Reduced-order, trajectory piecewise-linear models for nonlinear computational fluid dynamics.

by

David Gratton

B.S. Ecole Polytechnique de Montreal (2002)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2004

Author .....
Department of Aeronautics and Astronautics
June 1, 2004

Certified by .....
Karen E. Willcox
Assistant Professor
Thesis Supervisor

Accepted by .....
Edward M. Greitzer
H.N. Slater Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Reduced-order, trajectory piecewise-linear models for nonlinear computational fluid dynamics.

by
David Gratton

Submitted to the Department of Aeronautics and Astronautics
on June 1, 2004, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Computational fluid dynamics (CFD) is now widely used throughout the fluid dynamics community and yields accurate models for problems of interest. However, due to its high computational cost, CFD is limited for some applications. Therefore, model reduction has been used to derive low-order models that replicate CFD behavior over a restricted range of inputs, and various frameworks have been developed. Unfortunately, the majority of those methods are limited to linear cases and do not properly handle reduction of nonlinear systems.

In order to overcome restrictions of weak nonlinearity and the costly representation of the system's nonlinearity found in other nonlinear reduction approaches, a trajectory piecewise-linear (TPWL) scheme is developed for a CFD model of the two-dimensional Euler equations. The approach uses a weighted combination of linearized models to represent the nonlinear CFD system. Using a set of training trajectories obtained via a simulation of the nonlinear CFD model, algorithms are presented for linearization point selection and weighting of the models. Using the same training trajectories to provide a snapshot ensemble, the proper orthogonal decomposition (POD) is used to create a reduced-space basis, onto which the TPWL model is projected. This projection yields an efficient reduced-order model of the nonlinear system, which does not require the evaluation of any full-order system residuals, while capturing a large portion of the nonlinear space.

The method is applied to the case of flow through an actively controlled supersonic diffuser. Convergence of the TPWL approach is presented for both full-order and reduced-order cases. The TPWL approach and the POD combine naturally to form an efficient reduction procedure and the methodology is found to yield accurate results, including cases with significant shock motion. Reduced-order PWL models are shown to be three orders of magnitude more efficient than the nonlinear CFD for simulation of a representative test case.

Thesis Supervisor: Karen E. Willcox
Title: Assistant Professor

Acknowledgments

A number of peoples contributed in various ways to the last two years of my life here, in Boston, and made them more than enjoyable and memorable.

Foremost on my list is my advisor, Karen, who helped me more than anything else in getting through MIT. Even though a communication problem initially restricted the flow of information between us, and despite the fact that I thought for a long time that she was English, she has always shown me a great deal of patience. Also, the availability, guidance and responsiveness she provided during all that time went beyond any expectations I possibly could have had for a research advisor. I will never thank her enough for the opportunities to go to Rotterdam and Singapore.

Thanks to Jim Paduano and all the member of the supersonic project for their initial interest toward me and allowing me such an easy switch to model order reduction. Thanks to Mark Drela who provided me with precious hints in debugging *mtflow*. I must also acknowledge Michal Rewienski and Alexandre Megretski for the numerous and fruitful insights on model order reduction.

I thank the ACDL student researchers for contributing to such a nice working environment. Thanks to Garrett, for the grateful nickname he provided me in the lab, for his constant appreciation of my English speaking and mostly, for all the debugging and computer tips. Victor, who helped me understand better the world of computer. Mark, for his unique sens of humor and consistency in using it. Vivian, for all the nice conversations we had. Thanks to Jerome and Yann, who defended with such friendliness their French culture. Hector, for all the times he tried to get me out vainly during the last semester, but mostly, for his contribution to my learning of the Spanish language. Ryan, who I still struggle to understand perfectly even after two years, for all his insights on baseball and spider. A special thanks to Matthieu, for his numerous and mostly fruitful recommendations on whatsoever I was doing, but not for the Paris-French intonation I now bear with such indignation in front of my French-Canadian friends.

Outside the lab, I especially am indebted toward Chris, who helped me survive, even though in a somewhat abusive fashion and focused on a constant pursuit of my liver destruction. Matt for consistently keeping me alert of how bad my English was. Thanks to the beach volleyball crew, for all the good time we had on the East campus sand court. Thanks to the FlyingSquirrels team for helping me catching up in a sport I should have practiced way much more in the past. Thanks to the crew team for all the good time we had on the river, but not for the early practice schedule. Denis, for his patience and unique sens of courtesy in teaching me all the various aspects, subtleties and beauties of the world of finance.

All my parents, Jacques, Francine, Andree and Christian, deserve a special acknowledgment for their constant support and encouragement. And finally, Trang, from whom I already have spent too much time away, for her patience.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Computational fluid dynamics (CFD) is now widely used throughout the fluid dynamics community. It produces accurate models for many problems of interest in aerodynamic design and analysis. However, in order to ensure accuracy, CFD requires that the flow domain is finely discretized in critical regions. Therefore, the cost of obtaining solutions rapidly becomes prohibitive for nonlinear unsteady problems, even for today's computers.

Applying flow linearization is a first step toward resolving this issue [20]. If the unsteady disturbances are small and the unsteady solution can be considered to be a small perturbation about a steady-state flow, then the nonlinear CFD models can be linearized. Since the computing cost of the nonlinear CFD is mostly driven by solution of the nonlinear equations, which typically involves a large matrix factorization at every time step, a linearized model has greatly reduced computational time due to the fact that only one matrix factorization is required.

While a substantial gain can be obtained from using linearization, limitations remain in applications for which model size and cost are issues. Since the use of CFD has broadened outside aerodynamics into multidisciplinary applications such as aeroelastic analysis [53], even linearized models are computationally too expensive. In such an application, both fluid dynamics and structural dynamics equations are solved in an iterative fashion. If both use high-fidelity dynamical systems, the computational cost rapidly becomes expensive.

Another application where computational cost is an issue is control design. Active flow control has been used for various applications in order to enhance stability of unsteady flow, such as for stabilization of compressor surge in jet engines [15, 39]. Small disturbances in the incoming flow can have large repercussions on the system stability. Active flow control can be used efficiently in design in order to prevent such phenomena. The development of an efficient controller requires that the flow dynamics

are accurately captured. CFD offers the degree of accuracy required, but at a high computational cost. While a substantial gain can be obtained using linearization, the number of states in the system is still too large for control design and for online control implementation. Optimal control design involves solving the Riccati equations with a dynamical system of dimension n. Since the cost of computing these solutions is $O(n^3)$, it is necessary that the equations contain a low number of states.

Model order reduction techniques provide a way to systematically determine low-order models that capture the relevant dynamics of the CFD model while being computationally very efficient. The basic idea is to project the high-fidelity CFD equations onto a set of basis functions which span the flow solution space efficiently.

While model reduction is now a well established approach for large linear systems, addressing the problems that arise for consideration of nonlinearity remains a challenging task. Applying model order reduction directly on a linearized model neglects nonlinear effects, which restricts the range of validity to small perturbations about a steady state. Again, an application such as control design may require a range of perturbations outside this small domain of validity of linearized models. Therefore, both model order reduction and the capacity to handle large deviation from steady-state conditions are often required in addition to the need for computational efficiency.

The research presented here aims to achieve accurate, efficient, low-order models valid beyond the linearization restriction by combining two methods previously developed. The POD is used for model reduction, while a TPWL approach is used to capture nonlinear behavior efficiently. Before describing the approach in detail, the next section presents the background of relevant research in the area of model order reduction for both linear and nonlinear systems, as well as control design.

1.2 Background

1.2.1 Reduced-order linear models

A range of model order reduction algorithms has been developed for large-scale linear time-invariant (LTI) systems. The common approach is to project the governing equations onto a reduced-space basis.

One approach to obtain a projection basis is by computing the eigenmodes of the system [21]. However, solving such an eigen-problem is typically an issue in itself, since it requires computing the eigenmodes of a system with ten of thousands of degrees of freedom even in two dimensions.

Another group of model order reduction algorithms uses Krylov subspace methods to construct the projection basis. Using an orthonormal basis of the Krylov subspace, a reduced-order model is constructed which matches a certain number of moments of the transfer function of the original system. Thus, the transfer function of the

reduced-order system approximates well the original transfer function around a specified frequency, or collection of frequency points [18]. Krylov subspace algorithms are suitable for reduction of large-scale systems because of their low computation cost. A number of Krylov-based model order reduction algorithms have been developed, including techniques based on the Lanczos method [14, 16] and on the Arnoldi algorithm, such as for RLC circuits [48], compressor aeroelastic models [57] or active flow control [33].

Another possibility to obtain a basis is the proper orthogonal decomposition (POD) [7, 50]. This method, first introduced by Lumley [36] in the context of turbulence, is widely used for reduction in CFD applications. It computes a set of empirical eigenfunctions using flow solutions, which are more commonly called snapshots [50]. These snapshots can either be obtained from time domain simulations [45], or derived in the frequency domain by exploiting the linearity of the governing equations [19, 24]. Snapshots of the dual CFD system can also be included in the POD process, yielding an approximate balanced truncation [56]. A reduced-order model is then obtained by projecting the governing equations onto the reduced space spanned by the POD basis vectors. Both eigenmodes and POD methods for reduced-order modeling applications are reviewed by Dowell et al. [11] and a more generic overview of linear methods can be found in Antoulas et al. [2].

A newly developed technique, Fourier model reduction (FMR), resolves some of the issues found in other model order reduction methods of LTI systems [55]. While both POD and Krylov subspace algorithms do not guarantee quality of the reduced-order model, i.e. the resulting model may be unstable and no error bound is available, the formulation of the FMR algorithm preserves stability properties of the original system and provides a rigorous error bound on the reduced-order model output. Also, FMR makes efficient use of the system input-output behavior during reduction, while the other methods only account for the system inputs. FMR, which can be efficiently combined with balanced truncation, provides accurate, stable and reduced-order models for linear CFD applications if the transfer function of the original system is smooth.

However, all the methods just described are applicable to linear problems, which means assuming small disturbances about a given linearization point. When a substantial increase in nonlinearity is observed, an extension to those methods is required.

1.2.2 Reduced-order nonlinear models

Although most of the methods have been developed for LTI systems, a number of the techniques just described, or some of their variations, have been extended to the nonlinear case with varying success.

One approach to generate reduced-order models for nonlinear systems is a polynomial (Taylor) expansion of a system's nonlinearity, and subsequent application of Krylov projection methods [9, 37, 42]. Similar solutions can be found with bilin-

earization of the nonlinearity [42], as well as a Volterra series expansion followed by a suitable projection [28, 29, 47]. However, the main drawback of those methods is that they are limited to applications with 'small' input disturbances, or more generally called weakly nonlinear systems, and that the quadratic and higher order expansion terms are very expensive to compute.

POD has also been applied to nonlinear systems [6, 22, 30, 58]; however, in these applications, even if the problem of weak nonlinearity is solved, the issue of an efficient representation of the nonlinearity in the reduced-order model is inadequately addressed. While the resulting nonlinear models do have a reduced number of states, they still require flux evaluations of the original high-order CFD model. Thus, at each iteration, one needs to reconstruct the full state, compute the full-order flux and then project back the solution. Since most of the computation cost is driven by the flux evaluations, the resulting system is hardly more efficient than the original one when an implicit timestepping method is used [40], if not even worse. This fact was shown in [40], where using a reduced-order model provided small savings for a RC circuit application, while the high complexity of a second example, a one dimensional reaction diffusion problem, demonstrated a higher cost in doing so.

In order to overcome both issues of weak nonlinearity and the costly representation of the system's nonlinearity, a trajectory piecewise-linear (TPWL) scheme is developed in Rewienski [43, 44]. The basic idea is to represent a nonlinear system as a weighted combination of linear models. The linear models are obtained by linearizing the nonlinear system at various points along a training trajectory. By using a weighted combination of various linear models, a broader range of the nonlinear space is spanned compared with using a single model. In addition, the TPWL system allows an efficient representation of the reduced-order model. This technique, combined with Arnoldi reduced order models, has been successfully applied to nonlinear analogue circuits and micromachined devices [43, 44]. Also for this application, a truncated balanced realization (TBR) algorithm has been combined with the TPWL model order reduction approach by Vasilyev [51, 52]. TBR, demonstrated to yield superior accuracy when compared to Krylov-based methods, has guaranteed error bounds and a stability preservation property [49]. The resulting TPWL reduced-order models were shown to be accurate and demonstrated good performance for strongly nonlinear circuits and micro-electro-mechanical systems.

The various problems related to the use of reduced-order nonlinear systems are summarized within Figures 1-1, 1-2, 1-3 and 1-4.

Figure 1-1 presents a depiction of the use of LTI models inside a nonlinear space. While LTI models are economical in computational cost, their limited range of validity can be critical for applications where nonlinearity is an issue. Figure 1-2 presents the technique consisting of a polynomial expansion of the nonlinearity and clearly shows the problem of weak nonlinearity described before. While a broader range of the nonlinear space is spanned, the model is still limited to perturbations around the

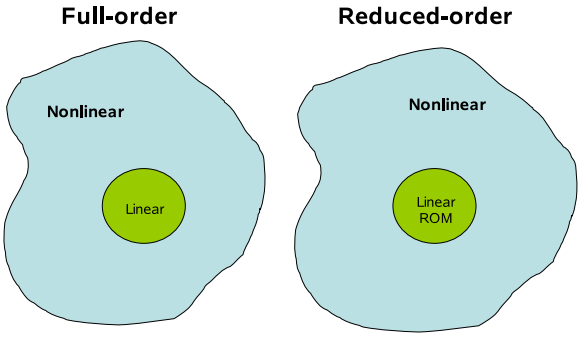

Figure 1-1: Linearization points inside a 2D nonlinear full-order state space (left) and reduced-order state space (right). Circles denote suitable regions for use of each linearization point.

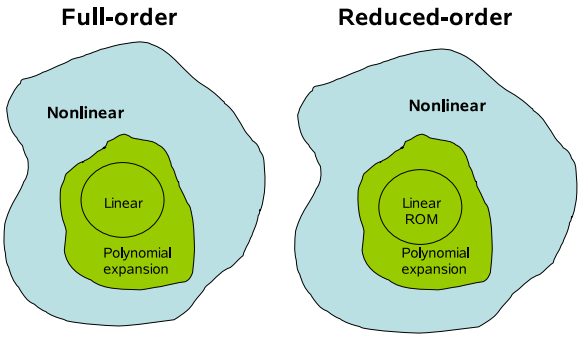

Figure 1-2: Polynomial expansions around linearization points inside a 2D nonlinear full-order state space (left) and reduced-order state space (right). Suitable regions for use of each methodologies are depicted.

initial linearization point.

The second type of reduced-order nonlinear systems described previously, consisting of direct application of a POD basis to the nonlinear system, is depicted in Figure 1-3. Here, two characteristics are emphasized. First, it can be seen that the use of the POD basis reduces the range of validity of the resulting reduced-order model. Also, in order to simulate the reduced-order model, information from the full-order nonlinear space is required. Thus, the resulting reduced-order model, despite spanning a smaller range of validity than the original system, is still inefficient for time simulation.

Finally, Figure 1-4 presents the PWL framework and its relationship to current methodologies. First, both reduced-order and full-order systems are able to capture a wide range of applications inside the nonlinear space through the use of multiple linearization points. Secondly, the resulting reduced-order model is efficient because it can carries out online simulations independently of the full-order systems. Therefore,

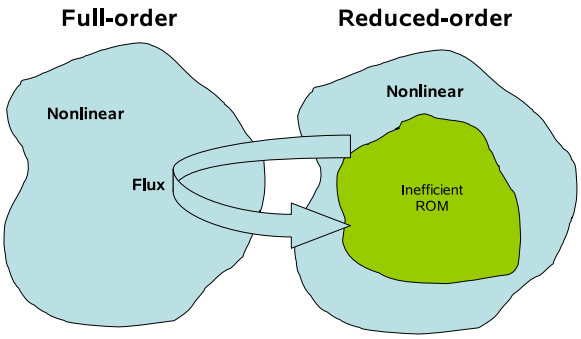

Figure 1-3: Region of validity of a nonlinear reduced-order model using direct projection of the full-order system (right). The arrow depicts the full-order nonlinear information (left) required by the reduced-order model in order to be computed.

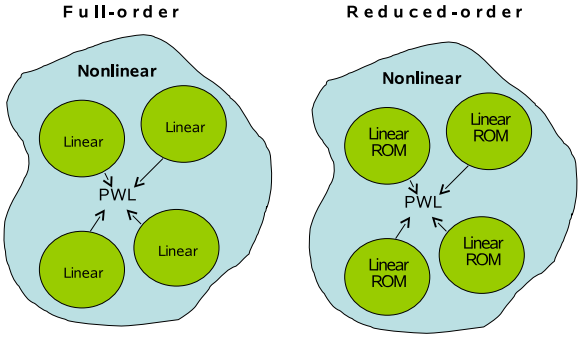

Figure 1-4: Collection of various linearization points inside a 2D full-order nonlinear state space (left) and reduced-order state space (right). Circles denote suitable regions for use of each linearization. In both cases, a piecewise-linear (PWL) set of linearization points is combined to widen the range of validity.

the PWL framework addresses the problem of weak nonlinearity while yielding an efficient representation of the nonlinearity.

This work considers the TPWL approach for a jet engine flow control problem. The TPWL method will be used in conjunction with a single POD basis, and will focus on overcoming weak nonlinearity restrictions while finding a cost-efficient representation of the nonlinearity in the reduced system. The steps toward the creation of a single linearized model are summarized in Figure 1-5. Starting from the full-order model (1), one could either take path (A) by projecting the system onto a reduced-order basis (2) and then applying linearization to obtain a reduced-order LTI model (4), or path (B) where linearization is performed first, followed by projection. Through the combination of multiple reduced-order LTI models (4), a TPWL scheme will be built for control design. The next section describes the background of control design and its close relationship with model order reduction.

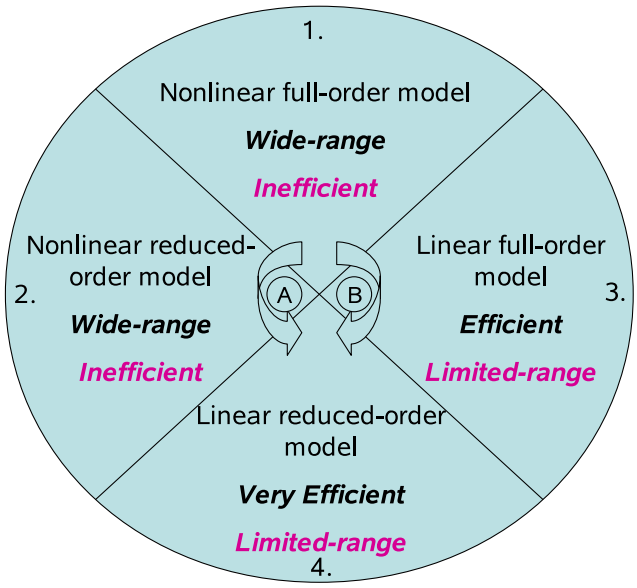

Figure 1-5: Advantages and disadvantages of various representations of the nonlinear space.

1.2.3 Control design

The optimal feedback control of a linear system is a subject which has been extensively studied [1]. With a quadratic functional cost in both state and control, and the assumption of full state knowledge, the optimal control is fully described by a linear state feedback law. The control gains can then be obtained by solving a differential/algebraic Riccati equation. The success of this linear-quadratic regulator problem lies in the development of robust and efficient algorithms for solving the Riccati equation.

When the system is described by nonlinear dynamics, the optimal state feedback law is described in terms of the solution of the Hamilton-Jacobi-Bellman (HJB) equation [35]. This equation provides the solution to the optimal control problem for generic nonlinear systems; however, in most cases, such as active flow control, it is impossible to solve it analytically. This issue led to a motivation in the development of new methods to approximately obtain the solution to the HJB equation.

Some of those methodologies made use of the systems described in the previous section. For instance, Garrard et al. [17] made use of power series in order to extend the linear system for nonlinear applications. Similar solutions using bilinearization of the nonlinearity have been obtained in [8].

Other techniques have also been developed specifically for control purpose and can be classified into two different branches. The first approach makes use of successive approximation, where an iterative process is used to find a sequence of approximations approaching the solution of the HJB. This is achieved by solving a sequence of

generalized Hamilton-Jacobi-Bellman equations (GHJB), and is discussed in [34, 46]. Other discussions of this technique have been performed for more concrete applications, and use a Galerkin procedure to find numerical solution to the GHJB equation [3, 4]. The second technique, the state-dependent Riccati equation (SDRE), is an extension of the Riccati equation to nonlinear systems [10, 54]. Here, the coefficients in the SDRE are functions of the state instead of being constant-valued as in the linear case. However, this approach makes the equation much more difficult to solve and the resulting control only suboptimal. A comparison between some of those methodologies is performed in [5].

The development of various reduction techniques provided another promising approach to perform suboptimal feedback design on reduced-order models. The typical approach, called "reduce-then-design", consists of reducing the model and then designing a controller using standard designs such as linear-quadratic-gaussian (LQG), minmax, or H_∞ [25, 31]. More specifically in fluid dynamics, the "reduce-then-design" approach has been studied for active flow control of linear cases in [33], using Krylov-based models, as well as for nonlinear cases in [13, 32], using POD, and [23], using Lagrange interpolation. Also, an alternative technique, called "design-then-reduce", performs the controller design prior to the reduction process. A benefit to this philosophy is that robust low order systems and insight into sensor placement and design can be obtained [25, 26, 27].

In summary, the methodologies currently used for control design of nonlinear system are far from established. In all cases, the resulting control is either weakly nonlinear and expensive to compute, or suboptimal. The TPWL is an attempt to address those issues, by combining the reduction offered by POD with the efficiency and accuracy of multiple linear systems .

1.3 Outline

This thesis is arranged as follows:

Chapter 2 first introduces the inlet problem in which active flow control is required and thus, model order reduction. The nonlinear CFD model is described, followed by a presentation of the POD reduction framework. The system of equations is then linearized for implementation in the TPWL framework.

The TPWL concept is fully explained in Chapter 3. Two different methods for linearization points generation are included, followed by the weighting procedure description. Then, using the POD reduction framework previously presented, the PWL models are projected onto a reduced-order state space.

Chapter 4 presents results of the application of the TPWL scheme for various sets of full-order linear models. Both methods for linearization points selection are compared. Conclusions on each are drawn, and the most promising method is selected with an appropriate set of linear models for reduction.

Chapter 5 presents the application of the POD to the selected PWL system. First, an analysis of the number of states required to match the full system is performed, for both time and frequency domain snapshot ensembles. Results of the application of the system to multiple disturbances is then presented. In addition, an analysis of the weighting procedure is performed, followed by a computational performance analysis of all nonlinear, full-order and reduced-order methods.

Finally, Chapter 6 presents conclusions and directions for future work.

Chapter 2

Computational Model

The first section of this chapter presents the computational framework for a flow control design inside a supersonic jet inlet. An unsteady formulation that results in a CFD framework to simulate supersonic flows in an engine inlet is described. The nonlinear set of equations obtained will be presented and described in the second section, in which an overview of the CFD model is presented; more details can be found in Drela [12] and Lassaux [33].

Due to computational expense, the nonlinear CFD model cannot be used directly for control design. Thus, one needs to replicate the behavior of the nonlinear set of equations with a reduced-order model. In this work, the POD method will be used. The theory of POD will be reviewed, and an algorithm to obtain the projection basis will be given. The projection of the nonlinear equations onto the POD basis will be presented. Finally, a framework to derive linearized models and their associated reduced-order models will be described.

2.1 Supersonic Inlet problem

The computational model is based on the case of flow through a supersonic diffuser. Figure 2-1 shows the contours of Mach number at steady-state conditions inside the fixed geometry of a supersonic diffuser that operates at a freestream Mach number of 2.2.

In steady-state operation, a shock forms downstream of the throat. In practice, the incoming supersonic flow is subject to perturbations, such as atmospheric density disturbances. Such perturbations in the flow may cause the shock to move upstream of the throat, and eventually to be expelled from the diffuser. This phenomenon, known as inlet unstart, causes huge losses in engine performance and thus is highly undesirable. The common practice to avoid such a phenomenon is to design the inlet in such a way that the shock forms far downstream of the throat. Hence, even if the geometry is fixed, small perturbations will not unstart the inlet. However, a shock

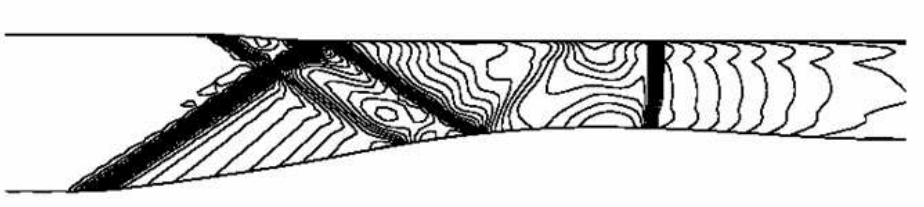

Figure 2-1: Contours of Mach number for steady flow through supersonic diffuser. Steady-state inflow Mach number is 2.2.

further downstream means a substantial increase in shock strength. Since the size of the shock is directly proportional to the pressure losses inside the inlet, a design tradeoff is how to keep a shock close to the throat while avoiding unstart situations.

In [38], an inlet geometry was designed in which the pressure losses were relatively low, i.e. in nominal operation, the shock is located nearby the throat. In order to prevent inlet unstart, an active control mechanism of the shock was designed. Using throat bleeds (mass flow dumping both upstream of the throat and between the throat and the normal shock), the position of the shock can be maintained away from the throat. The bleed flow must be determined in real time using a controller.

Figure 2-2 presents a schematic of the actuation mechanism. Incoming flow with possible disturbances enters the inlet and is sensed using pressure sensors. The controller adjusts the bleed accordingly in order to control the position of the shock and to prevent it from moving upstream. In simulations, it is difficult to automatically determine the shock location. The average Mach number at the diffuser throat provides an appropriate surrogate that can be easily computed. In practice, upstream pressure sensing is used, and the controller adjusts its output, the bleed, accordingly in order to control the position of the shock and to prevent it from moving upstream.

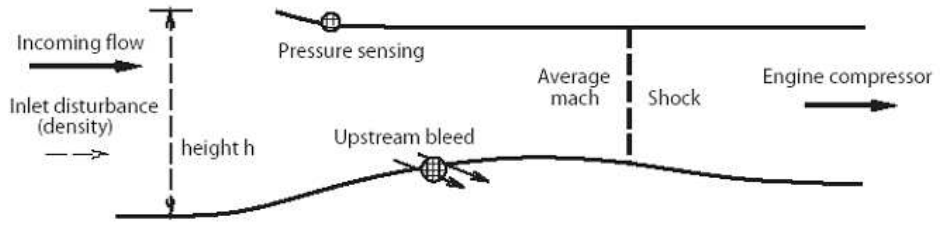

Figure 2-2: Supersonic diffuser active flow control problem setup.

In order to obtain a fully operational controller, one needs to include a number of inputs, such as acoustic waves, air bleed, temperature variation, etc. For simplification purposes, this thesis will focus only on a set inputs, $u(t)$, described by density

perturbations. The output of the system, $y(t)$, will monitor the position of the shock by computing the value of the average Mach number at the throat.

The next section describes the nonlinear set of equations obtained from fluid dynamics relations used by the computational models in order to model the flow inside the diffuser.

2.2 Nonlinear CFD Model

In order to track accurately the position of the shock with a full-order nonlinear model, the solution through the entire inlet must be known at every time during the simulation. The full nonlinear solution in the inlet can be obtained using a CFD model. Here, the problem is assumed to be two-dimensional, compressible and inviscid, thus the solution is governed by the Euler equations. A formulation based on a finite volume method developed on the computational domain showed in Figure 2-3 will be described. The code used was designed by Drela [12] and extended to unsteady cases by Lassaux [33]. First, a steady solver generates a structured grid, whose streamwise gridlines are unknowns of the numerical scheme and are constrained so that they approximate the flow streamlines. By knowledge of the gridline direction and the streamtube mass flow (constant), the flow velocity at each grid point is obtained. Figure 2-3 presents the streamwise gridlines and streamtubes.

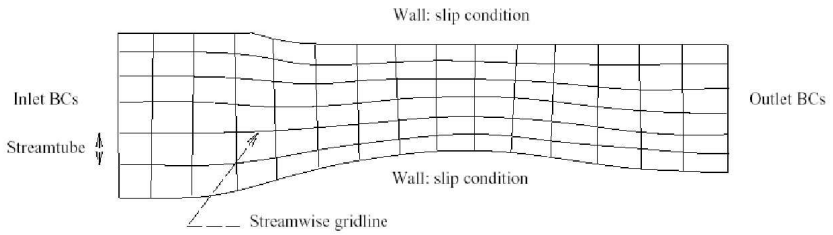

Figure 2-3: Computational domain for finite-volume formulation [33].

Lassaux [33] added the unsteady components required for time dependent analysis. First, the steady solver is used to determine the steady-state solution and steady-state grid. Then, the grid is frozen and during the unsteady calculations, the flow velocity is specified via its projections on both the streamwise gridline, q, and the normal-to-gridline direction, $q_\perp$.

The discrete Euler equations are derived from the integral form of the unsteady, two-dimensional equations, which are the usual statements of mass, momentum, and

energy:

$$\begin{aligned}
\frac{\partial}{\partial t}\iint \rho dV &+ \oint dm = 0 \\
\frac{\partial}{\partial t}\iint \rho \vec{Q} dV &+ \oint \vec{Q} dm + \oint p d\vec{A} = 0 \\
\frac{\partial}{\partial t}\iint \rho E dV &+ \oint H dm = 0
\end{aligned} \tag{2.1}$$

where the flow variables are the density, ρ, the total velocity vector, $\vec{Q}$, the pressure, p, the energy, E, and the total enthalpy, H. The quantity $dm = \rho\vec{Q} \cdot d\vec{A}$ is the mass flux element across the conservation cell boundary, $d\vec{A} = dA \cdot \hat{n}$, where dA is a surface element and $\hat{n}$ is a unit vector pointing outward from the control volume. The discrete Euler equations approximate the integral form of the continuous Euler equations on small control volumes or control cells. The flow solver uses as state variables q, $q_\perp$, ρ, and H, where q and $q_\perp$ are the streamwise and normal components of the velocity $\vec{Q}$, respectively. Hence, the flow velocity is fully specified with the knowledge of the q and $q_\perp$ variables. The location of each of those variables on the grid domain can be seen on Figure 2-4.

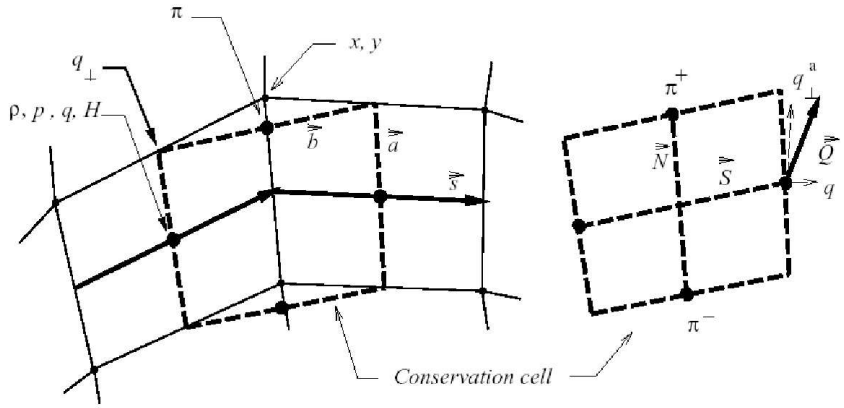

Figure 2-4: Geometric grid (solid lines), conservation cell (bold dashed lines), and variable locations [33].

Notice that in contrast with other state variables, $q_\perp$ is located at the vertices of the cell faces, rather that at their midpoints. Such formulation is convenient for imposing the wall boundary condition on the system [33]. The cell faces $\vec{a}$ and $\vec{b}$ are defined by the geometric grid nodes x and y. $\vec{N}$ and $\vec{S}$, the mean quasi-normal vector and mean streamwise vector, respectively, are obtained from the cell faces $\vec{a}$ and $\vec{b}$. They are used in order to obtain q^a and $q^a_\perp$, the velocities decomposition relative to the conservation cell coordinates. An auxiliary pressure π is also defined at the

midpoints of the $\vec{b}$ faces to distinguish it from the usual pressure p on the quasi-normal faces $\vec{a}$. The finite volume formulation is used to solve for the flow quantities at the midpoints of each of these vectors.

Using a structured grid for spatial discretization, the discrete Euler equations can be represented as a nonlinear dynamical system of the form:

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= h(x(t)) \end{aligned} \tag{2.2}$$

where $x(t) \in R^n$ is a generalized state vector containing the n unknown flow quantities q, $q_\perp$, ρ and H, at each of the L points in the computational grid:

$$\begin{aligned} x(t)^T = [&q_1(t), q_{\perp 1}(t), \rho_1(t), H_1(t), q_2(t), q_{\perp 2}(t), \rho_2(t), H_2(t), \\ &\cdots, q_L(t), q_{\perp L}(t), \rho_L(t), H_L(t)] \end{aligned}$$

f is a nonlinear vector-valued function, $u(t) \in R^l$ is the input to the system, and $y(t) \in R^k$ contains the system outputs, which are defined by the nonlinear function h. The number of inputs, l, and number of outputs, k, are usually small compared with the number of states, n.

However, since such a nonlinear system is computationally expensive to solve, it does not allow controller design. The next two sections will focus on finding a projection basis in order to convert this system into one of much smaller order, while preserving its relevant input-output behavior.

2.3 Proper Orthogonal Decomposition

The POD is a popular approach which creates a basis by extracting the most dominant modes of a given series of snapshots. Those snapshots can be obtained either in the time or frequency domain. In the former, a time simulation of the system for a characteristic unsteady flow is performed and instantaneous solutions (snapshots) are obtained at selected times. The latter obtains the snapshots from solutions of a frequency domain formulation of the governing equations at selected frequencies. In both cases, the snapshots obtained are combined to produce an orthogonal set of basis vectors which represents the solution x in some optimal way that will be described below. The criterion for choosing the basis vectors was first posed in variational form in [50]. The basis vector V is chosen in order to maximize the following cost [7]:

$$\max_{\Phi} \frac{\langle |(x, \Phi)|^2 \rangle}{\|\Phi\|^2} = \frac{\langle |(x, V)|^2 \rangle}{\|V\|^2} \tag{2.3}$$

where (x, Φ) denotes the scalar product of the basis vector with the field $x^{(i)}(z)$, so that:

$$\|V\|^2 = (V, V) \tag{2.4}$$

Also, $x^{(i)}(z)$ is the instantaneous flow field at a time t_i in the geometrical field z, and $\langle\rangle$ represents a time-averaging operation. We can rewrite Equation (2.3) so that we obtain a constrained optimization problem of the form

$$\max_{\|\Phi\|^2=1} \langle |(x, \Phi)|^2 \rangle \tag{2.5}$$

from which we can form the Lagrangian function

$$\ell(\Phi, \lambda) = \langle |(x, \Phi)|^2 \rangle - \lambda[\|\Phi\|^2 - 1] \tag{2.6}$$

where λ is the Lagrange multiplier for the constraint on the norm of the basis vector. Differentiating Equation (2.6) and setting its result to be zero gives that the maximum is obtained with the function V, which is an eigenfunction of the kernel K defined by

$$K(z, z') = \frac{1}{N} \sum_{i=1}^{N} x^{(i)}(z) x^{(i)}(z') \tag{2.7}$$

where the number of snapshots, N, should be sufficiently large. The eigenvectors of K are of the form

$$V = \sum_{i=1}^{N} \Psi_i x^{(i)} \tag{2.8}$$

where Ψ_i are coefficients yet to be determined. Assuming ergodicity, the interchangeability of the time average with the ensemble average, the following may be obtained:

$$C\Psi = \lambda\Psi \tag{2.9}$$

In which λ is the Lagrange multiplier, and R is the correlation matrix defined by the inner product of the snapshot quantities

$$R_{ij} = \frac{1}{N} \left(x^{(i)}, x^{(j)}\right) \tag{2.10}$$

where $\left(x^{(i)}, x^{(j)}\right)$ denotes the inner product between $x^{(i)}$ and $x^{(j)}$.

Equation (2.9) has a complete set of orthogonal eigenvectors with corresponding eigenvalues

$$\Psi = [\Psi^1, \Psi^2, \ldots, \Psi^N]$$

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$$

The following steps provides a guide for the construction of the POD basis V

using the method of snapshots in time domain.

1. Obtain N snapshots $x^{(i)}$ from a CFD calculation, where each snapshot corresponds to a flow solution at a particular instant in time. The time step between each snapshots is usually equally spaced in time, but this is not necessary. Also, this time step should be small enough to capture the important dynamics of the system.

2. Form the correlation matrix R by computing the inner product between every pair of snapshots with Equation (2.10).

3. Compute the eigenvalues λ_i and eigenvectors $\Psi^{(i)}$ of R.

4. The magnitude of the j^{th} eigenvalue, λ_j, describes the relative importance of the j^{th} POD basis vector, V_j, which is computed by

$$V_j = \sum_{i=1}^{N} \Psi_i^{(j)} x^{(i)} \tag{2.11}$$

   where $\Psi_i^{(j)}$ denotes the i^{th} component of the j^{th} eigenvector of R.

5. The orthogonal POD modes V can then be normalized such that

$$(V_i, V_j) = \delta_{ij}$$

   where δ_{ij} is the Kronecker delta defined as follows

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

In order to obtain a much smaller number of states in the reduced-order model, $m \ll N$, the basis V can be truncated using a heuristic criterion. This method is based on the fact that a sufficiently large amount of the "energy" contained in the snapshot collection can be captured by a reduced number of states. The relative energy e_j captured by each mode j is given by the POD eigenvalues as

$$e_j = \frac{\lambda_j}{\sum_{i=1}^{N} \lambda_i} \tag{2.12}$$

The value of m will be chosen such that $\sum_{j=1}^{m} e_j > E$, where E, the total energy included in the basis, is usually set to be 99% to 99.99% of the total energy of the system.

2.4 Reduced-order models

A reduced-order model can be obtained by considering a projection of the state vector x using the basis obtained from POD:

$$x(t) = V\hat{x}(t) \tag{2.13}$$

where $\hat{x}(t) \in R^m$ is the reduced-order state vector, containing the time-dependent amplitudes of m basis vectors, contained in the columns of the matrix V. The basis vectors must be selected appropriately, so that the state x can be accurately represented in the reduced space.

Applying the projection (2.13) to the nonlinear system (2.2), the resulting reduced-order model is of the form

$$\begin{aligned} \dot{\hat{x}}(t) &= V^T f(V\hat{x}(t), u(t)) \\ \hat{y}(t) &= h(V\hat{x}(t)) \end{aligned} \tag{2.14}$$

In order to preserve stability of the fully discrete system, simulation of system (2.14) requires implicit time stepping. Using a second-order accurate backward Euler time discretization scheme, Equation (2.14) simplifies to:

$$\dot{\hat{x}}(t) \approx \frac{3\hat{x}^n - 4\hat{x}^{n-1} + \hat{x}^{n-2}}{2\Delta t} = V^T f(V\hat{x}^n, u^n) \tag{2.15}$$

where Δt corresponds to the time stepping used, and the superscript n, to the time index of the current solution. Therefore, one needs to solve the nonlinear system of equations:

$$F(\hat{x}^n) = 3\hat{x}^n - 4\hat{x}^{n-1} + \hat{x}^{n-2} - 2\Delta t V^T f(V\hat{x}^n, u^n) = 0 \tag{2.16}$$

for $\hat{x}^n$, which is typically done using Newton iteration. Hence, the Jacobian should be evaluated at every Newton iteration inside a given time step to obtain the flux term $f(\cdot)$. Thus, simulation of system (2.14) is no more efficient than the original system (2.2), since most of the computation time is driven by the flux term evaluation [40]. To obtain an adequate reduced-order model for simulations, a more efficient representation of the nonlinearity in the reduced space is required, which is the subject of the next chapter. However, applying linearization is a first step toward resolving this issue, which is described in the next section.

2.5 Linearized Models

Efficient linearized models can be extracted from the system (2.2) by using a polynomial expansion of the nonlinearity, or more specifically a Taylor expansion about some state (x_i, u_i), which, following Phillips [41], expands f as:

$$\begin{aligned} f(x,u) &= f(x_i,u_i) + A_i(x-x_i) + B_i(u-u_i) \\ &\quad + \frac{1}{2}W_i(x-x_i)\otimes(x-x_i) + \ldots \end{aligned} \tag{2.17}$$

where $\otimes$ is the Kronecker product, and A_i and W_i are, respectively, the Jacobian and the Hessian of $f(\cdot)$ evaluated at the state (x_i, u_i). The matrix $B_i = \frac{\partial f}{\partial u}$ is also evaluated at (x_i, u_i). Dropping the quadratic and higher terms of (2.17), the nonlinear system (2.2) can be linearized about any given state to yield a state-space model of the form:

$$\begin{aligned} \dot{x}(t) &= A_i x(t) + B_i u(t) + [f(x_i,u_i) - A_i x_i(t) - B_i u_i] \\ y(t) &= C_i x(t) \end{aligned} \tag{2.18}$$

where $C_i = \frac{\partial h}{\partial x}$ is also evaluated at (x_i, u_i).

The vector of unknowns $x(t)$ can be written as

$$x(t) = x_i + x_i'(t) \tag{2.19}$$

where x_i, fixed in time, is the value of state vector x at the linearization point i, and $x_i'(t)$ contains the perturbation of the n unknown flow quantities about that linearization point x_i.

The linearized equation (2.18) can then be expressed as

$$\begin{aligned} \dot{x_i}'(t) &= A_i x_i'(t) + B_{1i}u(t) + B_{2i} \\ y(t) &= C_i x_i'(t) + C_{0i} \end{aligned} \tag{2.20}$$

where $B_{2i} = f(x_i,u_i) - B_i u_i$ and $C_{0i} = C_i x_i$.

The linearized system (2.20) is efficient for time computations, since a second-order accurate backward Euler time discretization scheme can be implemented as follows:

$$\dot{x_i}'(t) \approx \frac{3x_i'^n - 4x_i'^{n-1} + x_i'^{n-2}}{2\Delta t} = A_i x_i'^n + B_{1i}u^n + B_{2i} \tag{2.21}$$

and can be solved:

$$x_i'^n = [3I - 2\Delta t A_i]^{-1}\Big\{2\Delta t[B_{1i}u^n + B_{2i}] + 4x_i'^{n-1} - x_i'^{n-2}\Big\} \tag{2.22}$$

where I is the identity matrix. The advantage of Equation (2.22) is that only one matrix factorization is required and can be carried out before the simulation. However, due to its large number of states, the system (2.20) still remains too large for applications such as controller design. A reduced-order linearized model can be obtained by applying the projection (2.13) to the system (2.20) yielding

$$\begin{aligned} \frac{d}{dt}\hat{x}_i'(t) &= \hat{A}_i\hat{x}_i'(t) + \hat{B}_{1i}u(t) + \hat{B}_{2i} \\ \hat{y}_i(t) &= \hat{C}_i\hat{x}_i'(t) + C_{0i} \end{aligned} \tag{2.23}$$

where the reduced-order matrices are given by

$$\begin{aligned} \hat{A}_i &= V^T A_i V \\ \hat{B}_{1i} &= V^T B_{1i} \\ \hat{B}_{2i} &= V^T B_{2i} \\ \hat{C}_i &= C_i V \end{aligned} \tag{2.24}$$

Since the terms B_{2i} and $\hat{B}_{2i}$ are constant in time for a given linearization point, they can be considered as acting upon second "inputs" to the systems, which are constant in time and equal to one. Note that if (2.20) and (2.23) were linearized about steady state, i.e. x_0 and $\hat{x}_0$, the value of B_{2i} and $\hat{B}_{2i}$ would be zero.

2.6 Summary

This chapter first described a high-fidelity set of nonlinear equations able to model a complex flow structure over a wide range of inputs. However, such a level of accuracy is only obtained with expensive computation time. Thus, the next section presented the POD methodology, which applied to the nonlinear set of equations, provided a reduced-order model. Again, even though the number of states was reduced, this model was demonstrated incapable of being solved with higher efficiency than the original model. The last section linearized the reduced-order model and provided an efficient and low-order model, but restricted to small perturbations from steady state. Thus, the resulting reduced-order linearized model does not accurately capture nonlinear behavior.

The next chapter will therefore focus on finding a suitable way to capture nonlinear behavior within the reduction framework just described. To do so, a set of linearized models will be combined, and a PWL model will be created.

Chapter 3

Trajectory Piecewise-Linear Scheme

In the second chapter, we defined a flow control problem in which model order reduction is needed to design a controller in order to achieve stability of the engine. The nonlinear CFD has been linearized and then reduced using POD. However, as the nonlinearity in the system increases, such a model becomes inadequate.

This chapter presents a methodology to approximate nonlinear behavior using a set of linearized models described previously. The different steps comprising this methodology are summarized in Figure 3-1. First, using a relevant range of inputs, the nonlinear CFD is simulated. Using the set of snapshots obtained, both POD basis computation and choice of linearization points are performed. The set of linearized models can then be projected to create a reduced-order PWL model. The system obtained can be simulated according to a specified weighting procedure.

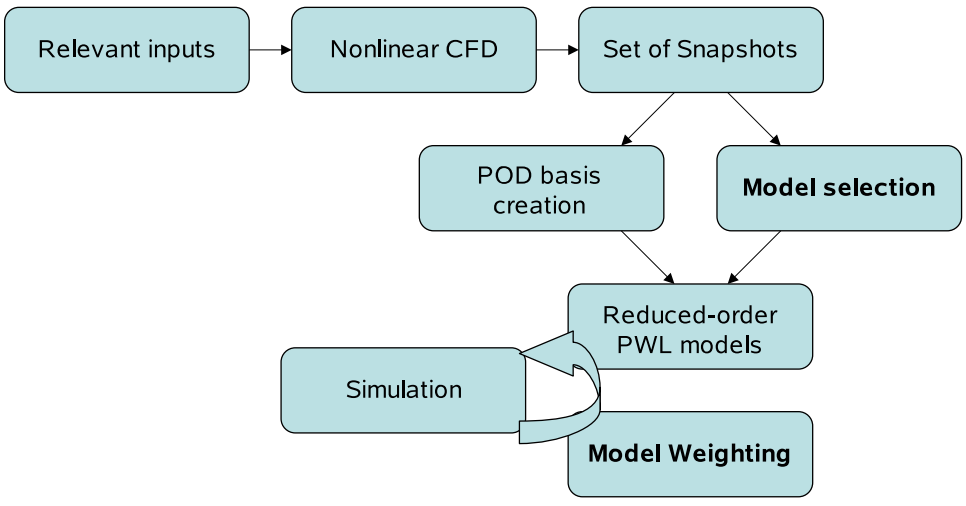

Figure 3-1: Steps in building and simulating a TPWL model.

The first section of this chapter will described the TPWL methodology. The two remaining sections will present the two pieces of Figure 3-1 not yet described, i.e. the model selection, using two different algorithms, and the weighting procedure. Both model selection and model weighting presented are applicable to either the full-order PWL model or reduced-order PWL model.

3.1 Methodology

In Rewienski [43], an efficient, approximate method to represent nonlinear circuit systems is presented and tested. It is proposed that by using a weighted combination of multiple linear models, nonlinear behavior can be modeled. The linear models are obtained via linearization of the nonlinear system at different solutions in time. An approximation to the nonlinear system can then be obtained by using a weighted combination of the closest linear models to the current solution in time.

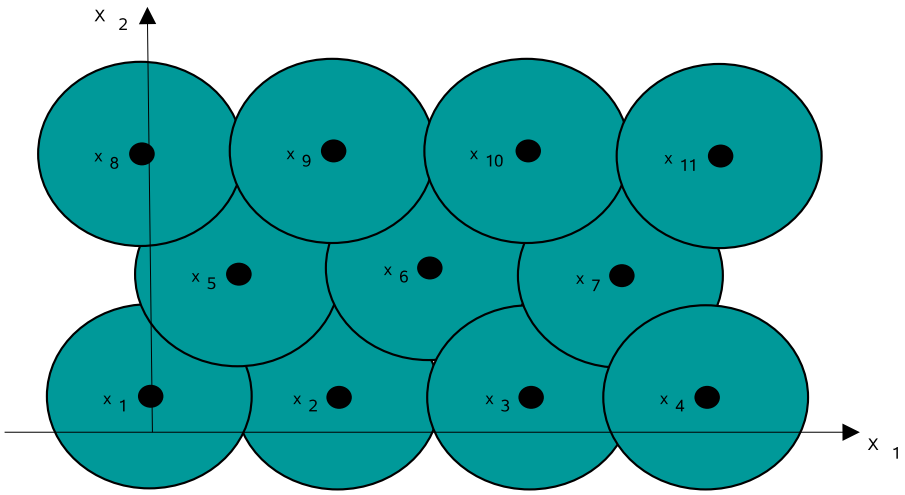

Figure 3-2: Collection of various linearization points inside a 2D nonlinear state space. Circles denote suitable region for use of each linearization point. Figure adapted from Rewienski [43].

Figure 3-2 presents a two-dimensional conceptual view of a series of linearized models. The range of validity of each of the corresponding linearized models is denoted by the circles. Creating such a set of linearized models could become much more expensive than solving the nonlinear equations themselves when either the number of dimensions or the size of the models increases. A more efficient way to create this set of models would be by using a "training trajectory", as shown on Figure 3-3.

Plotted on Figure 3-3 are four linearization points, x_1, x_2, x_3 and x_4, along a "training trajectory", which is obtained through a simulation of the nonlinear system. In

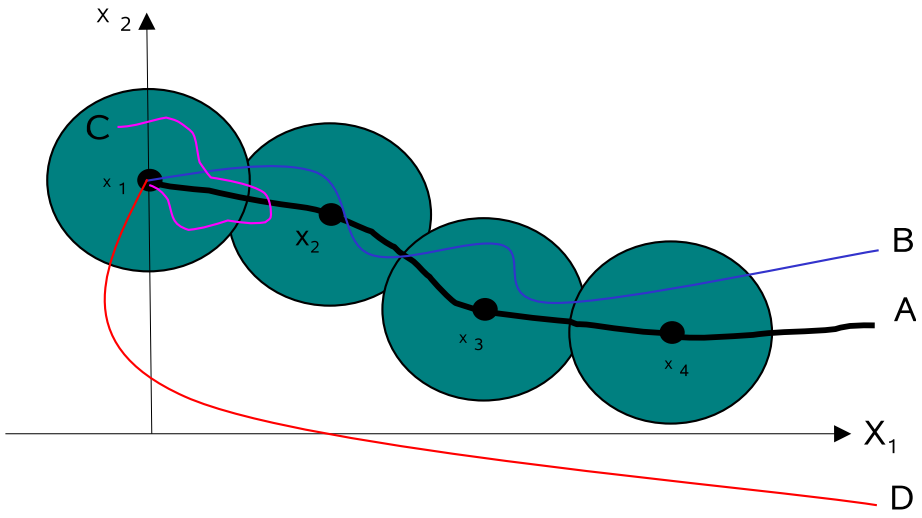

Figure 3-3: Collection of linearization points x_1, x_2, x_3 and x_4 in a 2D state space. Circles denote suitable region for use of each linearization point. Trajectory A is called the training trajectory. Figure based on Rewienski [43].

order to capture the most relevant dynamics of the system, the range of inputs simulated for the training trajectory should reflect dynamics of interest for the application at hand. For instance, in Figure 3-3, trajectories such as B and C will be well represented by the set of linear models, while trajectory D may demonstrate poor results, since it lies beyond the range of validity. The linearization points can be chosen using one of the two approaches defined in the next section.

3.2 Model selection

When traveling along the training trajectory, the selection of new linearization points can be executed in several ways. However, the method used should be able to define an appropriate range of validity for each of the linearized models. The next two subsections address this problem via two different methods. The first focuses on the relative distance between the current state $x(t)$ and a given linearization point x_i, while the second tracks the residual due to the truncation error in the linearization process.

3.2.1 State deviation criterion

The first approach to model selection considers N snapshots, taken from the training trajectory. The algorithm compares each pair of snapshots by computing the two-norm of the distance between them. When this difference is larger than a specified

criterion, δ_{min}, a new linearization point is selected. The value of δ_{min} sets the distance between subsequent linearization points; therefore, lowering its value implies increasing the number of models in the system. This approach is described by the pseudo-algorithm below, which takes as inputs δ_{min} and the matrix $X = \{x^{(1)}, x^{(2)}, ..., x^{(N)}\}$ containing N CFD snapshots, and returns the vector $linPt$, which contains the column index in X of the selected linearization points.

Algorithm 1
(Choice of linearization points, first method)
Function linPt = linearizationPoint1(X, δ_{min})

$N = size(X, 2)$

$linPt = [0]$

for $i = 1 : N$

$k = size(linPt)$

$\delta = \infty$

for $j = 1 : k$

$\delta' = \frac{\|x^{(i)} - x^{(linPt(j))}\|}{\|x^{(linPt(j))}\|}$

$\delta = min[\delta, \delta']$

end

if $(\delta > \delta_{min})$

$linPt = [linPt\ i]$

end

end

3.2.2 Nonlinearity deviation criterion

The second approach considers the same N snapshots. This time, the algorithm compares each pair of snapshots by computing the residual due to the truncation in the nonlinearity expansion, $g(x, u)$. The truncation used in Equation (2.18) implies the following residual about point x_i:

$$g_i(x, u) \;=\; f(x, u) - f(x_i, u_i) - A_i(x - x_i) - B_i(u - u_i) \tag{3.1}$$

One way to monitor the evolution of the nonlinearity is to compare the deviation of $g_i(x, u)$ of each snapshot with respect to steady-state conditions, (x_0, u_0). Thus,

Equation (3.1) becomes:

$$g_0(x,u) \;=\; f(x,u) - f(x_0,u_0) - A_0(x-x_0) - B_0(u-u_0) \tag{3.2}$$

The value of g_0 at a linearization point i will be denoted by g_0^i. When one of the 2-norms between a given g_0 and all residuals of linearization points previously saved, g_0^i, becomes larger than a specified criterion, Δ_{min}, a new linearization point is selected. The value of Δ_{min} sets the distance between subsequent linearization points, but this time by monitoring change in the residual due to truncation. The pseudo-algorithm below describes the method, and is using as new inputs the steady state Jacobian A_0, the steady state input matrix B_0, the matrix $U = \{u^{(1)}, u^{(2)}, ..., u^{(N)}\}$ containing the N inputs to the system, and the matrix $F = \{f^{(1)}, f^{(2)}, ..., f^{(N)}\}$ containing the N CFD fluxes corresponding to the snapshots in X. Again, it returns the vector $linPt$, which contains the column index in X of the selected linearization points.

Algorithm 2
(Choice of linearization points, second method)
Function linPt = linearizationPoint2($X, U, \Delta_{min}, A_0, B_0, F$)

$N = size(X, 2)$

$linPt = [0]$

for $i = 1 : N$

$g_0^{(i)} = f^{(i)} - f^{(0)} - A_0(x^{(i)} - x^{(0)}) - B_0(u^{(i)} - u^{(0)})$
$k = size(linPt)$
$\Delta = \infty$
for $j = 1 : k$
$\Delta' = \|g_0^{(i)} - g_0^{(linPt(j))}\|$
$\Delta = min[\Delta, \Delta']$
end
if $(\Delta > \Delta_{min})$
$linPt = [linPt\ i]$
end

end

A comparison between the efficiency in capturing the relevant domain of validity of a linearization point for both methods is performed in the next chapter. The next section describes a procedure to assemble a set of linearization points into a PWL model.

3.3 PWL model weighting

With an appropriate set of linearization points created, a TPWL scheme can be assembled in order to model nonlinearity. Consider a weighted combination of s linearized models of the form:

$$\begin{aligned}\sum_{i=0}^{s-1}\tilde{\omega}_i(x)\Big\{\dot{x}_i'(t) &= A_i x_i'(t) + B_{1i}u(t) + B_{2i}\Big\}\\ \sum_{i=0}^{s-1}\tilde{\omega}_i(x)\Big\{y_i(t) &= C_i x_i'(t) + C_{0i}\Big\}\end{aligned} \tag{3.3}$$

where $\tilde{\omega}_i(x)$ are weights depending on the value of the perturbation about the linearization point x_i. It is assumed that for all x, $\sum_{i=0}^{s-1}\tilde{\omega}_i(x) = 1$. The weights $\tilde{\omega}_i(x)$ are then obtained using the distance $\|x(t) - x_i\|$ between the linearization point x_i and the current solution $x(t)$. The procedure below, following Rewienski [43], ensures that the "dominant" model i is that corresponding to the linearization point x_i that is the closest to the current state of the system. The inputs are the current state x and the matrix $X_s = \{x_1, x_2, ..., x_s\}$ containing the s linearization points.

Algorithm 3
(Weights computation)
Function $\tilde{\omega} = \text{weighting}(x, X_s)$

1. For $i = 1, ..., s$ compute:
   $d_i = \|x(t) - x_i\|_2$.

2. $[m, k] = min\{d_i : i = 1, ..., s\}$.

3. a) For $i = 1, ..., s$ compute:
   $\tilde{\omega}_i = (exp(d_i)/m)^{-25}$.
   or

   b) For $i = 1, ..., s, \tilde{\omega}_i = 0$
   $\tilde{\omega}_k = 1$.

4. Normalize $\tilde{\omega}_i$.

First, Algorithm 3 obtains the difference d_i between the current state $x(t)$ and the linearization point x_i. The minimum distance is given by m and corresponds to the model with index k. Then, the weights can be computed in two different ways. The first method shows a weighted sum strongly concentrated on the closest model, while the second uses only the closest model at the time. As will be presented later, each formulation yields slightly different results. The last step in the algorithm ensures that the summation of the s weights is unity.

3.4 Reduced-order PWL model weighting

With the PWL methodology valid for any set of linearized models, it can be applied directly to the linear reduced-order models obtained previously. Using the TPWL representation of the nonlinear system, an efficient reduced-order model can now be obtained using the projection (2.13) applied to (3.3), yielding a reduced-order TPWL model as follows.

$$\begin{aligned} \sum_{i=0}^{s-1} \tilde{\omega}_i(\hat{x}) \Big\{ \frac{d}{dt}\hat{x}'_i(t) &= \hat{A}_i \hat{x}'_i(t) + \hat{B}_{1i} u(t) + \hat{B}_{2i} \Big\} \\ \sum_{i=0}^{s-1} \tilde{\omega}_i(\hat{x}) \Big\{ \hat{y}_i(t) &= \hat{C}_i \hat{x}'_{0i}(t) + C_{0i} \Big\} \end{aligned} \tag{3.4}$$

where the reduced-order matrices are defined as before in (2.24). As in the linear case, this representation is efficient, since all reduced-order matrices in (3.4) can be precomputed. Note also that the weights $\tilde{\omega}_i$ are computed as a function of the reduced-order state $\hat{x}$. Thus, Algorithm 3 now has as inputs $\hat{x}$ and $\hat{X}_s$, the reduced-order state and reduced-order linearization points matrix respectively. The TPWL approach fits well within the context of POD-based model reduction, since a simulation of the nonlinear system can provide both the snapshots for computation of the POD basis vectors and also a set of instantaneous flow states from which to select the linearization points.

3.5 TPWL summary

The final TPWL reduction approach can be summarized following Figure 3-1. First, simulate the nonlinear CFD model for a range of forcing functions and conditions that are representative of the application at hand. Second, from the resulting snapshot collection, calculate a set of POD basis vectors. Third, from the same snapshot collection, select a set of linearization points using either Algorithm 1 or Algorithm 2. Fourth, using the dominant POD basis vectors, project each linearized model to obtain a set of reduced-order linear state-space systems. Finally, combine these low-order state-space systems using the TPWL representation and a set of weights from Algorithm 3. More implementation details can be found in Appendix A.

This approach will now be demonstrated for the case of flow through the supersonic diffuser shown in Figure 2-2. First, each methodology for the linearization points selection will be discussed in the next chapter, with corresponding results for the full-order TPWL models. Next, in Chapter 5, reduced-order TPWL models will be constructed, and the results compared with full nonlinear CFD outputs will be presented.

Chapter 4

Full-order TPWL results

In order to obtain an accurate set of reduced-order TPWL models, the ability of the full-order set of models to replicate the nonlinear CFD first needs to be tested. The first section of this chapter presents the set of incoming density disturbances used for test cases in this thesis. Then, the next two sections present results of PWL models using both linearization point selection strategies developed in the previous chapter. A recommendation for the most appropriate methodology for the selection of linearization points is made, and a set of models is selected for reduction in the last section.

4.1 Test cases

A number of test cases will be presented to demonstrate the TPWL methodology. In all cases, the input considered is an incoming density disturbance and the output of interest is the average Mach number at the throat of the diffuser. The six different temporal distributions considered for the input are presented in Figure 4-1, and vary temporally either with a Gaussian pulse or a sinusoidal distribution of various frequencies and amplitudes. The Gaussian distribution is described by

$$\rho'(t) = -\Lambda\rho_0 e^{-\alpha(t-t_{peak}/f_0)^2} \tag{4.1}$$

while the sinusoidal distribution is described by

$$\rho'(t) = -\Lambda\rho_0 \sin\omega_0 t \tag{4.2}$$

where the nominal frequency f_0 equals a_0/h, the inlet speed of sound divided by the height of the inlet, $\omega_0 = 2\pi f/f_0$, and the non-dimensional time, t_{peak}, sets the time at which the perturbation peaks. The parameter α sets the sharpness of the perturbation (i.e. its frequency content), Λ corresponds to the amplitude of the perturbation, and ρ_0 is the nominal value of freestream density. In specifying a time step for unsteady

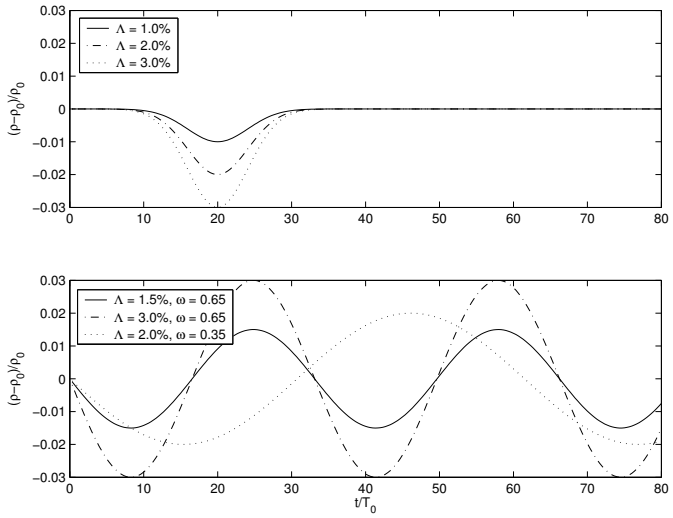

Figure 4-1: Incoming density disturbances. Top: Gaussian distributions. Bottom: sinusoidal distributions.

Case	Λ	t_{peak}	α
1	1%	20	$0.03 f_0^2$
2	2%	20	$0.03 f_0^2$
3	3%	20	$0.03 f_0^2$

Table 4.1: Data used for the Gaussian distribution.

simulations, one period is defined by $T_0 = 1/f_0$. The parameter values corresponding to the six different input functions are presented in Tables 4.1 and 4.2.

Nonlinear CFD results are obtained by simulation of the full system using those disturbances, and snapshots at each time step are saved. The next two sections will describe two different methodologies for the selection of linearization points, following the two algorithms described in the previous chapter. In each section, once the linearization points have been determined, the validity of the TPWL representation must be tested. In order to do so, the outputs of the system, the average Mach number at the throat, for both the nonlinear CFD and the full-order TPWL approximation given by Equation (3.3) are compared. Note that all tests in this chapter are performed using option [b] in step 3 of Algorithm 3, i.e. only the single closest linearized

Case	Λ	ω_0
4	1.5%	0.65
5	2%	0.35
6	3%	0.65

Table 4.2: Data used for the sinusoidal distribution.

δ_{min}	Case number			
	1	2	3	6
∞	1	1	1	1
0.030	1	3	4	7
0.020	2	4	8	16
0.015	3	6	16	31
0.012	4	11	23	41
0.010	5	16	28	50
0.008	6	20	34	70
0.006	12	29	48	100
0.005	15	36	56	118
0.004	20	42	69	147

Table 4.3: Number of models given by different values of δ_{min} for cases 1, 2, 3 and 6.

model at every time step is used. The second option of the weighting procedure will be discussed in the next chapter.

4.2 Linearization points selection using δ_{min} (state deviation criterion)

Using Algorithm 1 for different values of δ_{min} and the snapshots previously obtained, various sets of models are found. Table 4.3 shows the number of models as a function of the choice of δ_{min} for four of the cases, where Algorithm 1 was applied to each case separately. For each case, it can be seen by how much the number of models grows as the distance between linearization points is decreased. By comparing the number of models for a given δ_{min}, one gains some insight to the importance of the state fluctuations during a simulation. For example, a Gaussian distribution of 3% can be seen to introduce a higher degree of perturbation into the system than one of 1%, requiring substantially more models for a given δ_{min}.

The results using different sets of models from Table 4.3 are shown on Figures 4-2 and 4-3, where the variation of the average Mach number at the throat due to the disturbance of case 3 is plotted against time. Here, both the training trajectory and the disturbance were a Gaussian distribution of 3% amplitude.

First, it can be seen on Figure 4-2 that only one linearized model cannot capture the nonlinear behavior of the shock motion. Also, initially, as the value of δ_{min} is decreased, the match between the PWL model and the nonlinear CFD improves. With 28 models ($\delta_{min} = 0.01$), the PWL model provides a reasonable accuracy compared with the nonlinear CFD. However, inconsistent behavior arises when the number of models is increased beyond 28. Figure 4-3 presents the effect of lowering the value of δ_{min} below 0.01. First, when the number of models is slightly increased, some

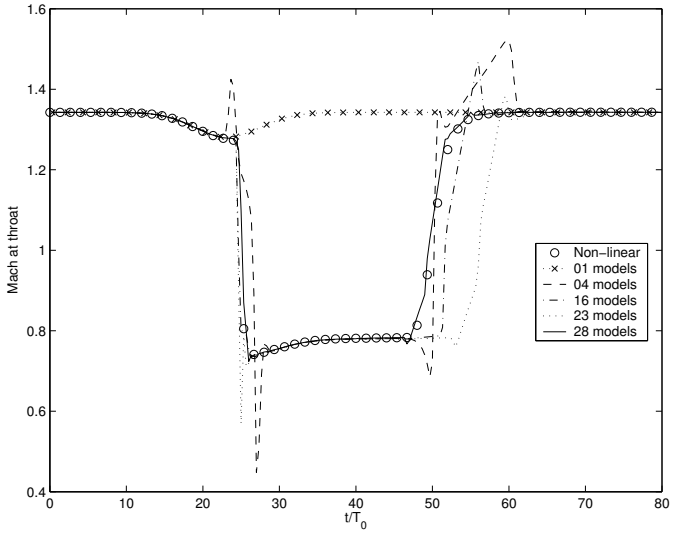

Figure 4-2: Nonlinear response plotted against various TPWL model combinations for a Gaussian incoming disturbance of 3% amplitude. Training trajectory obtained from the same simulation.

oscillations appear around the time when the shock moves back downstream of the throat, i.e. at time $t/T_0 \approx 50$. Then, as δ_{min} is decreased further, the system fails completely to capture the relevant behavior of the CFD. This behavior is caused by an inadequacy in the method of choosing the linearized models. Recall Equation (2.2)

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= h(x(t)) \end{aligned}$$

As can be seen from Figure 4-3, for $t/T_0 < 30$, the input $u(t)$ in Equation (2.2) perturbs the system and has the dominant effect over $\dot{x}(t)$. However, for $t/T_0 > 30$, the input $u(t)$ becomes zero and Equation (2.2) is only driven by the state correction itself ($\dot{x}(t) = f(x(t), 0)$). However, Figure 4-3 shows that large variations in the output are observed even when $u(t) = 0$, in particular as the shock moves back downstream of the throat at $t/T_0 \approx 50$. Thus, any error in the state induced for $t/T_0 < 30$, will be propagated into $\dot{x}$, back into x and so forth, until the system returns to steady state. For the system under consideration, the output y, the average Mach number at the throat, is also very sensitive to errors in the state. Thus, it it essential to accurately capture the passage of the shock in critical regions of interest, such as at $t/T_0 \approx 25$, in order to avoid an error propagating through the system. It can be inferred from Figure 4-3 that the state deviation criterion does not accurately capture the relevant behavior of the shock. Even as the number of models is increased, an error in state is observed at $t/T_0 = 30$. This error is large enough to cause poor prediction of the output for $t/T_0 > 50$. This prediction does not improve as δ_{min}

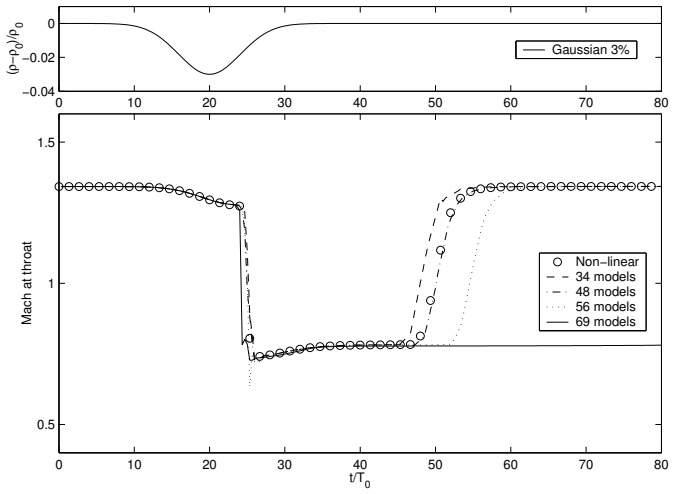

Figure 4-3: Top: Gaussian incoming density disturbance with 3% amplitude. Bottom: Nonlinear response plotted against various TPWL model combinations for a Gaussian incoming disturbance of 3% amplitude. Training trajectory obtained from the same simulation.

is decreased, since the state deviation criterion does not properly resolve the critical region.

Figures 4-4 and 4-5 show TPWL results for all of the Gaussian amplitudes, using values of δ_{min} equal to 0.01 and 0.005, respectively. For each case, the training trajectory corresponds to the desired incoming disturbance. In the first figure, a linear combination of 5, 16 and 28 models have been used for the cases corresponding respectively to Gaussian distributions of amplitude of 1%, 2% and 3%. For the second graph, a linear combination of 15, 36 and 56 models have been used.

Comparing these figures, one gains insight to the value of δ_{min} required in order to obtain a good match between the PWL model and the nonlinear CFD. As Figure 4-2 shows, a minimum number of models is needed to capture a sufficiently high degree of nonlinearity. However, as Figures 4-3, 4-4 and 4-5 demonstrate, taking too many models may cause undesirable results. In particular, oscillations may be observed or behavior may be inaccurately captured in regions where an error in state is propagated. This is observed in the middle plot of Figure 4-4 at a time $t/T_0 \approx 35$ as well as on lower plot of Figure 4-5 at a time $t/T_0 \approx 50$. Both of those errors were the result of an inappropriate resolution of the nonlinearity induced by the shock motion around $t/T_0 \approx 25$. While the output at that time was accurately captured, a small error was created in the state and was propagated through the system, which explains the later fluctuations in the results.

In summary, the state deviation criterion can provide basic guidance on the choice of the linearization points. However, as it has been demonstrated, the method is unreliable and an "optimal" number of models must be found for each case. Thus,

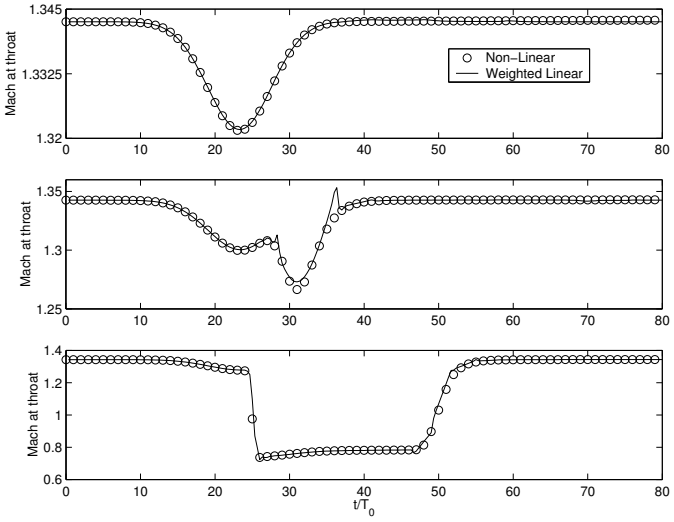

Figure 4-4: Nonlinear response plotted against TPWL models for a Gaussian incoming disturbance of various amplitudes. From top, amplitude of 1%, 2% and 3%. The training trajectory for each case was the same as the simulation. Linearized models were selected using $\delta_{min} = 0.01$. Hence, 5, 16 and 28 models have been used.

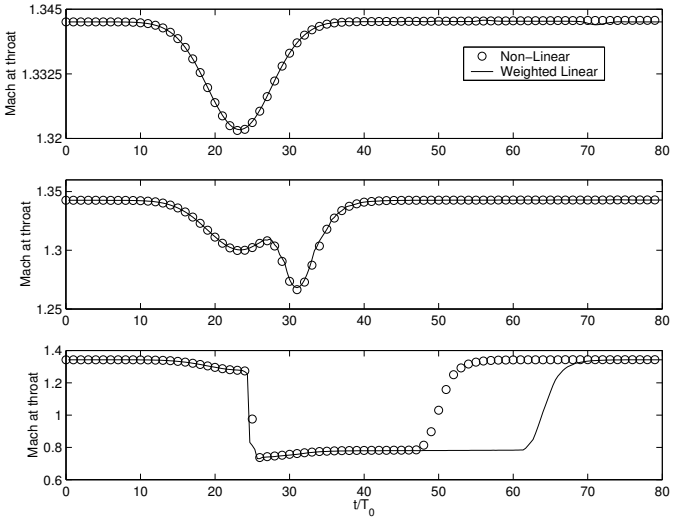

Figure 4-5: Nonlinear response plotted against TPWL models for a Gaussian incoming disturbance of various amplitudes. From top, amplitude of 1%, 2% and 3%. The training trajectory for each cases was the same as the simulation. Linearized models were selected using $\delta_{min} = 0.005$. Hence, 15, 36 and 56 models have been used.

this procedure would not be useful in the context of finding a set of models that can be used for a wide range of disturbances. A better approach is to use a criterion based on the linearization residual, as is demonstrated in the next section.

4.3 Linearization points selection using Δ_{min} (residual deviation criterion)

As demonstrated in the previous section, the quality of the TPWL models using the state deviation criterion is strongly sensitive to the set of models that are selected. The residual deviation criterion was applied to the same snapshot ensemble, resulting in a different set of linearization points. In order to compare the effectiveness of each method, the position of the chosen linearization points is examined for both algorithms. Figure 4-6 presents the location of the selected linearization points in time using each algorithm for cases 1, 2 and 3.

The usefulness of the second method can be seen rapidly; more models are concentrated in the region where the nonlinearity is high, i.e. when the shock motion is important ($t/T_0 \approx 25$ and $t/T_0 \approx 50$). Conversely, the first method concentrates models in regions were the perturbation is negligible ($t/T_0 \approx 22$ and $t/T_0 \approx 40$). Since the state deviation criterion is not concentrating enough models when the shock motion is important, an error is induced in the solution at this point ($t/T_0 \approx 25$) during the simulation. The position in time when the shock moves back downstream of the throat ($t/T_0 \approx 50$) is very sensitive to that error, as demonstrated in the last section. Therefore, an algorithm which concentrates the models where the nonlinearity peaks is more appropriate for constructing an accurate TPWL model.

Recall that the choice of linearization points for the residual deviation criterion is based on the quantity $g_0(x, u)$, which tracks the residual due to the truncation in the expansion of the nonlinearity about steady-state conditions. Figure 4-7 shows the value of $g_0(x, u)$ at four different points in time during a simulation of case 3. By definition, $g_0(x_0, u_0)$, the residual at steady-state conditions, is zero. Also at steady-state conditions, the components of the state related to the shock have indices in the range 8000-9000. During the simulation, the shock motion induces nonlinear changes in the state. The new indices where the nonlinearity of the shock motion occurs can be tracked by the value of $g_0(x_0, u_0)$. As it can be seen on Figure 4-7, nonlinear residuals clearly appear along the path of the shock. Thus, the value of $g_0(x_0, u_0)$ is a good indicator of the nonlinear effect due to the shock motion.

In addition, this indicator can provide information about the time step required for the training simulation. For instance, if for a given value of Δ_{min} it was found that new linearization points were selected at every time step, a finer time step should be used. Thus, it can be seen as a quantitative guidance in the process of generating a sufficient number of linearization points.

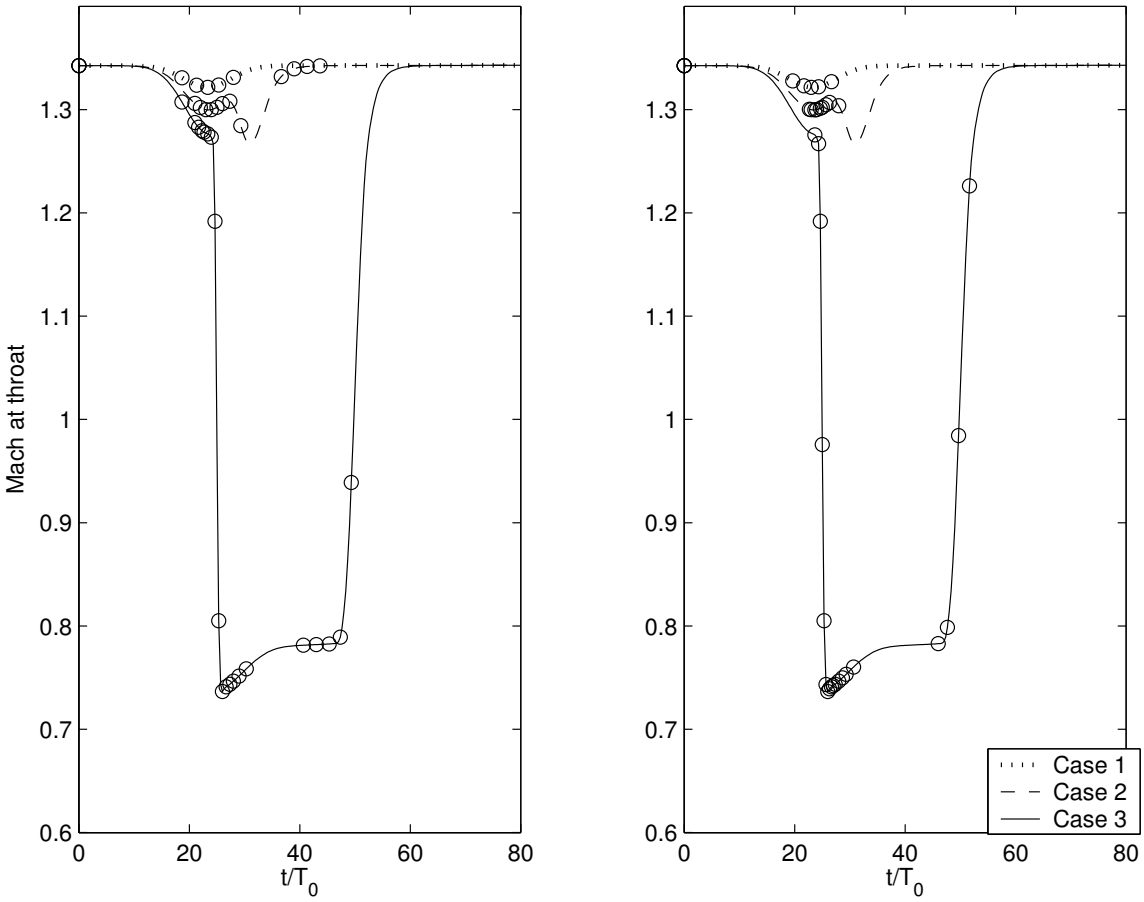

Figure 4-6: Location of the selected linearization points using each algorithm. Left: Algorithm 1, using $\delta_{min} = 0.008$, which corresponds to 38 models. Right: Algorithm 2, using $\Delta_{min} = 0.6$, which corresponds to 34 models. Model locations are denoted by circles.

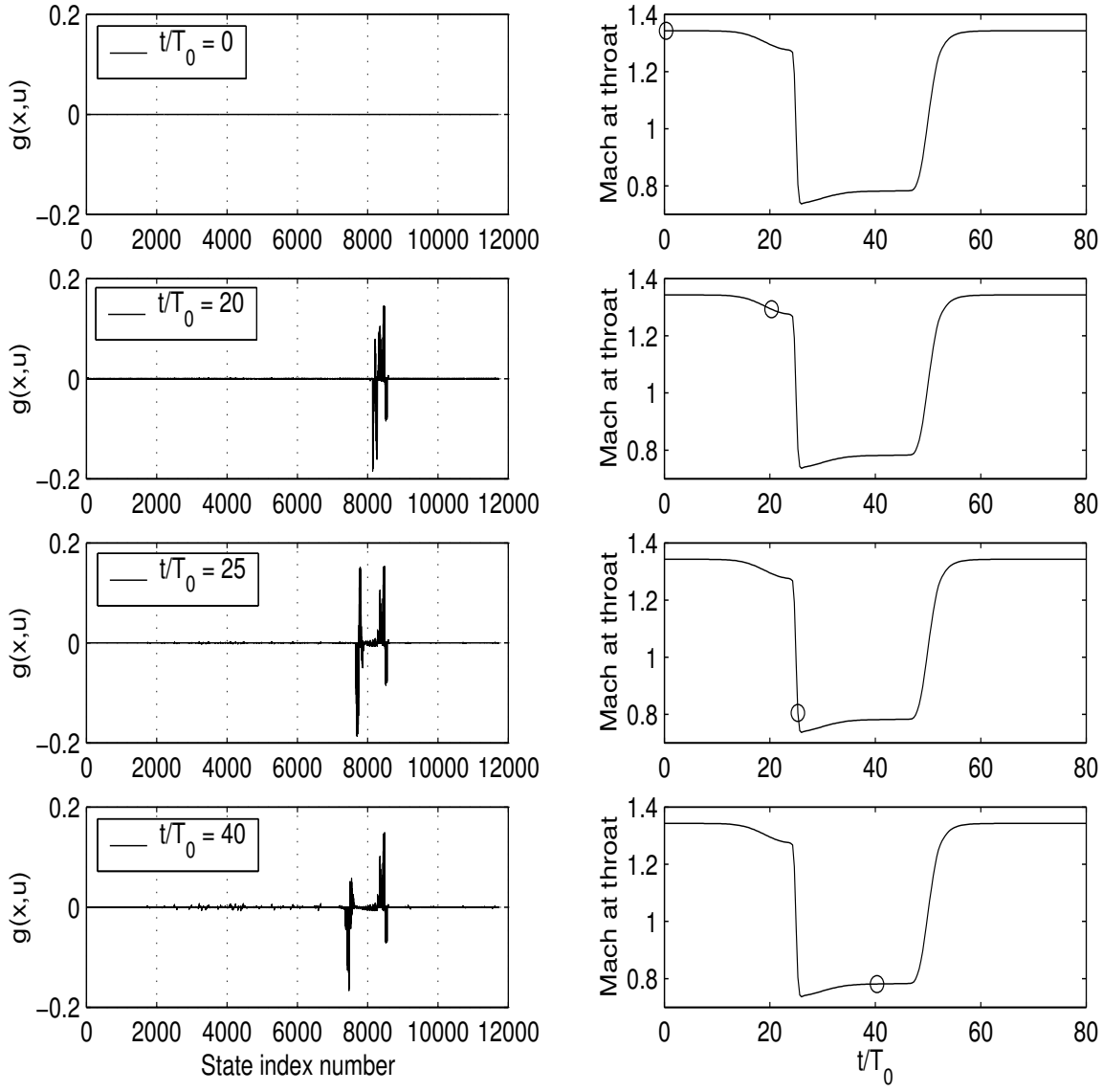

Figure 4-7: Plots on the left show the values of the components of $g_0(x, u)$ at various times during a simulation of case 3. The plots on the right show the corresponding instant in the simulation, denoted by a circle.

Δ_{min}	# of Models
∞	1
1.0	13
0.8	21
0.6	34
0.5	41
0.4	53
0.3	76
0.2	115
0.1	254

Table 4.4: Number of models given by different values of Δ_{min} applied to the complete snapshot collection.

Using this residual deviation criterion to select linearization points, simulation of the system for case 3 has been performed and good agreement with the nonlinear CFD has been obtained. In particular, unsatisfactory behavior as the number of linearization points is increased is no longer observed. The next section focuses on those results, but by using a combination of all the training trajectories instead of taking them independently.

4.4 Full-order TPWL model

In the context of finding a reduced-order model that is valid over a range of flow conditions, the different input cases would not be considered separately. Rather, the snapshots from each would be combined to derive a TPWL system that captures all training trajectories. To achieve this, all 1440 snapshots obtained from the three different training trajectories of 1%, 2% and 3% Gaussian disturbances were combined to form one large data set. Linearization points were then selected from the complete set using Algorithm 2 for various values of Δ_{min}. Table 4.4 shows the total number of models for a range of Δ_{min} values.

The results using different sets of models from Table 4.4 are shown on Figure 4-8, where the average Mach number at the throat is plotted against time. Here, the incoming density disturbance following a Gaussian of 3% amplitude was used, and the training trajectories were composed of cases 1, 2 and 3.

Recall that only one linearized model cannot capture the nonlinear behavior of the shock motion. As the value of Δ_{min} is decreased, the match improves with an increasing number of models. With 254 models, the TPWL system provides a very accurate approximation to the nonlinear CFD model for this case.

Those simulations used 12 time steps per period, i.e. a total of 960 steps for the complete simulation. It is important to note that with a coarse time step, high

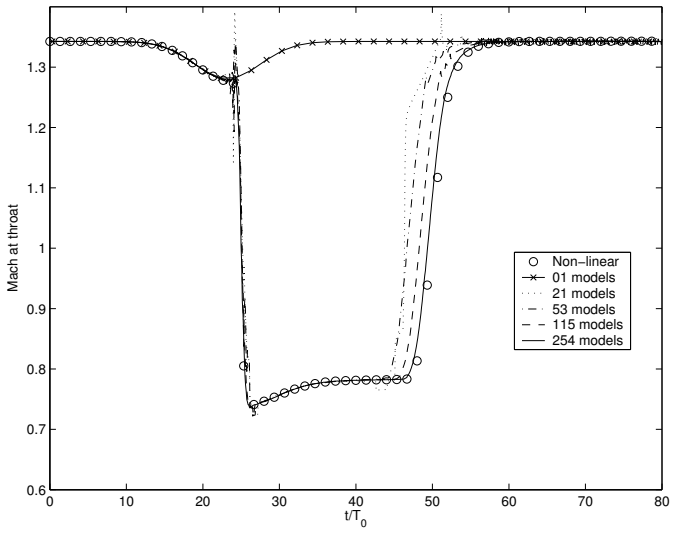

Figure 4-8: Nonlinear response plotted against various TPWL model combinations for a Gaussian incoming disturbance of 3% amplitude. Training trajectories obtained from the cases 1, 2 and 3. Model selection using the Δ_{min} criterion.

errors are induced in the shock portion ($t/T_0 \approx 25$) and the same problem previously observed with the state deviation criterion appears. The convergence for the time step procedure can be tracked on Figure 4-9.

From Figure 4-8, it can be seen that using 254 models yields the most accurate results; however, in order to balance an acceptable level of accuracy with computational efficiency, 115 models will be used for further calculations. Figure 4-10 presents the simulation of those 115 models applied to the sinusoidal inputs of cases 4, 5 and 6. Note that these cases were not considered explicitly as part of the training trajectory set; however, they would be expected to fall within the range of validity of the existing ensemble. Very good agreement between the full nonlinear CFD and the set of combined models can be seen for cases 4 and 5. For the larger amplitude of case 6, some discrepancy with the CFD is observed. This case is particularly challenging, since the inputs are oscillating at the limit of the range of validity of the models. Also, the nonlinear CFD model itself was close to instability in this case, since the amplitude of the shock oscillation was gradually increasing in subsequent periods. Nevertheless the TPWL approach shows a dramatic improvement over using a single linearized system and can capture important shock motion with reasonable accuracy.

Using this set of 115 models obtained with the residual deviation criterion, the next section will focus on the creation of appropriate PWL reduced-order models. Also, the weighting procedure using a combination of multiple models, which was too costly to study here, will be examined.

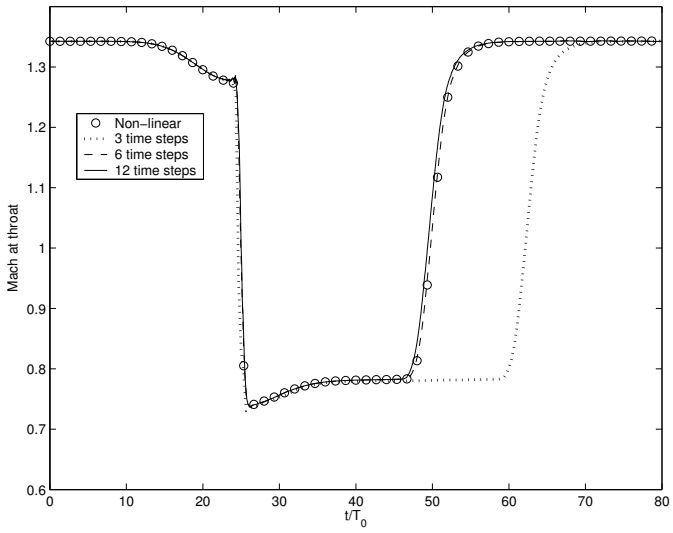

Figure 4-9: Increase in accuracy with time step refinement. Nonlinear response plotted against a TPWL model combination of 254 models ($\Delta_{min} = 0.1$), for a Gaussian incoming disturbance of 3% amplitude.

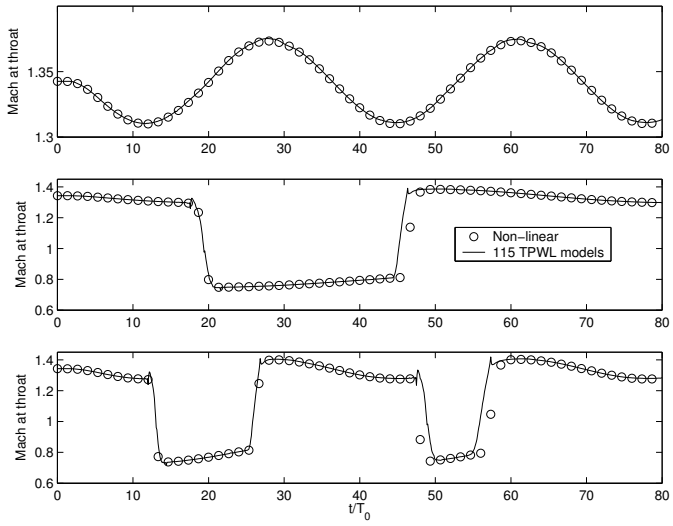

Figure 4-10: Full-order TPWL simulation for sinusoidal inputs. 115 models have been used, corresponding to a $\Delta_{min} = 0.2$. From top: Case 4, case 5, case 6.

Chapter 5

Reduced-order TPWL results

In the last chapter, a TPWL model was obtained by considering a range of training trajectories and a linearization point selection criterion based on the linearization residual. A model was obtained that was able to replicate the behavior of the nonlinear CFD for a range of inputs characterized by the training trajectories. The present chapter considers the same cluster of models on which model order reduction is performed.

The first section presents the different parameters to consider in order to obtain an accurate reduced-order TPWL model. In the second section, the set of models from the previous chapter is projected onto a reduced-order state space using only a small number of states. As for the full-order system, this cluster is tested for various incoming disturbances. The third section introduces the weighting procedure using a combination of multiple models. Results are shown and compared between each of the two weighting methods. Finally, the last section discusses the performance of the TPWL reduced-order models and compares with that of the nonlinear and full-order TPWL methods.

5.1 Reduction analysis

This section first describes the creation of the snapshot ensembles for both time and frequency-domain POD bases. Then, a range of results is presented which demonstrates the accuracy improvement gained by increasing the number of states in the reduced system. This is achieved through the comparison of the transfer functions of both full-order and reduced-order systems.

The advantage of generating the snapshots in the time domain in conjunction with the TPWL framework is that both can be executed efficiently at the same time. It is important to note that, as for the training trajectory used for linearization points selection, the snapshot selection should span all operating conditions of interest. Thus, the snapshots can be obtained efficiently from the same training trajectory used to

Energy (%)	# Time basis vectors	# Frequency basis vectors
99.0000	9	11
99.9000	18	30
99.9900	33	55
99.9990	61	80
99.9999	110	99

Table 5.1: Percentage of total energy captured by various number of basis vectors included in the basis.

create the PWL model.

Using this approach, the time domain snapshot ensemble has been generated from the same three training trajectories used in the previous chapter, i.e. the three Gaussian input pulses given in Table 4.1. For each trajectory, 480 snapshots were collected corresponding to the solution of the nonlinear CFD at every time step (6 time steps per period), yielding a total of 1440 snapshots for the time domain basis. The solution at every time step was saved; however, one could have selected snapshots differently, in order to focus more on a specific behavior of the flow. POD basis vectors were then calculated. The relative energies e_j associated to the first 120 POD eigenvalues are plotted in Figure 5-1. Following Equation (2.12), the number of basis vectors required in order to capture a given level of the total energy can be seen in Table 5.1.

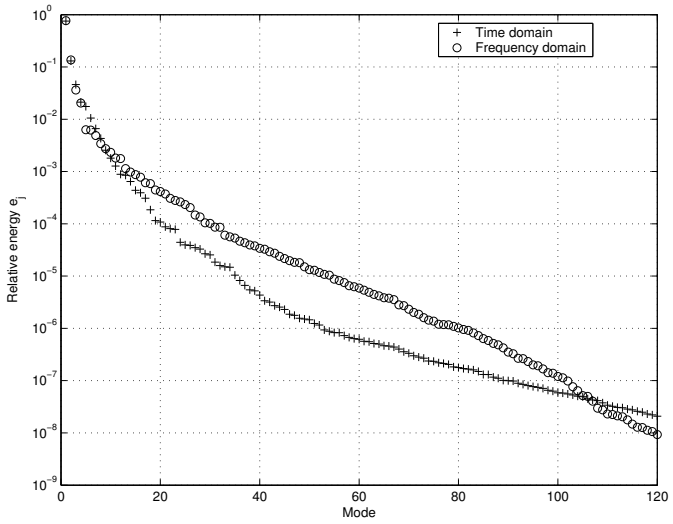

Figure 5-1: First 120 POD eigenvalues for both a cluster of 1440 snapshots in time domain, and 2415 snapshots in frequency domain, given as relative energy of the total of each ensemble.

While the time domain basis can be obtained efficiently from the same training trajectory of the PWL model, it could arise that relevant information at particular

frequencies is not captured accurately. Collecting snapshots in the frequency domain can resolve this issue. While obtaining the basis in the time domain requires snapshots from the nonlinear CFD simulation, the frequency domain methodology requires an extra step. It consists of extracting the frequency snapshots of the various linearized systems inside the PWL model cluster. In general, this option will require a much larger cost to execute, but could provide advantages over the time domain basis in some applications. It could also be used in cases where simulation of the nonlinear CFD model is prohibitive. For example, for turbomachinery applications, solution in the frequency domain allows spatial periodicity of the system to be exploited and thus provides a significant computational advantage.

The construction of the frequency domain basis first requires the sampling of an appropriate range of frequencies in each of the linearized systems of the PWL model. One way of choosing an appropriate frequency range is by looking at the sharpness of the disturbance, thus the significant frequencies present. The input takes the form:

$$\rho(t) = e^{-\alpha(t-t_0)^2} \tag{5.1}$$

In order to obtain the frequency content, a Fourier transform is performed on Equation (5.1):

$$\rho(\omega) = \frac{1}{2\sqrt{\pi\alpha}} e^{i\omega t_0} e^{\frac{-\omega^2}{4\alpha}} \tag{5.2}$$

Figure 5-2 shows that only frequencies ranging from 0.0 to 0.13 are relevant for the assumed range of incoming disturbance. Hence, we will want to match this entire range for all of the transfer functions of the PWL model. Transfer functions of both

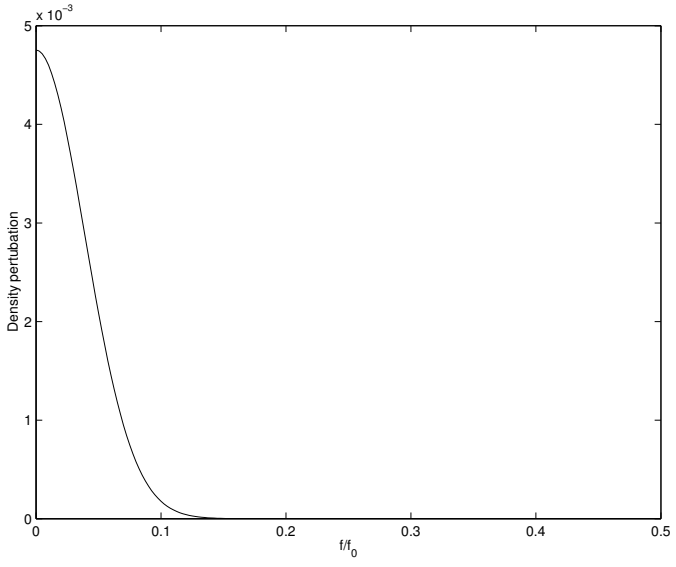

Figure 5-2: Frequency content of the input.

inputs of each of the 115 models previously selected were used for the snapshots ensemble. Ten frequencies for the first input between $f/f_0 = [0.0 - 0.20]$ were used, in addition to the zero frequency of the second input, which is constant in time and equal to one. Therefore, the resulting ensemble contained all the relevant frequencies ranging from 0.0 and 0.13. Using both the real and imaginary part of each solution, a total of 2415 frequency-domain snapshots were generated. Figure 5-1 also shows the relative energies e_j associated to the first 120 POD eigenvalues for this set of snapshots. Again, the number of basis vectors required in order to capture a given level of the total energy is presented in Table 5.1.

Using the fact that both time domain and frequency domain eigenvalues are decaying exponentially, a selection of the most dominant modes, corresponding to the largest eigenvalues, will be sufficient for the projection basis. However, a substantial amount of modes will still be required because of the high complexity of the flow problem. This can be observed on Figure 5-3, where the contours of perturbation Mach number of the six first modes of the time domain POD basis are plotted. It can be seen that five out of the six dominant basis modes are strongly driven by the normal shock motion. The remaining mode, the fourth one, captures the oblique shock. Although not plotted, the next twelve dominant modes also capture the normal shock motion. The main result of this phenomenon is that most of the basis energy is dominated by large perturbations in a few flow variables due to nonlinearities, with a resulting decrease in accuracy for the perturbations of smaller amplitude. Since all the different levels of this multi-scale problem can have similar importance, the quality of the resulting reduced-order PWL model can be badly impacted if not enough states are included inside the basis. In such a case, the resolution of the small perturbations would not be captured with sufficient accuracy. This could lead to poor results, such as those observed previously on Figures 4-5 and 4-9, where small errors in the state prediction translated into large errors in subsequent outputs. The main drawback of this phenomenon is that a POD basis is often limited in its number of states by a stability issue, i.e. a large POD basis often yields unstable reduced-order models.

Using both the time domain and the frequency domain basis, the PWL model is projected into reduced-order PWL models of various sizes. Figures 5-4 and 5-5 present the transfer function from incoming density disturbance to average throat Mach number for model number 57 out of the 115 models. They present the comparison between the full-order model and various number of states included in the reduced-order model for both the time domain and the frequency domain basis.

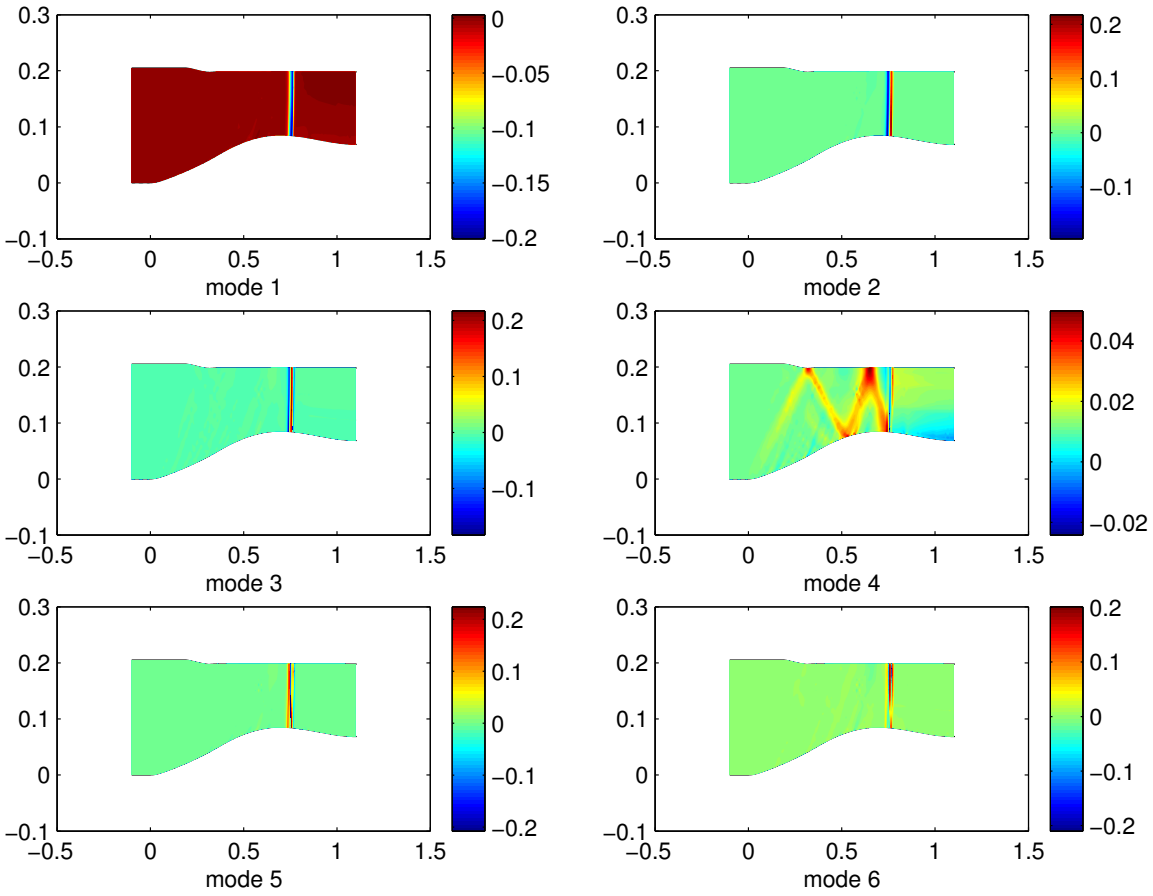

Figure 5-3: Contours of perturbation Mach number of the first 6 dominant modes of the time domain POD basis.

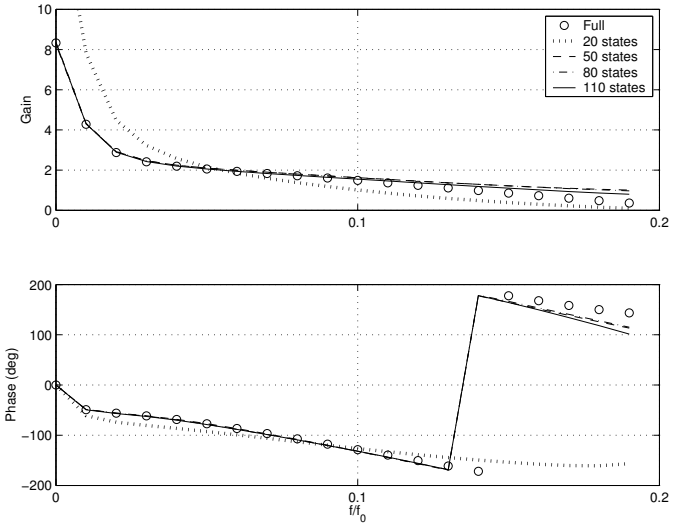

Figure 5-4: Transfer function from incoming density disturbance to average throat Mach number for model number 57 out of 115 ($\Delta_{min} = 0.2$). Comparison between full-order linear model versus various sizes of reduced-order models using a time domain basis.

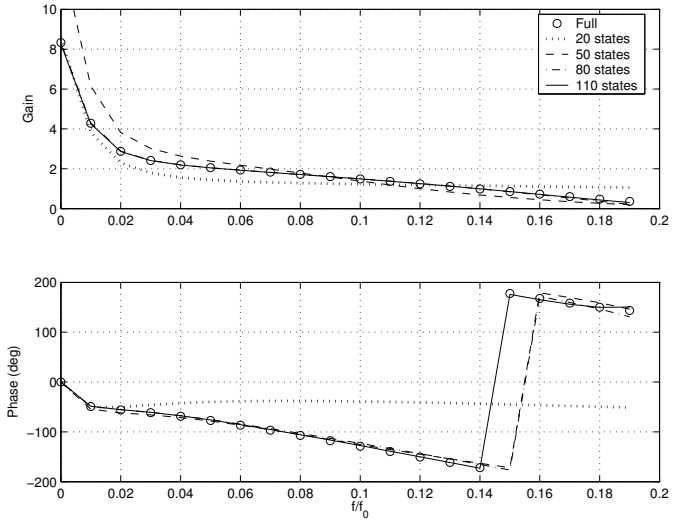

Figure 5-5: Transfer function from incoming density disturbance to average throat Mach number for model number 57 out of 115 ($\Delta_{min} = 0.2$). Comparison between full-order linear model versus various sizes of reduced-order models using a frequency domain basis.

Both bases show very good agreement for the relevant range of frequencies (f/f_0 = [0.0 - 0.13]) when they are composed of at least 110 states. The advantage of using the frequency-domain snapshots is that it allows matching of the transfer function over a range of frequencies to be controlled more easily. Note that for the frequency domain basis, the entire range of frequency plotted, f/f_0 = [0.0 - 0.20], was included in the basis. However, the deviations at higher frequencies in Figure 5-4 demonstrate the danger of applying the reduced models outside their range of validity for the time domain basis and emphasizes the importance of selecting the training trajectories appropriately. The relatively large number of states required in the reduced-order model reflects the complexity of the flow dynamics in this application.

The formulation (3.4) contains a second input term, B_2, for which the dynamics must also be accurately captured in the reduced-order model. Due to the form of this term (i.e. there is no explicit input u_2), only the match at zero frequency is relevant. At zero frequency, the transfer function of the second input is given by:

$$G_{B_2} = CA^{-1}B_2 \tag{5.3}$$

While the reduced-order transfer function will be denoted by:

$$\hat{G}_{B_2} = \hat{C}\hat{A}^{-1}\hat{B}_2 \tag{5.4}$$

The error associated to the second input can then be expressed as:

$$\epsilon = \frac{\hat{G}_{B_2} - G_{B_2}}{G_{B_2}} \tag{5.5}$$

Figure 5-6 shows the value of ϵ as a function of the number of states in the reduced-order model. Note that this is the same basis used to create the results shown in Figures 5-4 and 5-5. As before, using 110 reduced states yields an acceptable level of accuracy.

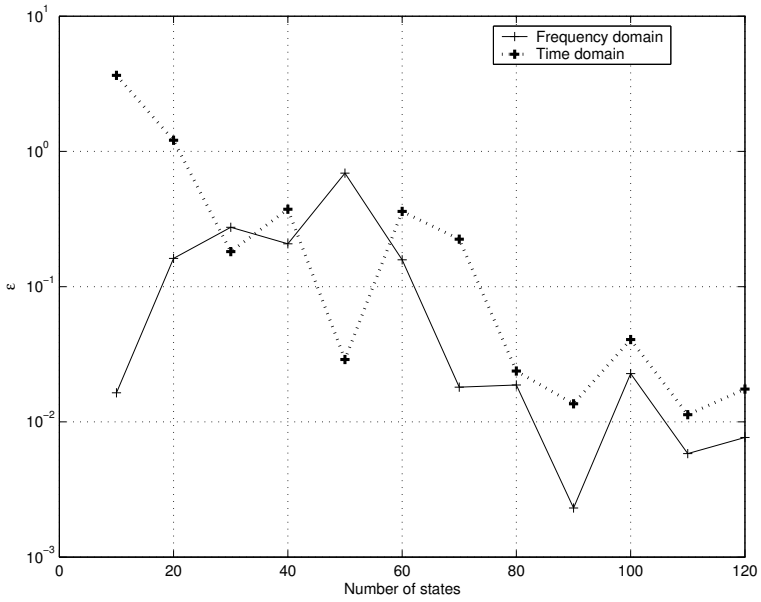

Figure 5-6: Error on the transfer function at zero frequency for the second input of model number 57 out of 115 ($\Delta_{min} = 0.2$) for various number of states.

5.2 Simulation results

Using a Δ_{min} of 0.2 and the corresponding 115 linearization points, a set of reduced-order models is created by projection of each linearized model onto the reduced space spanned by the first 110 POD basis vectors of both time-domain and frequency-domain bases. Since a sufficient number of POD basis vectors is used to define the reduced space, accurate reduced-order models can be obtained.

Using this set of 115 reduced models with the system defined by (3.4), the TPWL scheme is tested. Results are presented for the weighting procedure using only the closest linearized model at every time step (option [b] in step 3 of Algorithm 3). Figures 5-7 and 5-8 present the simulation results of this final system for various incoming density disturbances.

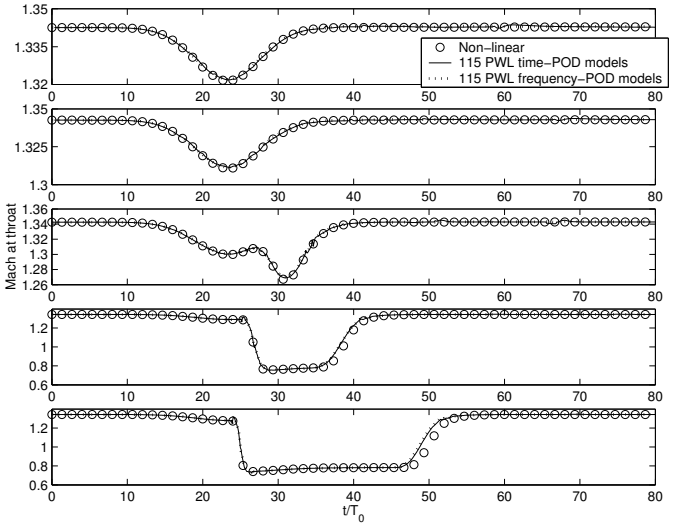

Figure 5-7: Simulation using 115 reduced-order PWL models composed of 110 states each. From top: Gaussian distribution of 1%, 1.5%, 2%, 2.5%, and 3% amplitude.

Figure 5-7 plots the average Mach number at the throat of two reduced-order PWL models in response to Gaussian pulses of 1%, 1.5%, 2%, 2.5%, and 3% amplitude. A time domain POD basis has been used to create the first reduced-order PWL model, while a frequency domain POD basis has been used to create the second one. Both systems were composed of 110 states. Good agreement is achieved for all disturbances. The small discrepancy noted for the response of case 3 is a replication of the one found with the full-order PWL system.

More results are shown on Figure 5-8, where the same models are tested under sinusoidal inputs (cases 4, 5 and 6), which were not explicitly included in the training sampling process. Similar behavior as for the Gaussian responses is observed; results for the smaller amplitude sinusoids are excellent, but discrepancy is again observed for the 3% amplitude in the lower plot. Again, as it was observed with the full-order

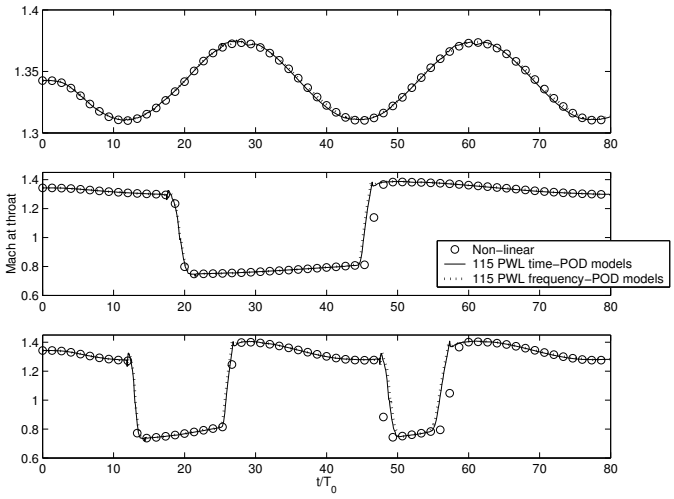

Figure 5-8: Simulation using 115 reduced-order PWL models composed of 110 states each. From top: Case 4, case 5 and case 6.

PWL system, case 6 was oscillating at the limit of the range of validity of the clusters of models and therefore is more challenging to capture accurately.

5.3 Weighting procedure

As the next section will demonstrate, using a combination of models at the same time during a simulation (option [a] in step 3 of Algorithm 3) becomes rapidly very expensive to execute. Therefore, only the second approach using the closest model has been examined so far. However, it might seem advantageous to combine multiple models at the same time to achieve a better approximation of the nonlinearity. Note that one particular aspect of the weighted combination is that it only takes the model(s) that are in the immediate neighborhood of the current solution. Therefore, adding more models at a given time may not be expected to have a large effect. Figure 5-9 presents the relative weights for three models during a simulation. Here, six time steps per period were used. It can be seen that Algorithm 3 strongly concentrates the weights on an individual model, and switches from one to another rapidly.

Both weighting procedures have been compared for various incoming disturbances. Figure 5-10 is descriptive of what is happens with this approach in most of the situations considered. Plotted are the simulation results of the 115 models reduced to 110 states for both weighting procedures. It can be seen that inside the sensitive area where the shock is oscillating nearby the throat ($t/T_0 \approx [25 - 35]$), the method using a combination of models fails to match the nonlinear CFD. This inaccuracy was also observed in other similar simulations. In addition, as discussed earlier, this weighting procedure is more computationally expensive than using a single model.

Therefore, no further investigations of this procedure have been performed, since the combination of inaccuracy and inefficiency make this option unattractive.

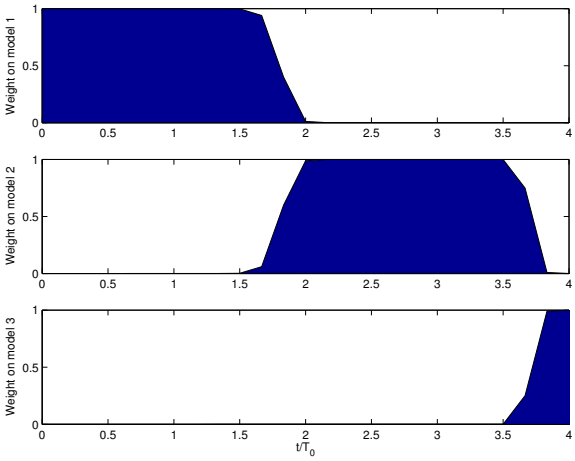

Figure 5-9: Weights distribution for 3 of the 115 linearized models during a portion of a simulation of case 3.

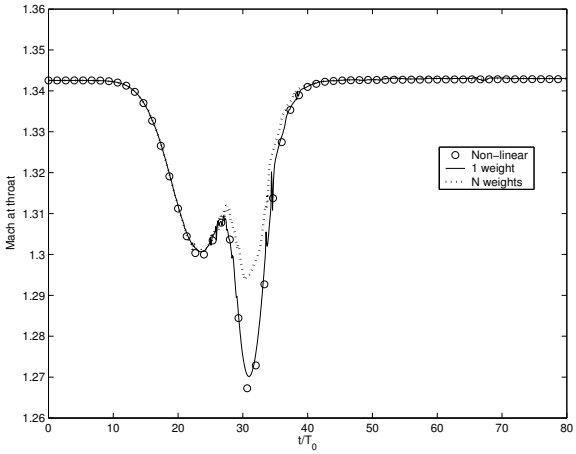

Figure 5-10: POD method for a Gaussian incoming disturbance of 2% amplitude using only one model at the time versus using a weighted sum of models. 115 models composed of 110 states each have been used.

5.4 Computational performance analysis

The performance of the proposed TPWL ROM technique, based on selecting a collection of linearization points from a state-space trajectory of the original nonlinear system, is considered. First, the efficiency of the basis creation for both time and frequency domain is discussed. Then, the performance of the optimal reduced-order PWL model is compared to that of the full-order and nonlinear systems. All algorithms were implemented in Matlab, with the exception of the nonlinear solver, which was running under Fortran. The tests were performed on a Linux workstation with Pentium IV processor and 512MG Ram.

Before performing a simulation, the projection basis must be created, and the full-order PWL model projected onto the reduced-order space. This step involves computing the correlation matrix (2.10) from the snapshot ensemble and extracting its eigenvalues and eigenvectors. However, an extra step was required for the frequency-domain. While the time-domain method made efficient use of the CFD training trajectories, the frequency-domain method required solution at each of eleven frequency points for each of the 115 models, which was computationally expensive. From the previous section, it has been seen that both methods yield similar matches for the transfer function. However, the advantage of using the frequency-domain method is that it is easier to ensure that the relevant range of frequencies in the transfer function is sampled. Conversely, the time-domain basis can be efficiently created during the training trajectory.

Figure 5-11 presents the various operations required by a simulation of the PWL model, broken into off-line and online computation time. Off-line time consists of the time to load and factorize, if possible, the PWL model, while the online time includes all the computations that would be necessary for simulation in real time. For reduced-order models, the off-line time also includes the time for projection. Path (A) and (B) denote respectively option [a] and option [b] in step 3 of Algorithm 3, the weighting procedure. All simulations have been performed using a second-order accurate backward Euler time discretization. Thus, one of the computational advantages of simulation with linearized state-space models is that factorization of the system matrix can be performed only one time during the off-line initialization. Online simulation is then very rapid, requiring only forward and backward substitutions. However, one particular aspect of the weighting procedure using a combination of models (option [a] in step 3 of Algorithm 3) is that it does not allow a priori factorization of the matrices, since the relative weighting of the combined Jacobian matrices varies at each time step. Thus, a matrix factorization is required at every time step during the simulation, which is computationally costly.

Table 5.2 presents the total off-line time required for each system in order to launch a simulation. Presented for each method are the number of piecewise-linear models, the number of states in each system, and the off-line time required. Those times are

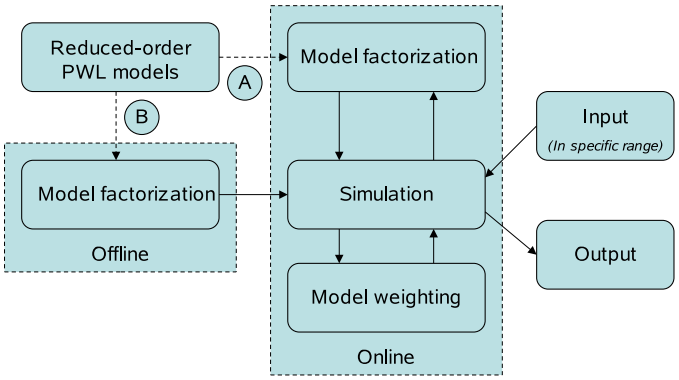

Figure 5-11: Off-line and online computational operations required for a simulation.

Method	#Models	#States	Pre-factorization (Y/N)	Off-line model generation
Full-order TPWL	254	11730	N	1371s
Full-order TPWL	115	11730	N	588s
Full-order TPWL	76	11730	N	388s
Full-order TPWL	76	11730	Y	539s
Reduced-order TPWL	115	110	Y	1656s
Reduced-order TPWL	76	110	Y	1073s

Table 5.2: Off-line computational cost to load, project and factorize each PWL method.

directly proportional to the size of each system (number of models and number of states). The pre-factorization column in Table 5.2 shows whether the system matrices have been factorized off-line (Y) or online (N). A relatively high level of memory is required in order to load at once a multitude of full-order systems. Thus, a lack of memory did not allow off-line factorization of the 115 and 254 set of full-order models.

Table 5.3 presents a comparison of the online computational time needed for the nonlinear CFD, the full-order PWL and the reduced-order PWL methods. The data in Table 5.3 correspond to a simulation of case 3. The difference between the different online times is attributable to two phenomena. The first one, as described before, is that off-line factorization allows a faster simulation of the system. Another issue that arises is that of storage requirements. For the large systems considered here, it is not practical to store a large number of Jacobian matrices in memory. The matrices must therefore be loaded as they are required. These two factors combine to make weighting option [a] extremely unattractive. This can be seen in the data for a simulation of 76 full-order PWL models using weighting option [a], which has a very high computational time. This time arises from the need to load 76 Jacobian matrices and compute one matrix factorization at every time step. Moreover, this weighting option did not yield accurate results as was described in the previous section.

Method	#Models	#States	Weights option	Pre-factorization (Y/N)	Online simulation
Nonlinear	-	11730	-	-	7200s
Full-order TPWL	254	11730	b	N	1316s
Full-order TPWL	115	11730	b	N	1233s
Full-order TPWL	76	11730	a	N	10065s
Full-order TPWL	76	11730	b	Y	803s
Reduced-order TPWL	115	110	a	N	301s
Reduced-order TPWL	115	110	b	Y	5s
Reduced-order TPWL	76	110	b	Y	5s

Table 5.3: Computational performance for each method applied to simulations of case 3.

Most relevant is the performance of the reduced-order PWL models. Table 5.3 shows the results for three different sets of models. All three show a significant decrease in computational time compared with the nonlinear CFD simulation. Once again, use of weighting option [a] is considerably more expensive (due now to factorization at every time step) and does not yield any significant improvement in accuracy. The recommended approach, using weighting option [b], results in models that are extremely computationally efficient. Using these models, a simulation that took several hours using CFD can be performed in just five seconds with a high level of accuracy.

Finally, it should be noted that direct projection of the nonlinear system onto the reduced-space basis would yield a "reduced" model of the form (2.14). Using an implicit time discretization, this implementation would yield almost no computational savings over the nonlinear CFD results, since the full-order nonlinear term $f(\cdot)$ remains in the model.

5.5 Summary

In summary, the method of snapshots was used in both time domain and frequency domain in order to create POD bases. Because of the high complexity and multi-scale nature of the flow dynamics, more states than usual were included in the POD bases. The input-output behavior of the reduced-order PWL models was then tested via the comparison of their transfer functions with the full-order systems. Using various disturbances, the resulting reduced-order PWL models have been tested and compared to the nonlinear CFD. The reduced-order PWL models shown very good agreement with the nonlinear CFD, while being computationally very efficient.

Chapter 6

Conclusion

This work combined two methodologies in order to model nonlinear fluid dynamic behavior in a cost-effective way. Through a POD projection of a TPWL system, efficient reduced-order PWL models have been obtained.

6.1 TPWL methodology

The assembly of a reduced-order PWL model relies on several choices and steps. The following final process is recommended and can be tracked on Figure 3-1:

1. Perform a series of nonlinear CFD calculations which should span all operating conditions of interest.
2. Using the snapshots created, select the linearization points using the residual deviation criterion described in Chapter 3.
3. Using the snapshots created, create an orthogonal time-domain POD basis for projection.
4. Project the full-order PWL model onto a reduced-order PWL model using the POD basis.
5. Simulate the reduced-order PWL model.
6. Weight the models according to the weighting procedure using only the single closest model at the time, option [b] in step 3 of Algorithm 3.

6.1.1 Algorithmic choices

The algorithm just described is based on recommendations concerning the linearization point selection methodology, the snapshots creation following either a time-domain or a frequency-domain analysis, and the choice of the weighting procedure.

- The linearization points selection can be performed in various ways. Two methodologies have been studied, and the residual deviation criterion has been demonstrated to be an effective tool in capturing the domain of validity of a linearized model, since it concentrates models in regions where the nonlinear changes are important. Conversely, the state deviation criterion demonstrated poor capabilities in doing so.

- The POD approach has been used for reduction of the PWL system. Snapshots have been obtained from both time-domain and frequency domain analysis. However, only the time-domain approach can be combined efficiently with the TPWL framework, since both require training trajectories which should span all operating conditions of interest. Thus, the snapshots only need to be created once and used for both linearization point selection and POD basis creation. The resulting efficiency of the methodology made this option very attractive when combined with the TPWL scheme. Conversely, the frequency-domain analysis can be useful for applications where particular frequencies are difficult to capture inside a time-domain analysis or when the nonlinear code is prohibitively expensive to simulate, as for some turbomachinery applications.

- Simulations require weighting of the PWL model. Due to its failure to capture with accuracy the nonlinear behavior and its high computational cost, the weighting procedure using a combination of models at every time step was rejected. Instead, only one model at a time was used during simulation. The choice of that model was based on a state deviation criterion.

6.1.2 Algorithm performance

The final model reduction methodology created a reduced-order PWL model with the following characteristics:

- A large number of states is required for the projection basis, since the basis energy must include multi-scale behavior, i.e. accurate resolution for both small linear perturbations and large nonlinear perturbations must be captured. Therefore, a reduced-order PWL model with a relatively large number of states has been obtained (~ 100) in order to maintain a reasonable level of accuracy. However, the size of the reduced-order PWL model will depend on the performance requirements and dynamical behavior of a given application, thus is considered to be problem-dependent.

- Accuracy of the projection has been tested through the comparison of the transfer functions of both full-order and reduced-order PWL models. Good agreement for both inputs have been obtained with 110 states.

- Disturbances to the operating conditions have been used to assess the quality of the resulting system when compared to the nonlinear CFD. An acceptable level of accuracy has been obtained.

- The reduced-order PWL system shown a strong gain in efficiency over the nonlinear CFD, demonstrating an economy of three orders of magnitudes on the computational time.

In summary, the TPWL methodology has been demonstrated as a viable approach for obtaining accurate, efficient, reduced-order models for nonlinear CFD applications. For the application considered, very good agreement with nonlinear CFD results was obtained, even in cases that included significant shock motion. The method yields reduced-order models that accurately capture important flow dynamics and are computationally very efficient.

6.2 Recommendations for Future Work

A number of open questions remain as the subject of future work. Although systematic algorithms have been suggested in this paper, the selection of the training trajectories, the snapshot collection, and the linearization points remains an ad-hoc process. The lack of quality guarantees on the resulting reduced-order models is also unsatisfactory. However, it is important to note that these are unaddressed issues associated with the use of POD which also appear in model reduction of linear systems and are commonly ignored. More rigorous reduction techniques with associated error bounds exist, such as reduction based on Fourier modes of the discrete time system [55]. Here are some extensions that should be considered for this work:

- Develop a TPWL framework using a more rigorous method to obtain the reduced-order models, such as FMR. Using such a technique will create reduced-order models in different spaces (no common basis). Therefore, a basis-to-basis framework will be required. The advantage of POD is the fact that only one basis is used for the projection of all models. Note that TPWL has been combined with TBR for reduction of stable models in [51, 52].

- Investigate the POD basis behavior when a wider range of information is included in the snapshot ensemble. As has been demonstrated, the system nonlinearity induces a requirement for a large number of states in the reduced-order PWL model. Therefore, it is important to study how well the model behaves when the information required increases.

- Identify a weighting procedure able to capture efficiently both state and nonlinearity residual deviation from a given linearization point. Both measures are relevant for the choice of linearization point because they both participate in the definition of the range of validity of a linear model. Therefore, the current weighting procedure could be enhanced by making efficient use of both criteria.

- Develop a controller design using the reduced-order PWL framework.

Appendix A

Implementation details

It should be noted that in order to simulate the set of equations given by Equations (3.3) and (3.4), one needs to find a common representation of the state for each model. Let recall Equation (2.19):

$$x(t) = x_i + x'_i(t)$$

For two linearization points i and 0, with x_0 being the steady state condition, Equation (2.19) can be expressed as:

$$x'_i(t) = x'_0(t) - [x_i - x_0] \tag{A.1}$$

Combining this result with Equations (3.3) and (3.4), we obtain a set of rotated equations with a common state perturbation about steady state as follows:

$$\begin{aligned} \sum_{i=0}^{s-1} \tilde{\omega}_i(x)\Big\{ \dot{x}'_i(t) &= A_i x'_i(t) + B_{1i}u(t) + B_{2i}\Big\} \\ \sum_{i=0}^{s-1} \tilde{\omega}_i(x)\Big\{ y_i(t) &= C_i x'_i(t) + C_{0i}\Big\} \end{aligned} \tag{A.2}$$

and

$$\begin{aligned} \sum_{i=0}^{s-1} \tilde{\omega}_i(\hat{x})\Big\{ \frac{d}{dt}\hat{x}'_i(t) &= \hat{A}_i \hat{x}'_i(t) + \hat{B}_{1i}u(t) + \hat{B}_{2i}\Big\} \\ \sum_{i=0}^{s-1} \tilde{\omega}_i(\hat{x})\Big\{ \hat{y}_i(t) &= \hat{C}_i \hat{x}'_{0i}(t) + C_{0i}\Big\} \end{aligned} \tag{A.3}$$

where the rotated matrices will now be composed of:

$$B_{2i} = B_{2i} - A_i(x_i - x_0)$$

$$\begin{aligned} \hat{B}_{2i} &= \hat{B}_{2i} - V^T[A_i(x_i - x_0)] \\ C_{0i} &= C_{0i} - C_i(x_i - x_0) \end{aligned} \tag{A.4}$$

Hence, a given state perturbation can easily be encoded. Also, since a time discretization, such as backward Euler, can be applied, the resulting system matrix can be inverted just once.

Bibliography

[1] Anderson, B.D.O. and Moore, J.B. *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

[2] Antoulas, A.C., Sorensen, D.C., and Gugercin, S. "A survey of model reduction methods for large-scale systems." *Contemporary Mathematics*, 280:193–219, 2001.

[3] Beard, R.W., Saridis, G.N., and Wen, J.T. "Galerkin approximation of the generalized Hamilton-Jacobi-Bellman Equation." *Automatica*, 33:2159–2177, 1997.

[4] Beard, R.W., Saridis, G.N., and Wen, J.T. "Approximate solutions to the time-invariant Hamilton-Jacobi-Bellman Equation." *Journal of optimization Theory and Applications*, 96:589–626, 1998.

[5] Beeler, S.C., Tran, H.T., and Banks, H.T. "Feedback control methodologies for nonlinear systems." *Optimization theory and applications*, 107(1):1–33, October 2000.

[6] Beran, P. and Silva, W. "Reduced-Order Modeling: New Approaches for Computational Physics." AIAA Paper 2001-0853, presented at 39th Aerospace Sciences Meeting & Exhibit, Reno, NV, January 2001.

[7] Berkooz, G., Holmes, P., and Lumley, J.L. "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows." *Annual Review of Fluid Mechanics*, 25:539–575, 1993.

[8] Cebuhar, W.A. and Costanza, V. "Approximation procedures for the optimal control of bilinear and nonlinear systems." *Optimization Theory and Applications*, 43:615–627, 1984.

[9] Chen, Y. "Model Order Reduction for Nonlinear Systems." Master's thesis, Department of Electrical Engineering & Computer Science, MIT, September 1999.

[10] Cloutier, J.R., D'Souza, C.N., and Mracek, C.P. "Nonlinear regulation and nonlinear H_∞-control via the state-dependent Riccati equation technique." *Proceedings of the 1st international conference on nonlinear problems in aviation and aerospace*, Daytona Beach, Florida, 1996.

[11] Dowell, E.H. and Hall, K.C. "Modeling of fluid-structure interaction." *Annual Review of Fluid Mechanics*, 33:445–90, 2001.

[12] Drela, Mark. *Two-Dimensional Transonic Aerodynamic Design and Analysis using the Euler Equations*. PhD thesis, Massachusetts Institute of Technology, 1987.

[13] Fahl, M. "Computation of POD basis functions for fluid flows with lancozs methods." *Mathematical and Computer Modelling*, 34:91–107, 2001.

[14] Feldmann, P. and Freund, R.W. "Efficient Linear Circuit Analysis by Padé Approximation via the Lanczos Process." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14:639–649, 1995.

[15] Ffowcs Williams, J.E. and Huang, X. "Active Stabilization of Compressor Surge." *Journal of Fluid Mechanics*, 204:245–262, 1989.

[16] Gallivan, K., Grimme, E., and Van Dooren, P. "Padé Approximation of Large-Scale Dynamic Systems with Lanczos Methods." Proceedings of the 33rd IEEE Conference on Decision and Control, December 1994.

[17] Garrard, W.L., Enns, D.F., and Snells, S.A. "Nonlinear feedback control of highly maneuvrable aircraft." *International Journal of Control*, 56:799–812, 1992.

[18] Grimme, E. "Krylov Projection Methods for Model Reduction." Master's thesis, Coordinated-Science Laboratory, University of Illinois at Urbana-Champaign, 1997.

[19] 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 Paper 99-0655, 1999.

[20] Hall, K.C. *A Linearized Euler Analysis of Unsteady Flows in Turbomachinery*. PhD thesis, Dept. of Aeronautics and Astronautics, MIT, May 1987.

[21] Hall, K.C. "Eigenanalysis of Unsteady Flows About Airfoils, Cascades and Wings." *AIAA Journal*, 32(12):2426–2432, December 1994.

[22] Hung, E., Yang, Y.-J., and Senturia, S.D. "Low-Order Models For Fast Dynamical Simulation of MEMS Microstructures." *Proceedings of the IEEE International Conference on Solid State Sensors and Actuators (Transducers '97)*, 2:1101–04, 1997.

[23] Ito, K. and Schroeter, J.D. "Reduced order feedback synthesis for viscous incompressible flows." *Mathematical and Computer Modelling*, 33:173–192, 2001.

[24] Kim, T. "Frequency-Domain Karhunen-Loeve Method and Its Application to Linear Dynamic Systems." *AIAA Journal*, 36(11):2117–2123, 1998.

[25] King, B.B. and Atwell, J.A. "Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations." *Mathematical and Computer Modelling*, 33:1–19, 2001.

[26] King, B.B., Atwell, J.A., and Borggaard, J.T. "Reduced order controllers for Burgers equation with a nonlinear observer." *Applied mathematics and computational science*, 11:1311–1330, 2001.

[27] King, B.B. and Burns, J.A. "A reduced basis approach to the design of low order compensators for nonlinear partial differential equation systems." *Journal of vibrations and control*, 4(3):297–323, 1998.

[28] Korenberg, M.J. "Identifying Nonlinear Difference Equation and Functional Expansion Representations: the Fast Orthogonal Algorithm." *Annals of Biomedical Engineering*, 16:123–42, 1998.

[29] Korenberg, M.J. and Hunter, I.W. "The Identification of Nonlinear Biological Systems: Volterra Kernel Approaches." *Annals of Biomedical Engineering*, 24: 250–68, 1996.

[30] Kowalski, M.E. and Jin, J.-M. "Karhunen-Loève Based Model Order Reduction of Nonlinear Systems." *Proceedings of the IEEE Antennas and Propagation Society International Symposium*, 1:552–5, 2002.

[31] Kunisch, K. and Volkwein, S. "Control of Burgers' equation by a reduced-order approach using proper orthogonal decomposition." *Journal of Optimization Theory and Applications*, 102(2):345–371, 1999.

[32] Kunisch, K. and Volkwein, S. "Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics." *SIAM Journal on Numerical Analysis*, 40:492–515, 2002.

[33] Lassaux, G. "High-Fidelity Reduced-Order Aerodynamic Models: Application to Active Control of Engine Inlets." Master's thesis, Dept. of Aeronautics and Astronautics, MIT, June 2002.

[34] Leake, R.J. and Liu, R.W. "Construction of suboptimal control sequences." *SIAM Journal on control and optimization*, 5:54–63, 1967.

[35] Lewis, F.L. and Syrmos, V.L. *Optimal Control*. Wiley, New York, NY, 1995.

[36] Lumley, J.L. "The Structures of Inhomogeneous Turbulent Flow." *Atmospheric Turbulence and Radio Wave Propagation*, pages 166–178, 1967.

[37] Mohammad, A.A. and De Abreu-Garcia, J.A. "A transformation approach for model order reduction of nonlinear systems." *Proceedings of the 16th Annual Conference of IEEE Industrial Electronics Society*, 1:380–3, 1990.

[38] Paduano, J., Merchant, A., Ashun, U., and Drela, M. "Design of an actively stabilized, near-isentropic supersonic inlet." AIAA Paper 2003-4096, 16th AIAA Computational Fluid Dynamics Conference, 2003.

[39] Paduano, J.D., Epstein, A.H., Valavani, L., Longley, J.P., Greitzer, E.M., and Guenette, G.R. "Active Control of Rotating Stall in a Low-Speed Axial Compressor." *Journal of Turbomachinery*, 115:48–56, 1993.

[40] Petzold, L. and Rathinam, M. "A new look at proper orthogonal decomposition." *SIAM Journal on numerical analysis*, 41(5):1893–1925, 2003.

[41] Phillips, J.R. "Model Reduction of Time-Varying Linear Systems Using Approximate Multipoint Krylov-subspace Projectors." *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 96–102, 1998.

[42] Phillips, J.R. "Projection frameworks for model reduction of weakly nonlinear systems." *Proceedings of the 37th Design Automation Conference*, pages 184–9, 2000.

[43] Rewienski, M. *A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems*. PhD thesis, Dept. of Electrical Engineering and Computer Science, MIT, June 2003.

[44] Rewienski, M. and White, J. "A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices." *Proceedings of the International Conference on Computer-Aided Design*, pages 252–7, 2001.

[45] Romanowski, M.C. "Reduced Order Unsteady Aerodynamic and Aeroelastic Models using Karhunen-Loéve Eigenmodes." AIAA Paper 96-194, 1996.

[46] Saridis, G.N. and Lee, C.S.G. "An approximation theory of optimal control for trainable manipulators." *IEEE transactions on systems, Man, and Cybernetics*, 9:152–159, 1979.

[47] Silva, W.A., Marzocca, P., and Librescu, L. "Nonlinear open/closed-loop aeroelastic analysis of airfoils via Volterra series." *AIAA Journal*, 42(4):673–686, 2004.

[48] Silveira, L.M., Kamon, M., Elfadel, I., and White, J. "A Coordinate-Transformed Arnoldi Algorithm for Generating Guaranteed Stable Reduced-Order Models of RLC Circuits." *Computer Methods in Applied Mechanics and Engineering*, 169 (3-4):377–389, February 1999.

[49] Silveira, M., Phillips, J., and Daniel, L. "Guaranteed passive balancing transformations for model order reduction." *Proceedings of the IEEE/ACM 39th Design Automation Conference*, 2002.

[50] Sirovich, L. "Turbulence and the Dynamics of Coherent Structures. Part 1 : Coherent Structures." *Quarterly of Applied Mathematics*, 45(3):561–571, October 1987.

[51] Vasilyev, D., Rewienski, M., and White, J. "A TBR-based trajectory piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and MEMS." *Chemical Engineering Science*, DAC:490–95, 2003.

[52] Vasilyev, D., Rewienski, M., and White, J. "Perturbation analysis of TBR model reduction in application to trajectory-piecewise linear algorithm for MEMS structures." *Proceedings of the Nanotechnology Conference*, 2:434–37, 2004.

[53] Verdon, J.M. "Review of Unsteady Aerodynamic Methods for Turbomachinery Aeroelastic and Aeroacoustic Applications." *AIAA Journal*, 31(2):235–250, February 1993.

[54] Wernli, A. and Cook, G. "Suboptimal control for the nonlinear quadratic regulator problem." *Automatica*, 11:75–84, 1975.

[55] Willcox, K. and Megretski, A. "Fourier series for accurate, stable, reduced-order models in large-scale applications." AIAA Paper 2003-4235, to appear SIAM Journal of Scientific Computing, 2004.

[56] Willcox, K.E. and Peraire, J. "Balanced Model Reduction via the Proper Orthogonal Decomposition." *AIAA Journal*, 40(11):2323–30, November 2002.

[57] Willcox, K.E., Peraire, J., and White, J. "An Arnoldi approach for generation of reduced-order models for turbomachinery." *Computers and Fluids*, 31(3):369–89, 2002.

[58] Zhang, Y., Henson, M.A., and Kevrekidis, Y.G. "Nonlinear model reduction for dynamic analysis of cell population models." *Chemical Engineering Science*, 58: 429–45, 2003.