9 can be executed. Option 7 requires a two section domain decomposition (instead of

three). The source code has to be modified to use this option. The source and executable provided are configured for three domain sections, so use either option 8 or 9. Choosing either of these, the user is prompted for the same inputs described for option 1 above. In addition, the user is prompted to select which model to use in the shock region: POD/ROM or full order. Next, the user is prompted for the CFL number. If the user has chosen a POD/ROM for the shock region, then the CFL number can be as high as 2.5 for stability, otherwise it must be less than 1. After initializing the flow field and the panel, the user is prompted to select the regional time stepping parameters. The first prompt requests a number that is the integer multiple of shock region time steps to skip between updates of the flow field in section II. The size of the time step in section III was determined by the CFL specification made earlier. The next prompt asks the user to specify the integer number of section II updates that will be allowed to pass between updates of section III.

Finally, the user is prompted to specify the update of the Jacobian for section II during the run. Jacobian updates are accomplished using full order function calls, so they are expensive. For the domain decomposition provided, no Jacobian updates are necessary. If the user chooses to update the Jacobian, the program prompts the user for the integer number of section II flow field updates between Jacobian updates.

*Appendix F. Analysis of Domain Size on Panel Response*

The large domain used for the aeroelastic panel problem is shown below in Figure F.1. The large domain was intended to contain all the flow dynamics, and the arbitrary choice of 25 chord lengths in all directions provides the same solution domain used by other authors for the same problem (9, 54). Extending the domain to include all flow dynamics facilitates the use of characteristic boundary conditions for the outer portion of the domain. This appendix explores the effect of reduced domain size on the aeroelastic panel response. For this analysis, section I of the domain was removed, and characteristic boundary conditions were used to model the outlying portions of this much smaller domain.

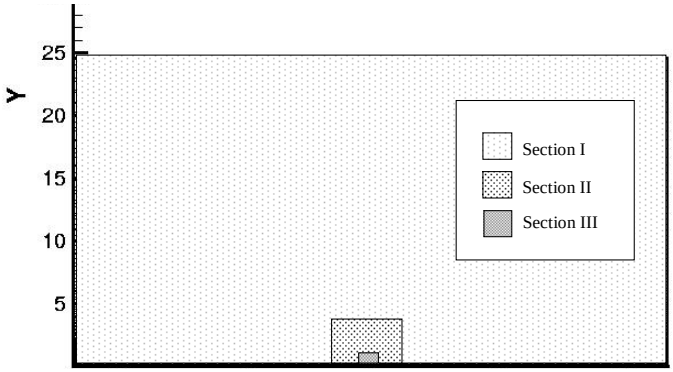



Figure F.1 Panel Problem Domain Decomposition

To evaluate the effect of eliminating the far field, the full system panel response was computed using a 100 node (streamwise) by 50 node (vertical) grid covering three chord lengths fore and aft of the panel, and four chord lengths in the vertical direction. This domain roughly equaled the near field and shock region portion of the larger domain used in the dissertation research. In addition, the number and spacing of grid points were comparable to the number and spacing of grid points contained in the same region of the larger computational mesh.

A time history for 100 non-dimensional time units was computed using the full system simulation, an initial flow condition of slug flow at Mach 0.95, and an initial panel deflection consisting of a sine wave with peak amplitude deflected downward by 0.001. The panel dynamic pressure ($\lambda$) was 2500. An identical time history was computed for the full system model using the larger domain. The results are shown in the Figure F.2.

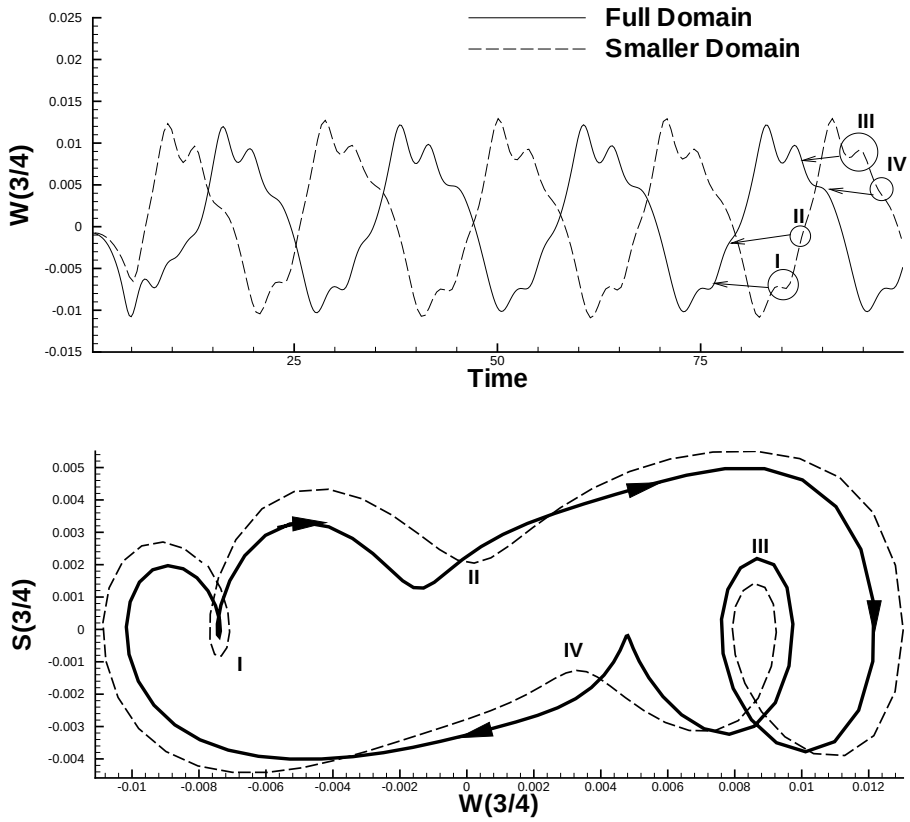



Figure F.2 Unsteady Validation Data

The transient time history results in the phase difference evident in the time history plot of the $\frac{3}{4}$-chord point deflection (top part of Figure F.2). Since the reduction in domain size does not preclude the onset of panel LCO, this difference in transient response is considered minor. However, the LCO that is established with the smaller domain is markedly different than the LCO obtained with the full domain. These differences are marked on the time history and phase plot (lower part of Figure F.2) as numbers I, II, III, and IV. Phase non-linearity (I) is much larger with the smaller domain than the same portion of the full-domain response. This extends the panel oscillation period, while non-linearities (II, III, and IV) reduces the panel oscillation period. The last three errors mute the panel reponse, resulting in an overall reduction in panel oscillation period relative to the full-domain response. More significantly, the amplitude of the panel response is over 10% greater with the smaller domain.

These differences in panel response warrant the additional degrees of freedom contained in the far field section of the domain. Clearly its presence has a significant impact to the panel response. The reduced order model results for both $\lambda = 2500$ and $\lambda = 2000$ (as depicted in Figure 5.8 and Figure 5.9 of the dissertation, respectively) do a much better job of modeling the panel LCO response than the full-system model with a smaller domain. The presence of the far field in both the POD/ROM/DD cases (Full-case and POD-case) clearly makes an important contribution to the overall solution.

*Vita*

Major David J. Lucia graduated from Klein Forest High School in Houston, Texas in May 1984. He entered undergraduate studies at the University of Texas at Austin where he received a B. Sc. degree with High Honors in Aerospace Engineering in May 1988. He was commissioned through the Detachment 825 AFROTC at the University of Texas where he was recognized as a Distinguished Graduate and nominated for a Regular Commission.

Upon receiving his commission, he spent 4 years assigned to the MILSATCOM Terminal Programs Office at Hanscom AFB, Massachusetts, where he worked as a systems engineer and project manager on the Milstar satellite communications system. He next spent two years at Draper Laboratories as a research assistant in Guidance, Navigation and Control, where he pursued his Masters Degree at the Massachusetts Institute of Technology in Cambridge, Massachusetts. He earned his M.S. in Aero/Astro Engineering from MIT in May 1995. Upon graduation, he was assigned the Space Warfare Center in Colorado Springs CO, where he was a space systems analyst specializing in operational modeling and analytical support of the Global Positioning System space segment for Air Force Space Command.

In December 1998, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. His current research activities include reduced order Computational Fluid Dynamics, Control/Fluids/Structures interactions, and design environment development for high-speed flow fields. Upon graduation, he will be assigned to the Air Force Research Laboratories.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
**PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

**1. REPORT DATE** *(DD-MM-YYYY)*
08-11-2001

**2. REPORT TYPE**
Dissertation

**3. DATES COVERED** *(From - To)*
Sept 2000 - Nov 2001

**4. TITLE AND SUBTITLE**
REDUCED ORDER MODELING FOR HIGH SPEED FLOWS WITH MOVING SHOCKS

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**
Lucia, David, J., Major, USAF

**5d. PROJECT NUMBER**
DAGSI Project VA-WSU-99-02

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640
WPAFB OH 45433-7765

**8. PERFORMING ORGANIZATION REPORT NUMBER**
AFIT/DS/ENY/01-03

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
AFRL/VASD
ATTN: Philip S. Beran
Bldg 146, 2210 Eighth Street,
WPAFB, OH 45433-7531
DSN: 785-2521
email: Philip.Beran@wpafb.af.mil

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**
The use of Proper Orthogonal Decomposition (POD) for reduced order modeling (ROM) of fluid problems is extended to high-speed compressible fluid flows. The challenge in using POD for high-speed flows is presented by the presence of moving discontinuities in the flow field. To overcome these difficulties, a domain decomposition approach is developed that isolates the region containing the moving shock wave for special treatment. The domain decomposition implementation produces internal boundaries between the various domain sections. The domains are linked using optimization-based solvers which employ constraints to ensure smoothness in overlapping portions of the internal boundary. This approach is applied to three problems with increasing difficulty. The accuracy and order reduction of the domain decomposition POD/ROM approach is quantified for each application. ROMs with as large as three orders of magnitude reduction in degrees of freedom (DOFs) produce flow fields with maximum errors below 5%. One order of magnitude in computational savings for the non-Galerkin solver implementations accompanies this reduction in DOFs. Finally, the robustness of the reduced order models across a wide parameter space is demonstrated.

**15. SUBJECT TERMS**
Proper orthogonal decomposition, reduced order modeling, moving shock waves, high-speed flow, aeroelastic analysis, domain decomposition, optimization-based solvers

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | King, Paul, I., CIV, (ENY) |
| U | U | U | UU | 181 | 19b. TELEPHONE NUMBER *(Include area code)* (937) 255-6565 ext 4628, email Paul.King@afit.edu |