Unsteady Aerodynamics for Aeroelastic Applications Using the Impulse Response Method

by

Randal Edmund Guendel

B.E., Vanderbilt University
(1998)

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 2000

© Massachusetts Institute of Technology 2000. All rights reserved.

Author ....
Department of Aeronautics and Astronautics
May 18, 2000

Certified by ....
Carlos E. S. Cesnik
Assistant Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by ....
Nesbitt W. Hagood, IV
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

SEP 07 2000

LIBRARIES

Aero

Unsteady Aerodynamics for Aeroelastic Applications Using the Impulse Response Method

by

Randal Edmund Guendel

Submitted to the Department of Aeronautics and Astronautics
on May 18, 2000, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Aeroelasticity is a critical issue in the design of aircraft and other aerospace vehicles, particularly those with highly flexible components. A reliable but efficient analysis tool is required to aid decision-making in the preliminary design phase. This thesis focuses on the unsteady aerodynamics component of the total aeroelastic system. Classically unsteady aerodynamics has been grounded on the Theodorsen function, which identifies the response of a 2-D wing section to harmonic pitch and plunge oscillations. Recently, however, the Aerodynamic Impulse Response has emerged, identifying a more fundamental aerodynamic response of a discrete-time system as that to a unit impulse. With this response, the response to any motion in the time domain can be easily predicted. This thesis examines the Aerodynamic Impulse Response method using an aerodynamic panel code, PMARC, to obtain impulse responses. The basic formulation of the method is limited to rigid-body analyses and is thus of limited use to aeroelastic studies. To this end, the method is extended to flexible-body responses by considering impulse distribution functions that are related to structural mode shapes of the body. Both linear and nonlinear responses are considered: the former uses convolution to generate arbitrary responses, the later the Volterra series. Linear results for both rigid and flexible bodies are encouraging. Predictions for a range of input motions closely match the unsteady response from PMARC for the same motion. However, for harmonic motion accuracy erodes for $f\Delta t < 0.05$, limiting the frequency range over which the model is accurate. Nonlinear responses are erratic and further study is required.

Thesis Supervisor: Carlos E. S. Cesnik
Title: Assistant Professor of Aeronautics and Astronautics

Acknowledgments

A project of this scale would have been impossible without the support and assistance of a veritable army of friends and associates. Thanks go foremost to my advisor, Prof. Carlos Cesnik. Without his guidance this project would never have come to completion. I'm grateful for all his advice along the way, both with respect to my thesis research and to my academic development.

I'm especially grateful to all the people at NASA Langley who have supported my research endeavors over the past few years. The LARSS program was invaluable in acquainting me with the research environment; I have many fond memories from three summers at NASA Langley. Thanks to Rafaela Schwan and the others in the Office of Education for making it all happen and, especially, for helping win the GEM Fellowship. I'm grateful to each of my mentors at Langley: Dr. Mark Lake, Greg Gatlin, and Dr. Keats Wilkie. Each gave me a tremendous opportunity to develop myself as a researcher. In addition, I'd like to thank Dr. William Milholen, who first introduced me to PMARC and who has provided willing support since then. Perhaps most importantly, Dr. Walter Silva. Without Walt and the time he spent introducing me to the Aerodynamic Impulse Response method, this project would have taken a very different form. Thanks, Walt, for giving me that boost and for your support along the way.

I also want to thank all the faculty, staff, and students of TELAC. Being the only aerodynamicist in a structures lab was awkward at times, but you all made me feel at home and offered valuable insight along the way. Special thanks go to Seth Kessler, for all the advice and working out coding puzzles with me; Dennis Burianek, for his words of wisdom as the lab's senior graduate student and for all his help with LaTeX; and Barbara Huppe, for combing through my thesis with me and for lending a willing ear on the occassions when my frustration exceeded my motivation. Outside of TELAC, Tony Chobot and Chris Dunn also deserve sincere thanks for helping to tame my computer system. And finally, grateful thanks go to Andrea Cook at Purdue University for struggling with PMARC alongside of me—the frequency of the word

"PMARC" in our correspondence was certainly frighteningly high.

Further thanks are extended to those at my undergraduate institution, Vanderbilt University, who have been instrumental in my development as a reseracher and engineer. In particular, thanks go to Prof. Ephrahim Garcia and Dr. Joel Barnett.

Nor could I have done this without the support of friends to keep me sane and (nearly) level-headed. Zac and Hootan, those tried and true souls met early at Vanderbilt who have stuck with me through the years, have unfailingly helped keep me afloat. Nothing brightens a weekend quite like a long discussion of music and literature with Zac or of the latest Formula 1 race with Hootan. And the new friends I've made here in Cambridge, thanks for everything. Memories of nights on the town with Jeff, Tony, and Dan will never fade. Jamie I can't thank enough for her simple optimism, her insistence that I need only to fix my "equation." And many thanks go to Seth, my roommate and officemate, for everything that we've shared and for just being there.

And of course none of this would have been possible without my loving and supporting family. Thanks to my parents for pushing me hard, for always emphasizing the value of a good education, and for always being behind me.

This research was partially supported by the Air Force Office of Scientific Research under the subgrant E-16-N96-G1.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A vital ingredient of the design of modern aerospace vehicles is an aeroelastic stability analysis. Aeroelasticity is the study of the interaction of a vehicle's structure and the forces exerted on it by the flow of a fluid around it. Though generally considered in its aerospace context, aeroelastic behavior extends into marine and civil structures as well. Aeroelastic instabilities occur when a perturbation to the structure, rather than being damped out, grows exponentially. Aeroelastic instabilities are typically catastrophic. A well-known example of an aeroelastic instabilty is the collapse of the Tacoma Narrows bridge in 1944. Wind flowing across the deck of the bridge excited an unstable vibration that caused the deck to "gallop" until finally breaking apart. Because of the catastrophic nature of aeroelastic phenomena, it is crucial that the aeroelastic behavior of a new flight vehicle be well understood prior to its production.

Aeroelastic models consist of an unsteady aerodynamics component and a structural dynamics component with an interface between the two. Structural dynamic models predict deflections of a structure due to an applied force, and, generally speaking, are well developed for both linear and nonlinear problems. Forces exerted on the structure due to the flow over it are computed with the unsteady aerodynamics component and fed back to the structural model. These forces are generally time and motion dependent. The unsteadiness of the flow is key: the forces exerted on a body in an unsteady flow field may be very different than those exerted on it in a steady flowfield. In particular, aerodynamic lags or transient repsonses become important in unsteady analyses. Thus, robust aerodynamic models are called for. In many cases, the bulk of the computational resources for an aeroelastic stabilty analysis must be devoted to the aerodynamics component. High-fidelity fluid analyses often require hours or tens of hours of machine time even on high-performance computers. Hence, there is considerable interest in developing methods that will produce an accurate result while not being prohibitively expensive to run. This is particularly important for preliminary design work, as the design process is iterative and it would not be feasible to run these high-fidelity analyses repeatedly.

Various methods to cut back the computational expense of unsteady aerodynamic analyses have been proposed and investigated. One such model that looks promising is the Aerodynamic Impulse Response (AIR) method, which has been recently developed at the NASA Langley Research Center [8, 9, 10]. This technique identifies the response of the aerodynamic system to a unit perturbation as the fundamental response of the system; given this fundamental response, the response of the system to an arbitrary input can be quickly calculated by convolution of the input and the impulse response. This method offers considerable computational savings over having to execute the aerodynamic analysis code for every input change. To date, the method has primarily been employed using CFD models to obtain the impulse response; however, because the theory is valid for arbitrary mathematical systems, any unsteady aerodynamics model can be used with the method. The benefit of this method is that the aerodynamic model needs only to be run once for a given set of freestream flow parameters (e.g., Mach number and Reynolds number) to obtain the impulse response. Once obtained, this impulse response can be used in conjunction with a convolution routine, much faster than the flow solver, to obtain the response to any input.

In some cases, further simplification of the model can be achieved by employing the impulse response method with a low-order aerodynamic model instead of a CFD code. Low-order aerodynamics models are obtained by approximating the nonlinear Navier-Stokes equations. The assumptions involved have the effect of reducing the accuracy of the solution and limiting the range of applicability of the model; however, when employed appropriately, low-order models provide preliminary design-level accuracy results with substantial computational savings. The aerodynamic panel method is one such low order aerodynamic model. By assuming inviscid, irrotational, and incompressible flow, the Navier-Stokes equations reduce to the Laplace equation for velocity potential. The panel method uses a distribution of aerodynamic singularities, solutions to the Laplace equation, to simulate the body and the flow around it. Once the distribution of these singularities—some combination of sources, doublets, and vortices—is known, the flow conditions at any point in the field can be found. Given the flow velocity, surface forces can be calculated with the unsteady Bernoulli equation.

The primary goal of this research is to develop an efficient and capable unsteady aerodynamic model that can be coupled with a structural model for the aeroelastic analysis of flexible flight vehicles. This is accomplished with the Aerodynamic Impulse Response method by employing an aerodynamic panel code to identify the responses to various impulse excitations. As a stand-alone aerodynamic model, using this method is a relatively simple way to extend the unsteady capabilities of the panel code. When formulated to identify responses of flexible-body motions and coupled with a structural dynamics code, it makes an efficient and reliable design-phase aeroelastic tool.

The unsteady aerodynamics model developed is applicable to arbitrary flight vehicle geometries operating in incompressible flow (traditionally, flows with Mach number less than 0.3). However, the recent interest within the aerospace community of employing highly flexible structures offers a unique test case around which this research centers. In particular, High Altitude, Long Endurance (HALE) Unmanned Aerial Vehicles (UAVs) have been a recent focus of research and development activities. Examples of this type of vehicle include the Aurora Flight Sciences Theseus and the AeroVironment Pathfinder as can be seen in Figures 1-1 and 1-2. The United States Air Force has fostered an active interest in these vehicles for a wide range of missions [11, 12], and civilian agencies such as NASA have additionally been heavily involved [1].

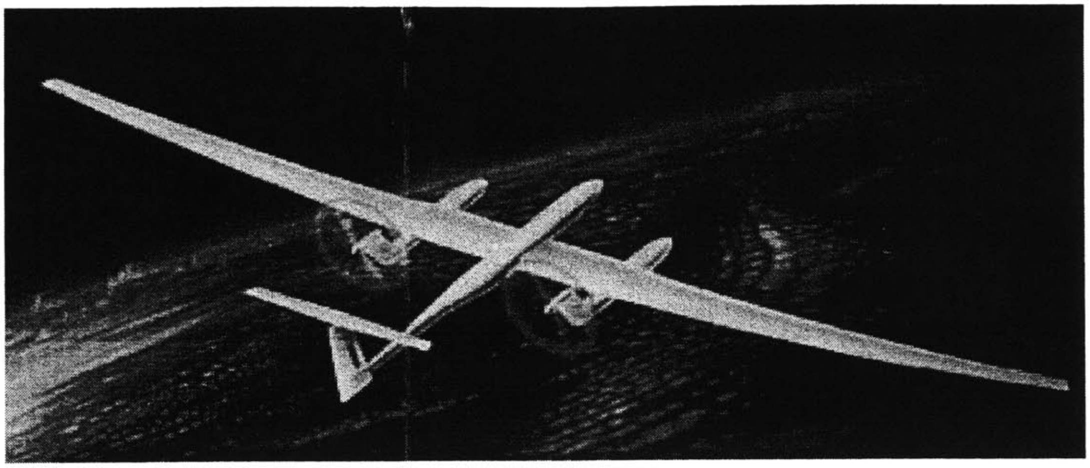


Figure 1-1: Aurora Flight Science's Theseus HALE UAV, with Endurance of 50 hrs at 88,000 ft [1]

These vehicles are commonly designed with exceptionally long and slender wings with aspect ratios as high as 30 to 35 because of the aerodynamic efficiency of such wings. Conventional design paradigms have dictated that wings be stiffened to prevent significant bending and twisting and possible associated aeroelastic instabilties. Wings of this length, however, would require an unreasonable amount of stiffening to prevent large deflections. Because weight is a premium commodity in the design of HALE vehicles, designers have turned to working with the inherent flexibility of the wings rather than trying to prevent it.

This research is primarily motivated by the need for a reliable and accurate aeroelastic analysis tool for the preliminary design of and the control-law development for highly flexible aerospace vehicles. High wing flexibility poses some interesting and complex engineering problems; namely, understanding the nonlinear aero-structural interactions and a robust active control system to stabilize the wing. Traditional designs encounter only small deflections that are relatively inconsequential in the analysis, and thus the problem often may be linearized. In a linear formulation like this, the aerodynamic forces are computed about the undeformed wing. In the case of high aspect ratio wings, however, tip deflections can be as much as 30% of the semi-

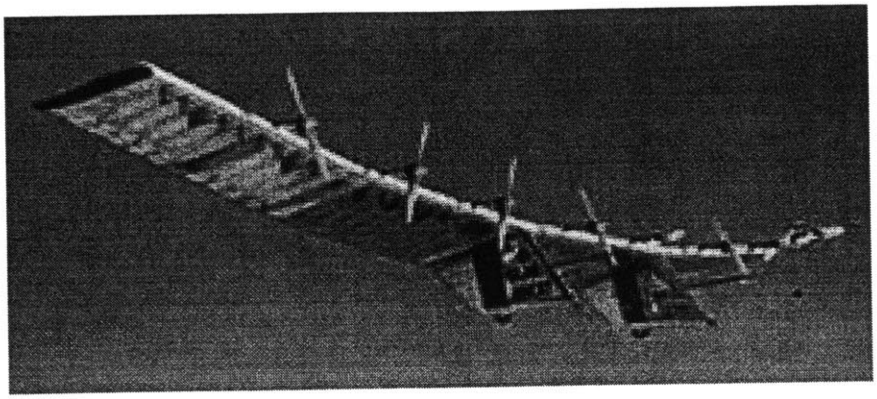


Figure 1-2: AeroVironment's Pathfinder HALE UAV, with Endurance of 16 hrs at 70,000 ft [1]

span. Deflections of this magnitude require nonlinear structural analysis as well as some way to account for the nonplanar lifting surface. Large twisting deformations also require the incorporation of nonlinear, high angle-of-attack aerodynamics and account of dynamic stall phenomena.

This research seeks to provide an accurate and computationally efficient aerodynamic tool for the aeroelastic analysis of highly flexible aerospace structures. The nature of the problem is ideally suited to the building block methodology. The foundation of the model is the aerodynamic panel code, to which additional "blocks" can be added. Each successive "block" extends the capabilities of the model. The first step involves the implementation of the linear Aerodynamic Impulse Response method for rigid-body motions. The model can subsequently be extended to capture limited nonlinear effects, inclusion of flexible-body modes, and incorporation of compressibility and dynamic stall effects. With this framework in mind, the specific objectives of this study are as follows:

- Reduce the computational effort required for unsteady aerodynamic analyses using the Aerodynamic Impulse Response Method

- Formulate the Aerodynamic Impulse Response method for flexible-body motions

- Implement and test the rigid- and the flexible-body Aerodynamic Impulse Response in an existing panel code

This research focuses on the linear formulation of the impulse response method; however, some initial studies of the the nonlinear form will be presented.

Chapter 2 of this thesis explores the background of this problem and highlights significant achievements by other researchers in the field, concluding with some comments on what has been lacking in developments to date and why a new formulation is important. Chapter 3 continues with an overview of the first component of the proposed formulation, the Aerodynamic Impulse Response technique. The second primary component of the research, the platform for the Aerodynamic Impulse Response, an aerodynamic panel code, is discussed in detail in Chapter 4. Chapter 5 discusses the modifications to PMARC necessary to implement the Aerodynamic Impulse Response method, detailing each the linear and nonlinear rigid-body and linear flexible-body implementations. Chapter 6 covers modeling procedures and details the generic wing geometries that are investigated. Preliminary tests prior to the modification of the panel code are reviewed in Chapter 7, followed by results of the three implementations and a discussion of these results. The final chapter concludes by summarizing the accomplishments of this research and highlighting future directions of study.

Chapter 2

Background

The ultimate goal of the research presented in this thesis is to contribute towards the development of a computationally efficient aeroelastic analysis tool for design-phase analysis of vehicles with a high degree of structural flexibility. This chapter opens with a brief introduction of the field of aeroelasticity. Bisplinghoff, Ashley, and Halfman provide a thorough examination of the subject [13]. There are two primary components to any aeroelastic analysis, structural dynamics and unsteady aerodynamics, and both are briefly introduced here, though unsteady aerodynamics will be the primary focus of this research. The most direct approach to an aeroelastic analysis is the direct solution of the coupled fluids and structural equations. Often referred to as Computational Aeroelasticity, this method has become a feasible one due to the development of computational tools and algorithms. Computational Aeroelasticity relies on the coupling of Computational Fluid Dynamics (CFD) and Computational Structural Dynamics (CSD). However, regardless of advances in computational sciences, for anything but simple configurations, the CFD/CSD method is very computationally demanding and hence is often not a practical solution. Most research in the field has been in the development of more efficient alternative approaches. The second section of this chapter surveys such work that has appeared in the literature concerning or contributing to unsteady aerodynamic analysis with application to the aeroelastic analysis of highly flexible structures. Besides the pure CFD method, two main approaches are presented: reduced-order models and lower-order models. While there is some overlap between these two, the primary distinction is that reduced-order models attempt to describe the system with a smaller set of "states" while lower-order methods seek to simplify the governing equations. Finally, these contributions are scrutinized with respect to the problem at hand, the aeroelastic response of highly flexible structures.

2.1 Aeroelasticity

Loosely stated, aeroelasticity is the study of the interaction of structural and aerodynamic forces on the static and dynamic behavior of a structure. Though most commonly a concern for aerospace vehicles, there are numerous other instances in which aeroelasticity is an important consideration, such as bridge and building design, undersea cables, and turbomachinery. Collar's Triangle, as shown in Figure 2-1, illustrates the interplay of aerodynamic, elastic, and inertial forces in classical aeroelasticity. More recent developments have added additional nodes to this triangle, controls and thermal effects, resulting in the so-called "aero-servo-thermo-elastic hexadron" [14]. In this research, thermal effects are negligible and will be discounted; however, the importance of a robust active control system necessitates keeping control-law design considerations in mind. This coupling of classical aeroelasticity with controls and flight dynamics is often referred to as aeroservoelasticity and is a field of considerable research interest recently.

Aeroelastic systems can be classified as either linear or nonlinear depending on the formulation of the aerodynamics and the structural dynamics. Linear aeroelasticity, because of its relative simplicity, has been the most studied. There are three primary phenomena associated with linear aeroelasticity. For level flight with steady flow conditions, inertial forces are neglected. Structural deformations are due only to the steady aerodynamic loading on the wing. There are two primary static aeroelastic phenomena, divergence and control reversal. Divergence is a result of uncontrolled wing twist; disturbances grow exponentially with responses of the form e^{kt}. Control reversal results when deflection of a control surface is negated by deformation of

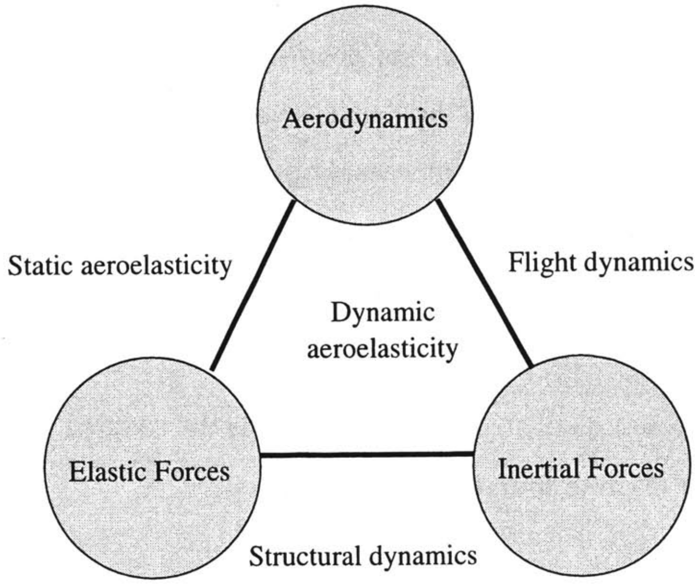


Figure 2-1: Collar's Triangle of Aeroelastic Forces

the wing. For an aircraft undergoing unsteady motion, the inertial terms must be retained. The dominant dynamic aeroelastic phenomenon is known as flutter. Flutter is an instability similar to divergence; it appears when oscillatory disturbances grow exponentially, with responses of the form $e^{(k+i\omega)t}$. Because all of these phenomena tend to be catastrophic in nature, it is important that they be identified and avoided in the design phase; the flight envelope is typically limited by the velocity at which these phenomena occur.

To better understand the aeroelastic behavior of real structures, particularly those such as highly flexible wings, nonlinear aerodynamic and structural effects must be considered. There are two primary differences from linear behavior introduced by nonlinearity. First, the onset of flutter becomes a function of the trim condition. In linear theory, steady conditions such as the root angle of attack do not appear in the formulation; this is not the case in nonlinear problems. Secondly, rather than diverging to infinity, nonlinear aeroelastic instabilties are generally amplitude-limited. This phenomenon is known as limit cycle oscillation (LCO) and occurs because once the twist of the wing exceeds the stall angle of attack, the wing loses lift. The aerodynamic loading on the wing consequently drops, and the instability does not grow past this point. Additionally, the geometric stiffening of the structure with deformation further contributes to the limitation of the amplitude of the oscillations. While this phenomenon may prevent catastrophic failure, the vibrations may make the ride uncomfortable and induce fatigue-related failures.

2.1.1 Structural Dynamics

The fundamental equation of structural dynamics is the coupled mass-spring-damper system as given in discrete form in Equation 2.1

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F\} \tag{2.1}$$

where $[M]$ is the mass matrix, $[C]$ is the damping matrix, $[K]$ is the stiffness matrix, $\{F\}$ is the forcing vector, and $\{u\}$ is the displacement vector. Time differentiation is shown in dot notation, so $\dot{(\,)} = \frac{\partial}{\partial t}$. This governing equation can be either linear or nonlinear depending on the form of the mass and stiffness matrices and the forcing vector. In aeroelasticity the forcing vector is generally motion dependent, a function not only of time but also of $\{u\}$ and its successive derivatives, so $F = f(u, \dot{u}, \ddot{u}, ..., t)$. It is often written as the sum of aerodynamic forces and controls forces:

$$\{F\} = \{F_{aero}\} + \{F_{controls}\} \tag{2.2}$$

Motion-dependent forces like damping, either structural or viscous, can be integrated into the forcing vector.

In the case of a linear problem, the most common solution method is to use modal techniques, in which the problem reduces to an eigenvalue problem. The set of eigenvalues represent the natural frequencies of the system, and the set of eigenvectors represent the corresponding shape functions, or modes, of the system. The modal representation relies on linear superposition to represent a complex shape as a series of simpler mode shapes. Often this series can be truncated, retaining only a few modes, without significant loss of accuracy. Thus,

$$u(x,t) \approx \sum_{r=1}^{N} \gamma_r(x) q_r(t) \tag{2.3}$$

where u is the deformed shape, γ_r are the mode shapes, q_r are the corresponding modal amplitudes, and x is a set of spatial coordinates. Other methods such as the Rayleigh-Ritz method and Galerkin's method are refinements of this technique. Modal methods have also been employed for nonlinear problems, though with much greater complexity [15, 16, 17].

For large linear and nonlinear problems, the solution in time of the governing equation is ordinarily carried out computationally using either the finite difference method or the finite element method.

2.1.2 Aerodynamics

In the context of aeroelasticity, aerodynamic modeling differs in two respects from conventional aerodynamic analyses. First, the flow is inherently unsteady. Second, it is primarily concerned with determining $\{F_{aero}\}$ from Equation 2.2—that is, the surface pressure forces acting on the body. A full description of the flowfield is not required for an aeroelastic analysis.

The most pronounced effect of unsteady flow is aerodynamic lag. Where in steady flow the response of the body is a function of its geometry only, in unsteady flow, higher-order time derivatives influence the response. The result is that the response is time-shifted from the motion. In addition, the amplitude of the response is affected by non-circulatory inertial effects. Finally, in unsteady flow, the nonlinear phenomenon of stall behaves differently than in steady flow. In unsteady flow, loss of lift often is delayed to a pitch angle significantly larger than the static stall angle of attack. Dynamic stall is a cyclic process and manifests itself in various forms throughout the motion of the wing. The ONERA stall model is a semi-empirical model that is often used to account for dynamic stall phenomena [18].

As mentioned in the Introduction, there is a wide range of aerodynamic models to choose from, ranging from the most complex, the fully nonlinear, three-dimensional Navier-Stokes equations, to the simplest, the two-dimensional linear potential equations. The Navier-Stokes equations are an analytically exact representation of real, unsteady flowfields accounting for viscosity, compressibility, flow rotationality, and thermal effects. Because of the complexity of the problem, though, approximations to the full Navier-Stokes equation set are often employed. Some of these simplified models will be seen later in this chapter in the section on lower-order aerodynamic models.

Generally either the finite difference method or the boundary element method (or, panel method) is used to solve the aerodynamic governing equations in time and space, though recently the finite element method has been growing in popularity.

2.1.3 Aerodynamics and Structures Coupling

As is evident from Equations 2.1 and 2.2, there is a close coupling of the aerodynamics and structures components of the aeroelastic system. Structural deformations affect the aerodynamic loads, and changing the aerodynamic loads further affects the structural deformations. There are several methods to solve the coupled equations, often broken into three categories: unified methods, tightly-coupled methods, and loosely-coupled methods [19, 20, 21, 22, 23]. Unified methods combine the structural and fluids equations into a single formulation and solve it directly. Tightly-coupled methods solve the structural and fluids equations separately at each time t, iterating for a converged coupled solution; loosely-coupled methods are similar though interaction of the two systems is only considered after several time steps.

The coupling of the structural and aerodynamic components is generally complicated by two factors, first the coordinate frame and discretization method, and second, the interface between structural and aerodynamic surfaces. Most commonly, fluids problems are solved with finite differences based on an Eulerian coordinate frame, whereas structural problems are typically solved using a Langrangean-based finite element scheme. Once this conflict is resolved, data must be transferred from

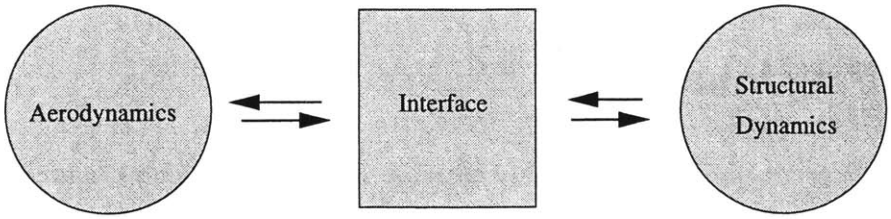


Figure 2-2: Aero-Structural Coupling Through an Interface

one component to the other. This is generally complicated, requiring interpolation and extrapolation of data, because the aerodynamic and structural meshes rarely coincide. This remains a subject of considerable research interest. There are currently a variety of interface methods being used [24, 25]; however, this subject is beyond the scope of this research, and it will be assumed that the wing is modeled as a set of 2-D sections and that the 3-D lift distribution on the wing can be modeled as a concentrated force at the aerodynamic center (quarter-chord) of each section.

2.1.4 Controls Coupling

The advent of active control systems in the past two decades has spurred the need to couple control forces into the aeroelastic system as Equation 2.2 suggests. This field of study is commonly known as aeroservoelasticity. The aeroelastic problem becomes more complex in this case because controls add an additional source of energy to the problem [26]. While controls considerations are mostly oustide the scope of this research, there is nonetheless some impact. Highly flexible wings such as those considered here will require a robust active control system to remain stable, hence any aeroelastic model must be formulated to allow for future coupling with a controls analysis. This places two primary requirements on the aeroelastic model. First, it is desireable to have a system which may be cast in state-space form [27]. Secondly, because of matrix size limitations in controls analysis, the state-space system must be small. This has been one of the primary driving motivations of Reduced Order

Modeling (ROM) as is discussed later in this chapter.

2.2 Previous Work

Because of recent demands for highly capable and highly efficient aeroelastic models, there is a sizeable body of work for this research to draw upon. This section will review some of the more notable developments in unsteady aerodynamic modeling. Computational fluid dynamics models (CFD) are briefly introduced; while still impractical for preliminary design, they set the standard for computational accuracy. The unsteady aerodynamics models reviewed subsequently are organized into two categories, reduced-order models (ROMs) and lower-order models. Reduced-order models use various techniques to extract important modes or states from a flow analysis to represent the flow more cost-effectively with minimal loss of accuracy. The reduced-order methods are not aerodynamic models themselves, but rather they work in conjunction with an aerodynamic model. Lower-order models, also referred to as simplified aerodynamics models, directly manipulate the governing equations to reduce computational costs. Though represented independently, there is some overlap between these methods because a lower-order model may be used in conjunction with a reduced-order method for additional savings. The section closes with a brief summary of the developments presented here and an examination of their limitations towards application to highly flexible aircraft.

2.2.1 CFD Methods

While the exact boundaries of CFD are somewhat nebulous, in this thesis CFD will refer to computational methods employing a finite difference or finite element scheme to numerically solve the Navier-Stokes or Euler equations. The Navier-Stokes equations, as given in Equation 2.4, analytically describe fluid motions. In their most general form, the Navier-Stokes equations are given as follows:

$$\rho \frac{D\mathbf{V}}{Dt} = \rho \mathbf{g} - \nabla \cdot \tau_{ij} \tag{2.4}$$

Indicial notation is used here to express the equations compactly, where $\mathbf{V}$ is the velocity vector, $\mathbf{g}$ is the body force (generally negligible for fluids problems), and τ is the stress tensor. For practical purposes, however, solutions of the Navier-Stokes equations are approximate. The stress may be modeled linearly, giving

$$\rho\frac{D\mathbf{V}}{Dt} = \rho\mathbf{g} - \nabla p + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + \delta_{ij}\lambda div\mathbf{V}\right] \tag{2.5}$$

where μ is the coefficient of viscosity, and λ is the bulk viscosity. Consideration of turbulent flow requires extensive modeling that is still the subject of considerable research. The Euler equations are the same with zero viscosity.

These methods could be called "brute force" methods because limiting assumptions are few but computational cost is high, often requiring tens of hours on high-performance computational platforms. Three-dimensional CFD analyses require discretization of the entire flowfield with a grid density near the body surfaces fine enough to resolve boundary layer phenomena, typically about ten elements across the boundary layer thickness. This results in problems with tens of millions of degrees of freedom, generally a considerably larger problem than the coupled structural analysis. CFD analyses produce highly accurate results, but for applications outside of basic research and detailed analyses the method is unwieldy. Edwards provides a thorough review of the application of CFD analyses to aeroelastic problems [28].

2.2.2 Reduced-Order Methods

Because of the high computational costs involved with CFD methods, researchers have been investigating options to improve the efficiency of the analyses. One or more of three tactics are generally employed. The CFD scheme itself can be optimized, the governing equations can be simplified where appropriate, or an additional analysis tool in conjunction with the flow solver can be used to extract only the most important information about the flow. Reduced-order methods are of this last sort. It is important to note that a reduced-order aerodynamic model is comprised of two parts. First, an appropriate aerodynamics model is required to describe the flow.

Then a reduced-order method is applied to make that model more manageable. The reduced-order method itself is only a mathematical conditioning tool and has nothing to do with the aerodynamics. Three such reduced-order methods will be reviewed here, the eigenanalysis method, proper orthogonal decomposition (POD), and the Arnoldi basis vector method. These three methods all share a common tactic: in each method, a small set of orthogonal basis vectors describing the system is assembled. The Aerodynamic Impulse Response is an additional reduced-order type method; however, discussion of this method will be deferred to Chapter 3.

Eigenanalysis

Modal analysis is a technique in widespread use in structural dynamics. With this technique, complex structural deformations are described as the sum of a set of simpler shape functions, which describe the geometry only, and a separate set of amplitude functions describing the amplitude of each shape function with respect to time. In conjuction with eigenanalysis techniques, these shape functions are orthogonal normal modes. If the dynamic behavior of a structure can be represented this way, it can be expected that the same technique might apply to the analysis of dynamic flows. The aerodynamic eigenanalysis method that has been developed primarily by Hall, Dowell and their co-workers [2, 29, 30, 31] does just this.

Eigenanalysis methods are based on the time linearization of the fluid system by casting the problem into the frequency domain. The unsteady motion is assummed to be small with respect to the mean flow, and the motion has the form $e^{i\omega t}$. This harmonic oscillation causes the time derivative operator to drop out of the equation and results in a set of linear equations that can readily be solved. This linear analysis can be extended to nonlinear formulations such as the Navier-Stokes or Euler equations; however, the method is limited to small disturbances about a nonlinear static flowfield [29]. Even in the presence of nonlinear features such as shocks or separated flow, this linearization remains valid for sufficiently small disturbances.

This eigenanalysis method has been applied to a variety of aerodynamic formulations. The seminal work on the method is by Hall [2]. The foundation of his work is a three-dimensional vortex lattice method, which is discussed in greater detail later in this chapter. This formulation is a potential flow model, assuming the flowfield to be incompressible, inviscid, and irrotational. Point vortex elements are placed on each element of the discretized wing and wake. Flow tangency (also referred to as the non-penetration boundary condition) requires that at each element the velocity induced by the distribution of discrete vortices matches the downwash velocity that is a product of the unsteady motion of the wing. Then, given the time history of the downwash and the vortex strength at time n, the vortex strength at time n+1 is

$$\mathbf{\Gamma}^{n+1} = \mathbf{A}^{-1}[\mathbf{w}^{n+1} - \mathbf{B}\mathbf{\Gamma}^{n}] \tag{2.6}$$

where $\mathbf{\Gamma}$ is the vortex strength, $\mathbf{w}$ is the downwash, and $\mathbf{A}$ and $\mathbf{B}$ are coefficient matrices derived from the vortex lattice kernel function, the conservation of circulation, and wake convection equations. When coupled with a simple linear structural dynamics model, this then yields aeroelastic equations as follows:

$$\begin{bmatrix} \mathbf{A} & \mathbf{E} \\ \mathbf{C}_2 & \mathbf{D}_2 \end{bmatrix} \begin{Bmatrix} \mathbf{\Gamma} \\ \mathbf{q} \end{Bmatrix}^{n+1} + \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{C}_1 & \mathbf{D}_1 \end{bmatrix} \begin{Bmatrix} \mathbf{\Gamma} \\ \mathbf{q} \end{Bmatrix}^{n} = \begin{Bmatrix} \mathbf{w}_g \\ \mathbf{0} \end{Bmatrix}^{n+1} \tag{2.7}$$

where $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{D}_1$, and $\mathbf{D}_2$ are coefficient matrices from the structural formulation, $\mathbf{E}$ describes the relationship between the airfoil motion and the downwash, and $\mathbf{w}_g$ is downwash due to external sources such as gusts.

Eigenanalysis can proceed from Equation 2.7; however, a reduced order model may be generated by writing Equation 2.6 in terms of the normal natural modes, with

$$\mathbf{\Gamma} = \mathbf{X}\mathbf{c} \tag{2.8}$$

where $\mathbf{X}$ is a matrix whose columns are the right eigenvectors of the system and $\mathbf{c}$ is an amplitude vector. Results show that the solution can be represented accurately using only the first few eigenmodes [2]. By retaining only m of the original N modes and manipulating Equation 2.7, the size of the problem and consequently the computational effort required are considerably reduced. However, Hall notes that not all problems are well represented by only a few eigenmodes; in these cases, a static correction must be added to the formulation.

Eigenvectors were extracted using standard Eispack routines for small problems and specialized methods based on the Lanczos algorithm for larger problems [29]. Figure 2-3 shows some of Hall's representative results. Heeg extends Hall's work with the vortex lattice method to study the method as an aeroelastic tool in greater detail [30].

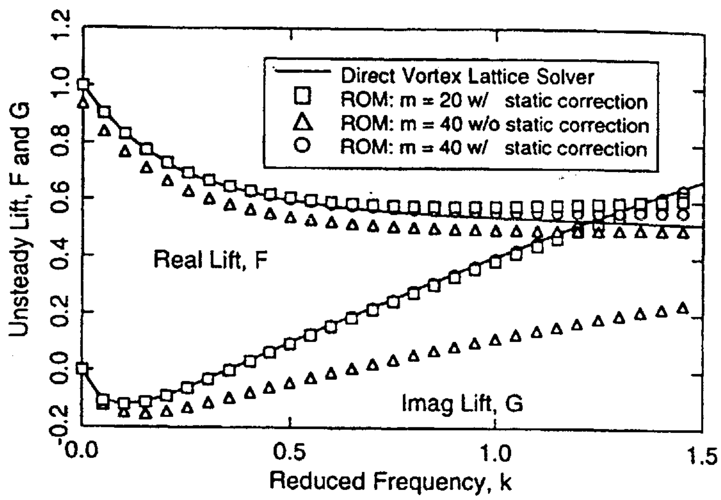


Figure 2-3: Real and Imaginary Lift Components Using Eigenanalysis [2]

Florea *et al.* use the same techniques with the full potential equation instead of the linear potential equation [31]. This allows for density variations (i.e., compressibility) in the flow but cannot account for strong shocks, and thus is useful for the regime of high subsonic and incipent transonic flows. In this case, the governing equation employed is

$$\frac{\partial \hat{\rho}}{\partial t} + \nabla \cdot (\hat{\rho} \nabla \hat{\phi}) = 0 \tag{2.9}$$

where $\hat{\rho}$ is the local density and $\hat{\phi}$ is the velocity potential. After coupling with structural dynamic equations, a reduced order model was obtained using two methods, the eigenmode technique as employed by Hall and by using Arnoldi-Ritz vectors, a method similar to those described in the next section. Florea also finds that the unsteady flowfield can be well represented with only a few eigenmodes. He also notes that the Arnoldi-Ritz reduction method is more efficient than the eigenanalysis method when the structural excitations are known a priori.

Approximate Eigenanalysis Methods

Proper Orthogonal Decomposition, commonly referred to as POD, and Arnoldi basis vectors are two other methods of reducing the order of a system. Both are methods that approximate the eigenmodes of a system. As will be seen, the two methods are actually quite similar. The basic goal of both methods, as for the eigenanalysis method, is to take a set of data, as might be obtained from a CFD analysis, and indentify a small set of orthogonal basis vectors that will describe that system. Obviously, there is some loss of information; however, with some knowledge of the nature of the flow, the basis vector identification process can be tuned such that relevant features of the flow are captured. In effect, these methods single out the dominant effects within the flow and discard less important effects.

The set of POD basis vectors is calculated by first finding a set of M *snapshots* of the flow with whatever flow solver is being used. In the time domain, these snapshots are merely the instantaneous solutions at selected times. Because, however, it is often difficult to capture the appropriate dynamics of the system, a more effective approach is to linearize and cast the problem into the frequency domain [3]. In this case, the snapshots represent the solutions for a set of excitation frequencies spanning the frequency range of interest. It is important to note that the choice of sample frequencies has a strong effect on the accuracy of the resulting reduced-order model [3]. Once these snapshots are obtained, the basis vectors $\mathbf{\Phi}_i$ are then linear combinations of the snapshots, and the overall solution $\mathbf{U}'$ is a linear combination of the basis vectors.

The Arnoldi vector method is similar except that, instead of finding a set of basis vectors for a range of frequencies, it determines multiple basis vectors for a single frequency. Willcox gives an algorithm for computing the Arnoldi vectors in [3]. It is worth noting that computing a single Arnoldi vector for M frequencies produces an equivalent data set to finding M snapshots over the same range of frequencies [3]. Willcox postulates that combining the POD and Arnoldi methods could produce a very efficient model.

Thomas *et al.* uses the POD method to reduce the order of the 2-D nonlinear Euler equations for the investigation of unsteady transonic flow [32]. To time-linearize the problem, small disturbances about a nonlinear steady-state condition are considered, casting the problem into the frequency domain. The governing equation is

$$\frac{\partial \hat{\mathbf{u}}}{\partial t} + \frac{\partial \mathbf{F}(\hat{\mathbf{u}})}{\partial x} + \frac{\partial \mathbf{G}(\hat{\mathbf{u}})}{\partial y} = \mathbf{0} \tag{2.10}$$

where $\hat{\mathbf{u}}$ is the vector of conservative variables and $\mathbf{F}$ and $\mathbf{G}$ are flux vectors. The authors employ a second-order, explicit, cell-centered, finite-volume, Godunov-based method utilizing the Roe approximate Riemann solver. Using this scheme an ensemble of solutions for various pitching and plunging motions is assembled. From this solution set, a reduced-order model was obtained by employing POD techniques. The reduced-order fluid model was coupled with a two degree-of-freedom structural airfoil model to form a reduced-order aeroelastic model. The authors study the pitch response of an airfoil comparing the POD method to a CFD simulation [32]. The number of snapshots used to construct the basis and the number of POD modes retained are varied. Plotting real and imaginary parts of the lift coefficient transfer function, excellent matching of the POD and CFD results when retaining all of the POD vectors (from 21 total snapshots) was found; reasonable accuracy is demonstrated for other cases.

Willcox employs both methods to develop a reduced-order aeroelastic model for the aeroelastic analysis and control of turbine blades [3]. Here, too, the time-linearized 2-D Euler equation is used. The POD model was constructed using twenty snapshots; six POD modes were retained. Six vectors were used in the Arnoldi model as well. The eigenvalue spectra of these two aerodynamic systems show that the Arnoldi approach captures a greater portion of the system's dynamics than does the

POD method. It is noted that increasing the number of basis vectors in the POD model does not add any additional information since the model is only valid over the frequency range the original snapshots were sampled over. On the other hand, the use of additional Arnoldi vectors increases the portion of the system dynamics the model will represent [3]. Results with just the reduced-order aerodynamics models match well with full CFD solutions. Figure 2-4 shows the response of a turbine blade section to a pulse using the POD and Arnoldi models and a full CFD simulation. These two reduced-order aerodynamics models were subsequently coupled with a 2-D, two degree-of-freedom structural model of a turbine blade to study the aeroelastic response. The resulting model is ideal for controls analysis because of the small number of states.

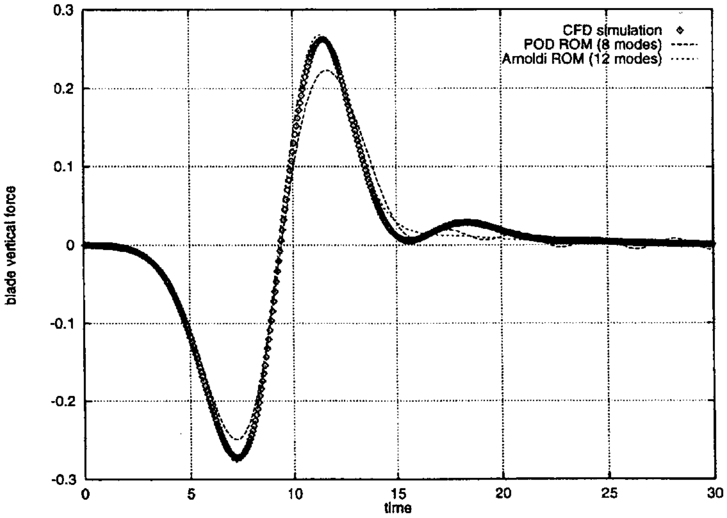


Figure 2-4: Comparison of POD and Arnoldi Models to the CFD Simulation of a Pulse Response [3]

The Arnoldi-Ritz method that was employed by Florea *et al.* [31] with the full potential equation, discussed in the previous section, is a similar method to Willcox's Arnoldi method [3].

2.2.3 Simplified Aerodynamics Models

The study of highly flexible vehicles has mostly been limited to lower-order, or simplified, aerodynamics models. Lower-order aerodynamics models essentially provide approximate solutions to the Navier-Stokes equations by making a series of simplifying assumptions. One of the most immediate simplifications is the assumption of inviscid flow; often empirical correlations can be used to add viscous effects to an inviscid analysis more efficiently than retaining the viscous terms throughout the calculation. Incompressibility and irrotationality are other simplifications taken advantage of by most of the methods reviewed here. Examples of simplified aerodynamics models are abundant. Those presented here include inflow methods, vortex lattice methods, panel methods, and lifting line methods, all with various formulations, and several other techniques. Anderson provides an overview of the basic theory and implementation of several such methods [33] and Bisplinghoff, Ashley, and Halfman discuss unsteady formulations [13].

Peters' Finite State Inflow Method

Peters and his co-workers have developed a modal-type method [4, 34, 35] that has been employed by Patil and co-workers in the aeroelastic analysis of a highly flexible wing [6, 36]. Two primary differences between this method and the eigenmode approach mentioned earlier are worth noting. First, this method is self-contained; that is, the reduced-order form is achieved from within the aerodynamic formulation. Secondly, the aerodynamic modes used to describe the flow are not orthogonal eigenmodes but are expansions of the Glauert transformation with the form

$$f(x) = \sum_{n=0}^{\infty} A_n \cos(n\phi) \tag{2.11}$$

Peters prefers these Glauert expansion modes to the eigenmodes because they are more physically based. For instance, while the physical nature of any of the eigenmodes is entirely ambiguous, the first airfoil deformation mode using Peters' method corresponds to a rigid body plunging motion, the second mode corresponds to rigid body pitch, and the third mode corresponds to a camber change.

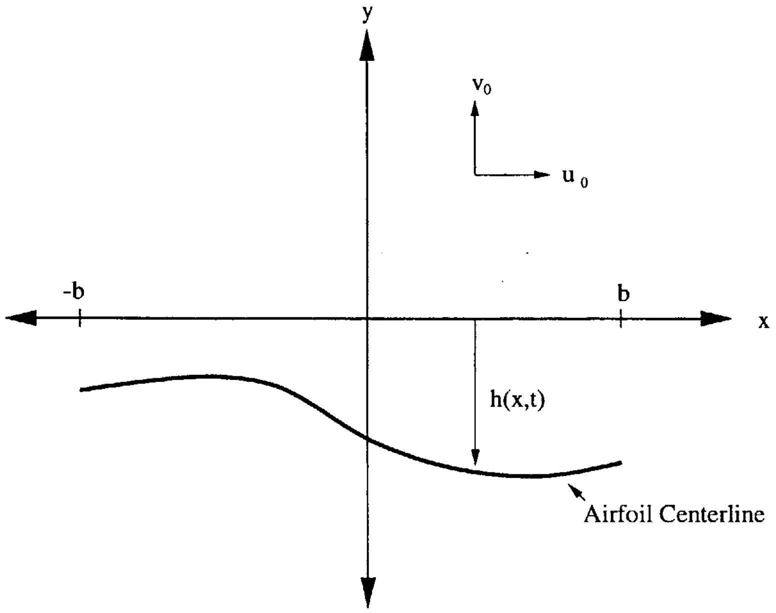


Figure 2-5: Airfoil Coordinate System for Peters' Finite-State Inflow Model

In the 2-D formulation [34], a thin airfoil undergoing small motions is considered. This allows the bound vorticity to be placed along the x-axis. Figure 2-5 shows the coordinate frame employed in the formulation. The flow is calculated to satisfy the non-penetration boundary condition [35],

$$\bar{v} = u_0 \frac{\partial u}{\partial x} + \frac{\partial h}{\partial t} + \frac{v_1 x}{b} - \lambda \tag{2.12}$$

where $\bar{v}$ is the downwash due to bound vorticity, u_0 and v_0 describe the uniform freestream velocity in the x- and y-directions, respectively, v_1 is a velocity gradient due to coordinate translation and rotation, and λ is the downwash due to all other sources (the shed wake, gusts, etc.). Glauert transformation expansions in the form of Equation 2.11 are found for the downwash due to bound vorticity, the downwash due to other sources, and the airfoil deformation, $\frac{\partial h}{\partial x}$. Combining these expansions and expressions for vorticity, the general airloads are found to be [35]:

$$u_0(w_0 - \lambda_0) = \tau_0 \tag{2.13}$$

$$b(\dot{w}_0 - \frac{1}{2}\dot{w}_2) + u_0 w_1 = \tau_1 \tag{2.14}$$

$$\frac{b}{2n}(\dot{w}_{n-1} - \dot{w}_{n+1}) + u_0 w_n = \tau_n \quad \text{for} \quad n > 2 \tag{2.15}$$

where $w_n = \gamma_n + \lambda_n$ is the relative motion of the airfoil (where γ_n is the Glauert expansion for the bound vorticity) and τ_n are the Glauert expansion coefficients for $\frac{\Delta P}{\rho}$. Finally, Equation 2.13 is rewritten in a form amenable for aeroelastic analyses. This involves changing from relative velocities w_n to actual airfoil deformations h_n and forming the generalized forces L_n associated with each airfoil deformation. This process yields

$$L_0 = [-2\pi\rho u_0(w_0 - \lambda_0) - \pi\rho u_0 w_1 - \pi\rho b(\dot{w}_0 - \frac{1}{2}\dot{w}_2)]b \tag{2.16}$$

which is the negative lift per unit span, and

$$L_1 = [\pi\rho u_0(w_0 - \lambda_0) - \frac{1}{2}\pi\rho u_0 w_2 - \frac{1}{8}\pi\rho b(\dot{w}_1 - \dot{w}_3)]b \tag{2.17}$$

which is the pitching moment per unit length about the midchord normalized on the semi-chord. The remaining modes are written as

$$L_n = \frac{1}{2}\pi\rho b(\tau_{n-1} - \tau_{n+1}) \tag{2.18}$$

Findings show that good accuracy can be achieved with this model by retaining only several modes. Peters also shows that his finite-state inflow model is easily adjustable to empirical corrections. In this manner he is able to incorporate the ONERA stall model to account for dynamic stall effects. Prediction of dynamic stall effects is an essential ingredient in the final analysis of a highly flexible wing undergoing large deformations.

Some of Peters' results using this model (without dynamic stall corrections) are shown in Figure 2-6. The figure compares the real part of the Theodorsen function as computed exactly (XCT in the figure) to that computed with the finite state method with four states, where the expansion coefficients are determined in two ways, a binomial expansion (BNX) and an augmented least squares fit (AUG). An even better fit is found by retaining eight states.

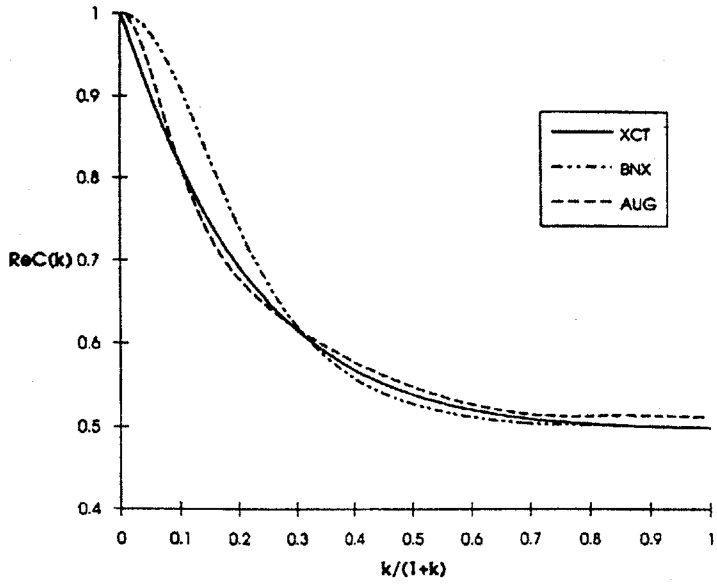


Figure 2-6: Comparison of Results Using the Peters Inflow Model with Theodorsen Theory [4]

Lifting Line Methods

Classical lifting line theory was an early contribution to the study of three-dimensional aerodynamic bodies by Prandtl [33]. An infinite set of horseshoe vortices are superimposed along the span of a 3-D wing to produce a continuous spanwise circulation distribution, $\Gamma(y)$. As described by the Biot-Savart law, each of these horseshoe vortices induces a downwash velocity on the wing that produces an induced angle of attack which must be added to the effective angle of attack. An integrodifferential equation for the circulation distribution results [33]:

$$\alpha(y) = \frac{\Gamma(y)}{\pi V_\infty c(y)} + \alpha_{L=0}(y) + \frac{1}{4\pi V_\infty}\int_{-b/2}^{b/2} \frac{d\Gamma/d\bar{y}}{y-\bar{y}} d\bar{y} \tag{2.19}$$

where α is the geometric angle of attack, $\alpha_{L=0}$ is the angle of zero lift, c is the chord distribution, and y measures along the span from $-\frac{b}{2}$ to $\frac{b}{2}$. In practice, a discrete distribution of horseshoe vortices is used. This gives a piecewise continuous circulation distribution and the integrals in Equation 2.19 are replaced with summations. With the circulation distribution in hand, the lift can be calculated by the Kutta-Joukowski theorem,

$$L'(y) = \rho_\infty V_\infty \Gamma(y) \tag{2.20}$$

A Fourier series expansion of $\Gamma(y)$ is often employed to describe the circulation in a mode-like form.

Drela utilizes an extension of this theory for the aerodynamics component in the integrated simulation model ASWING [5]. This model integrates aerodynamic, structural, and control-law design into a single package ideal for prelimary design work. The aerodynamics analysis in ASWING features an unsteady version of the lifting line theory, a wind-aligned wake, and the Prandtl-Glauert compressibility correction. Air loads are computed in terms of a Fourier series, and the unsteady wake is accounted for with an empirical lag term, resulting in the following expression for the induced downwash velocity at a control point:

$$(\vec{V}_{ind})_{c.p.} = \sum_{k=1}^{K} \vec{w}_k A_k(t) + \vec{w}_\infty V_\infty(t) - \frac{b}{V_\perp}\frac{\partial \Gamma}{\partial t}\hat{n}_{c.p.} \tag{2.21}$$

where $\vec{w}_k$ is the downwash induced by the horseshoe vortices per the Biot-Savart law, $\vec{w}_\infty$ is externally induced downwash, such as from a gust, A_k is the set of Fourier coefficients, V_∞ is the freestream velocity, and the final term is the empirical lag term. Lift along the wing span is computed two-dimensionally using a lumped vortex at each span station with a flow tangency boundary condition. Harmonic, small amplitude pitch and plunge are assumed and the empirical lag factor, b, is selected to match classical Theodorsen results for high reduced frequencies. Figure 2-7 shows that Drela's numerical results for flutter speed with respect to mass centroid are a near approximation to classical Theodorsen results.

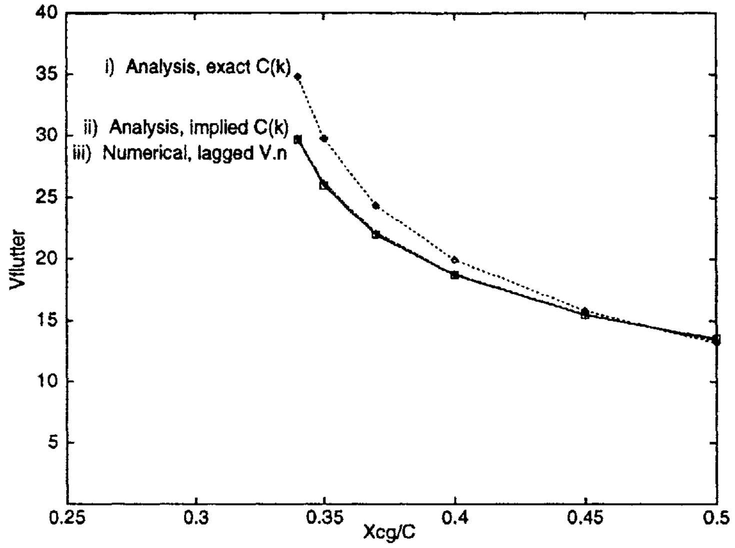


Figure 2-7: Flutter Speed Predicted with Various Models [5]

Vortex Lattice Methods

Vortex lattice methods are also popular for aeroelastic analyses. This method was used successfully as a platform for the eigenanalysis method discussed earlier [2, 30]. The vortex lattice method is an extension of Prandtl's lifting line method, which is limited to straight wings with a high aspect ratio. Instead of distributing a series of horseshoe vortices in a single line along the span of the wing, the vortex lattice method distributes horseshoe vortices along the span and chord of the wing. The Biot-Savart law is again used to calculate a downwash velocity induced by the vortices, although here a surface is considered instead of a line.

Mracek and Mook use a vortex lattice-type method with triangular elements to couple dynamics and controls to the aerodynamics analysis [37]. Dynamics equations of motion, aerodynamics, and control inputs are coupled by way of a fourth-order predictor-corrector scheme for a simultaneous solution for all three components. Using this formulation, a low aspect ratio delta wing with control surfaces is studied. The method is found to successfully predict the performance of a control law.

A similar study was performed by Hall *et al.*, who also utilized the vortex-lattice method in an integrated aerodynamic, structural, and controls model for the analysis of actively-controlled wings undergoing gust loads [38]. In this case, a high aspect ratio wing, as might be found on HALE vehicles, was studied under gust loading. While the test case did not prove successful in controlling the response to a gust load, it did illustrate the capability of the model.

Patil and his co-workers compare a vortex lattice method and Peters' finite state method to investigate the aerodynamic effects of large deformations for aeroelastic problems [6, 36]. A structurally nonlinear high aspect ratio wing was studied with both methods, where Peters' method neglects the nonplanar geometry of the deflected wing. Figure 2-8 shows results of a flexible wing at 25 m/s and a steady angle of attack of 10°. The spanwise lift distribution at steady-state equilibrium is shown for the Peters model, labeled "flexible: linear," and the vortex-lattice model, labeled "flexible: nonlinear." (Rigid results are shown as well.) These results indicate that 3-D nonplanar aerodynamic modeling of the wing geometry offers only slight improvement on the planar model of Peters. In the aeroelastic context, there was a 5% margin in the predicted flutter frequencies and 1.5% margin in the predicted flutter speed.

Panel Methods

Panel methods are also a popular alternative. In principle similar to the vortex lattice method, they use a distribution of velocity sources and doublets or pressure doublets rather than vortices; indeed, panel methods employing pressure doublets are often called doublet lattice methods. Sources and doublets, as are vortices, are fundamental solutions to the potential flow equation, the most fundamental form of fluid flow equations. This is a linear, inviscid, incompressible method, though some nonlinear effects can be incorporated with advanced wake models. The theory of

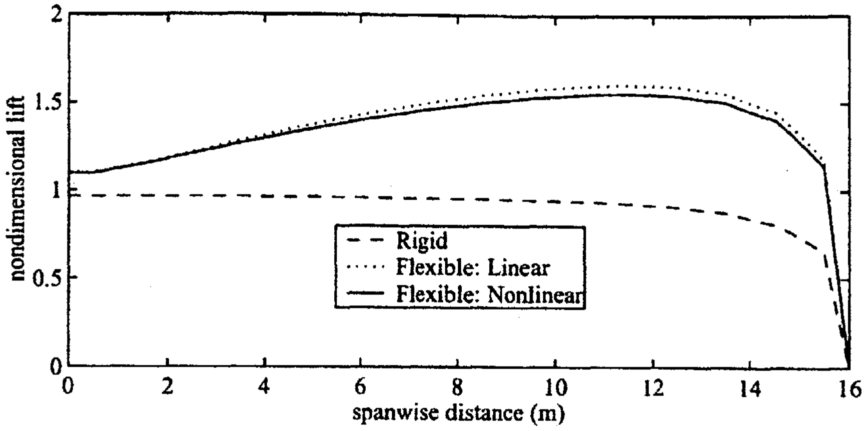


Figure 2-8: Spanwise Lift Predicted with Various Models [6]

panel methods will be discussed in greater detail in Chapter 4.

Katz and Maskew employ a velocity source/doublet panel method to predict the unsteady aerodynamic response of a complete aircraft configuration [7], a challenging task for Navier-Stokes methods because of the incredible computational cost. The use of simplified fluid dynamics equations allows for greater geometric complexity while keeping computational cost to a minimum. The panel code employed in [7], VSAERO [39], is a steady-flow code, and several modifications were needed to adapt it for application to unsteady flows. A time-stepping wake was added so that the wake would convect with the local velocity and roll up, and the non-penetration boundary condition was modified to include a rotation component. Finally, the local velocities used in Bernoulli's equation to calculate pressure forces from the velocity potential were modified to include the rotation component. The resulting model was applied to several configurations: a wing undergoing periodic heaving, a rotor in hover, a two-dimensional wing airfoil undergoing pitching oscillations, pitching oscillations of a complete aircraft configuration, and a delta wing with separated flow at the leading edge. Figure 2-9 illustrates one of the results of this study: a wing's response to plunging motion compared to the theoretical response due to Wagner [7].

One of the most versatile of the panel codes is PMARC [40], a code developed in

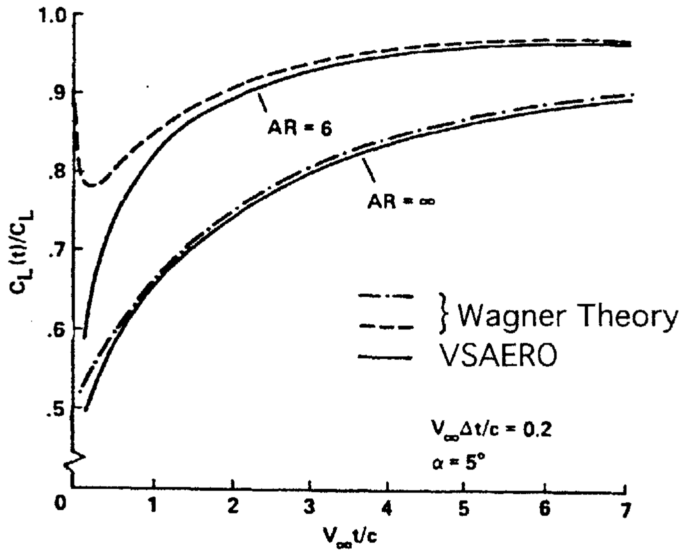


Figure 2-9: Comparison of Theoretical Response Due to Wagner to the Response Predicted Using the VSAERO Panel Code of 2-D and 3-D Wings to Plunging Motions [7]

the late 1980s at the NASA Ames Research Center. The development time and costs for this code were reduced by using an existing code, VSAERO [39], as a foundation. PMARC is the code selected fo use in this research, and it will be discussed in greater detail in Chapter 4.

Other Methods

Other methods associated with unsteady aerodynamics and flexible lifting surfaces that appear in the literature are the two-dimensional strip model based on Theodorsen theory of van Schoor and von Flotow [41], and Laplace domain and rational approximation techniques (also based on Theodorsen theory) presented by Edwards *et al.* [42]. Van Schoor and von Flotow analyze the aeroelastic response of MIT's human-powered aircraft, the *Michelob Light Eagle*, a highly flexible aircraft with some features similar to HALE-type UAVs. Steady and dynamic lift are found as a function of the Theodorsen function, which is calculated analytically by way of Hankel functions with a complex reduced frequency. The lift is assumed to be distributed elliptically across the wing span. Both quasi-steady and fully unsteady aerodynamic formulations were used. This aerodynamic analysis was coupled with a finite-element structural dynamics formulation. Results showed that, in the formulation, many flexible modes need to be included for reliable results and that the fully unsteady aerodynamic model is required.

Edwards, Ashley, and Breakwell used Laplace transform techniques to develop a generalized Theordorsen function for arbitrary airfoil motions in two-dimensional incompressible flows and two-dimensional supersonic flows [42]. Two components of the unsteady response were identified, a rational part and a nonrational part. The nonrational part was found not to contribute to the oscillatory response. This exact aerodynamics model was coupled with structural equations for a flutter study of a two-dimensional thin airfoil with a flap. Approximation methods using augmented states and the Padé approximants were also investigated and were found to be in good agreement with the exact calculations for frequencies near the imaginary axis.

2.2.4 Limitations Towards the Application to Highly Flexible Vehicles

As is evident, considerable effort has been given to the problem of an efficient and capable unsteady aerodynamics model for aeroelastic applications. All the works presented here are significant contributions to the general field; however, for application to the preliminary design phase of a highly flexible vehicle, most of these methods are limited. CFD methods are the most accurate and have the fewest limitations but are, of course, too unwieldy to be of much use early in the design process. The bulk of the methods are limited by their implicit assumption of small deformations that is employed to linearize the problem and hence make it more manageable. The lifting surface is generally, as it is in Drela and Peters' work, assumed to be planar, and aerodynamic singularity solutions (sources, doublets, and vortices) are assumed fixed on the planar spanwise axis. This is valid for problems with small deformations because the distance change between elements is inconsequentially small; however, deformations on the order of 25-30% of the semi-span do result in non-trivial changes to this distance and consequently to the influence coefficients. This will be discussed again in Chapter 5.

Chapter 3

The Aerodynamic Impulse Response Method

As was presented in the previous chapter, there are a variety of ways in which to simplify, or make more manageable, a fluid flow analysis. Two categories of methods were presented, reduced-order models and lower-order models. The method used in this research, the Aerodynamic Impulse Response method, is of the former sort. By itself, it does not model a fluid flow. Rather, it is used in conjunction with an existing aerodynamic model to make the representation more efficient. In fact, the method is a general one. Not limited to aerodynamic problems, it can be applied to any linear or nonlinear system. This chapter presents a brief background of the method and the theory behind it, largely culled from digital signal processing techniques. Both linear and nonlinear formulations are examined, and the chapter closes with a look at the application of the method to aerodynamics and aeroelasticity problems.

3.1 Background

The impulse response method has been in existence for years as, primarily, a digital signal processing tool. The method is a general systems identification tool and hence can be applied to a wide range of problems. There have been limited studies using the method as a structural analysis tool [43, 44]. However, the application of the technique to unsteady aerodynamics problems was only recently proposed by Silva at NASA Langley Research Center [9, 10, 8]. Classically, indicial (step) functions or harmonic functions have been employed to predict unsteady aerodynamic responses (e.g., the Wagner function and the Theodorsen function, respectively) [13]. Silva, however, recognized that neither of these responses is the fundamental aerodynamic response; rather, borrowing from mathematical systems theory and signal processing techniques, the true fundamental aerodyanmic response is the response to an impulse. Indicial or harmonic responses can then be derived from the fundamental response, as he shows in his Ph.D. dissertation [8]. Silva focused his research on CFD applications of the method. His work shows a significant reduction of the computational cost of unsteady aerodynamic analyses by requiring the problem to be solved only once; thereafter, a convolution scheme is employed to derive arbitrary responses from the impulse response. Convolution is much less computationally demanding than the flow solver. He presents results using two codes, CAP-TSD (Computational Aeroelasticity Program–Transonic Small Disturbance), which solves the nonlinear transonic small disturbance potential equation, and CFL3D, a Reynolds-averaged Navier-Stokes solver. He finds, using CAP-TSD, nearly a twenty-fold saving in CPU time required to process ten jobs, with results nearly indistinguishable from those predicted directly by the code.

3.2 Mathematical Foundations

3.2.1 Clasification of Mathematical Systems

There are a variety of ways to classify mathematical systems; three such classifications are referred to in this discussion. These classifications are as follows:

- Time-varying or time-invariant systems
- Discrete-time or continuous-time systems
- Linear or nonlinear systems

First, a system may be time-varying or time-invariant. Note that this is distinct from time-dependence or independence. A time-invariant system may be a function of time; however, a time-invariant system's fundamental properties do not vary with time. Time-invariance is also known as shift-invariance because the response can be shifted arbitrarily in time. These systems are not explicit functions of time; that is, the response to a given input at time t_1 is the same response to the input at a later time t_2. For example, the response in a wind tunnel of an aircraft model to an angle-of-attack change is the same at 10:00 a.m. as it is two hours later. On the other hand, the dynamics of a thrusting rocket is a time-varying system because the mass of the propellant is an explicit function of time. Though the impulse response method is not restricted from time-varying systems, development has been limited to time-invariant systems for which the impulse response is the same at all times.

Mathematical systems can also be designated as either discrete-time systems or continuous-time systems. Mathematically, a continuous-time system is defined as a set of complex-valued functions f defined on the real line; a discrete-time system is a set of sequences of complex numbers. Thus, the time domain of a continuous-time system is the real line $\mathbb{R}$, and the time domain of a discrete-time system is the set of integers $\mathbb{Z}$ [45]. In other words, a signal in continuous-time is a function of time with a value for every real value of time. A signal in discrete-time is a sequence of values of the signal at integer units of time. Real-world systems are continuous-time systems; however, they can only be described digitally as a set of values sampled at discrete time units. No information is available between time units. All numerical analyses operate on discrete-time systems; the impulse response method outlined here is a discrete-time technique. There are small but significant differences between the mathematical theory of continuous-time systems and that of discrete-time systems. The impulse function is one such distinction. In continuous-time the definition of the impulse is somewhat ambiguous, and hence it is difficult to work with impulse functions; however, in discrete-time the impulse function is well defined. Namely,

$$u[t] = \begin{cases} 1.0 & \text{for } t = t_0 \\ 0.0 & \text{for } t \neq t_0 \end{cases} \tag{3.1}$$

This function is properly known as a unit sample function or the Dirac delta function; note, however, that "impulse" is used throughout this thesis analogously.

The final classification that will be referred to is whether the system is linear or nonlinear. A system $F(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}, \ldots) = 0$ is linear if the coefficients of all its derivatives are functions of the independent variables only and it is linear in the unknown function u. A system that cannot be described this way is considered a nonlinear system. A simple example of a nonlinear system common in fluid mechanics is the Burgers equation,

$$u_t + uu_{xx} = \nu u_{xx} \tag{3.2}$$

which is a simple one-dimensional model for diffusive waves and is often used as a test case for CFD models. The key difference between linear and nonlinear systems is that superposition of solutions is valid for linear systems and is not valid for nonlinear systems. The Aerodynamic Impulse Response method is valid for both linear and nonlinear systems. Convolution, which is based on superposition, is the foundation of the linear Aerodynamic Impulse Response method; to use the method with nonlinear aerodynamic systems the nonlinear analog of convolution, the Volterra series, must be employed.

3.2.2 The Linear Impulse Response Method

As Silva indicates, the linear impulse response method has been well developed in the field of digital signal processing [8, 46, 47]. In essence, the method states that the response of a linear system to an arbitrary input function can be derived from the response of the same system to a unit sample function by using convolution.

Convolution in discrete-time form is defined as follows:

$$y[n] = \sum_{k=0}^{N} h[n-k]x[k] \tag{3.3}$$

where y is the response to the arbitrary input, h is the impulse response, x is the arbitrary input, and $n, k \in \{0, 1, \ldots, N\}$ are time units. Thus the response of a linear system to an arbitrary input is the superposition of scaled and shifted impulse responses as is shown in Figure 3-1. Because the response to any input can be constructed from the impulse response, the impulse response is the fundamental response of the system. It is also known as a memory function because it indicates how long a perturbation affects the system; a system with an impulse response that dies quickly to zero has little memory. That the impulse response dies to zero is one of the benefits of the method—impulse responses, while just as accurate, require less computational time than the identification of indicial-type responses [8].

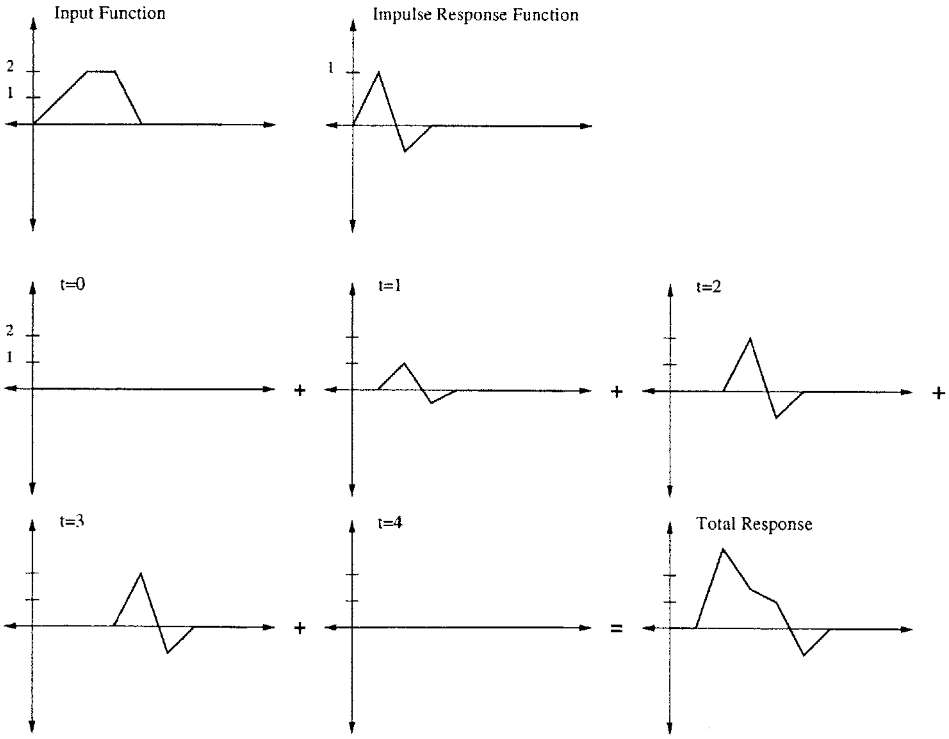


Figure 3-1: Simple Example of the Discrete-Time Convolution Method

Figure 3-2 describes the method diagrammatically. Whatever flow solver is being used makes up the core of the system identification component of the process. This code is used to determine the response of the system—an aerodynamic body—to a unit impulse. Once the impulse response is acquired, the flow solver is no longer needed. Subsequent responses to arbitrary inputs are calculated using the impulse response and a convolution code.

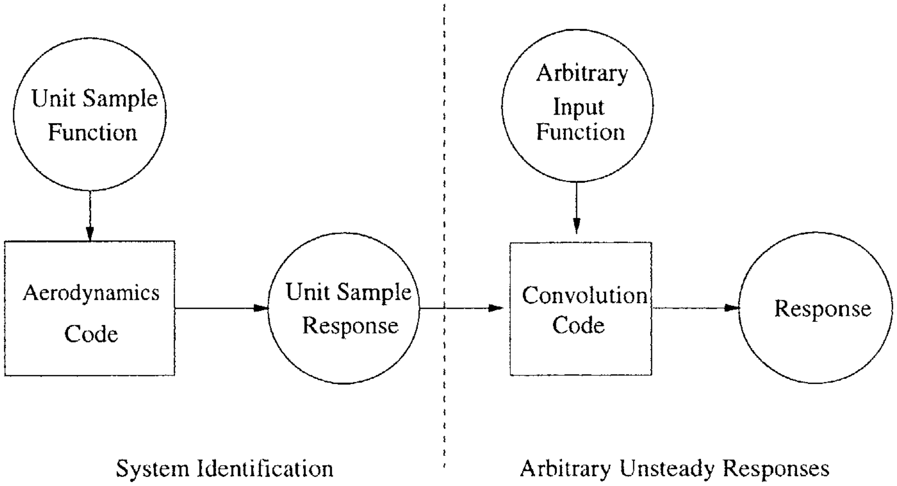


Figure 3-2: Block Diagram of the Linear Aerodynamic Impulse Response Method

3.2.3 The Nonlinear Impulse Response Method

Not surprisingly, the analysis of nonlinear systems with the impulse response method is more complicated than for linear systems; however, the method is still applicable; nonlinear applicability is, in fact, one of the primary benefits of the method. Instead of linear convolution, its nonlinear analog, a result of the Volterra-Wiener theory of nonlinear systems, must be used. The discrete-time Volterra series is defined as follows [8]:

$$y[n] = h_0 + \sum_{k=0}^{N} h_1[n-k]x[k] + \sum_{k_1=0}^{N}\sum_{k_2=0}^{N} h_2[n-k_1, n-k_2]x[k_1]x[k_2] + \cdots \qquad (3.4)$$

In this case, h_i is called the i-th order kernel of the impulse response. The zeroth-order kernel, h_0, accounts for some initial offset and is generally zero. The first-order kernel,

h_1, measures the amplitude dependence in the system; if the system is purely linear, the first-order kernel is equivalent to the linear impulse response and the second-order and higher kernels will be zero. Second- and higher-order kernels give a measure of the interaction of multiple excitations at various times. This formulation has the added benefit of providing a clear quantification of the degree of nonlinearity in the system; the closer to zero the higher-order kernels are, the more linear the system is.

Higher-order kernels are found from the response of the system to multiple unit impulses, where the number of impulses applied is equivalent to the order of the kernel. Thus three impulses are applied at varying times t_1, t_2, and t_3 for the third-order kernel. The nonlinear kernels thus describe the interaction of multiple impulses. Silva details the method of determining the higher-order kernels in his dissertation [8]. This kernel identification process consists of varying the time at which each impulse is applied until there is no interaction between them. All kernels are symmetric, and the i-th order kernel is an i-dimensional function of time. Thus, the second order-kernel can be represented as a matrix. An example of a second-order kernel is given in Figure 3-3 [8]. Because they require more than three-dimensions, higher-order kernels are more difficult to visualize.

To identify nonlinear kernels, several responses must be acquired from the system. The number of responses needed depends on the number of kernels desired and the memory of the system. Here, only second-order kernels are considered. The responses needed to assemble the first- and second-order kernels include the following:

- Response to a single unit impulse, $y_1[t]$
- Response to a single double impulse, $y_2[t]$
- Responses to impulses at t_0 and $t_0 + T$, $y_{11}[t, T]$

Once these responses are acquired, the kernels can be assembled. The first-order kernel describes the amplitude dependence of the system, comparing the response to a unit impulse to the response of a double-amplitude impulse. Mathematically, this is

$$h_1[t] = 2y_1[t] - \frac{1}{2}y_2[t] \tag{3.5}$$

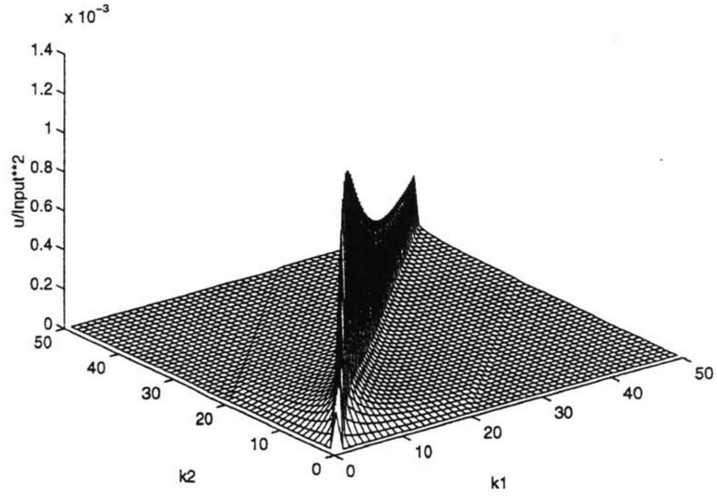


Figure 3-3: Three-Dimensional View of the 2nd-Order Kernel of the Burgers Equation [8]

If the system is linear, then doubling the amplitude of the impulse will give the same response as doubling the response of a unit impulse, thus the linear impulse response is recovered. Nonlinear kernels describe the difference of the response to two impulses at different times and two superimposed and appropriately time-shifted impulse responses. This difference is given as follows:

$$h_2[t,T] = \frac{1}{2}\left(y_{11}[t,T] - y_1[t] - y_1[t+T]\right) \tag{3.6}$$

The response at each impulse interval T is called a *component* of the kernel. The first component is for $T = 0$, or two impulses occurring simultaneously—which is equivalent to a single double impulse. For a system with short memory, only the first several components will be nonzero. Once these componets have been obtained, they can be assembled in matrix form.

Figure 3-4 depicts the method in block diagram form. As with the linear form, the analysis is divided into two processes. First, the system identification process uses the flow solver to acquire the impulse response kernels of the system. For moderately nonlinear systems, it is generally sufficient to truncate the series after the second-order kernel. The kernel identification process is done iteratively, varying the time between the impulses until there is no interaction—the kernel goes to zero. Once the impulse response kernels are obtained, the flow solver is no longer needed. Subsequent responses to arbitrary inputs are determined with the impulse response kernels and the Volterra series.

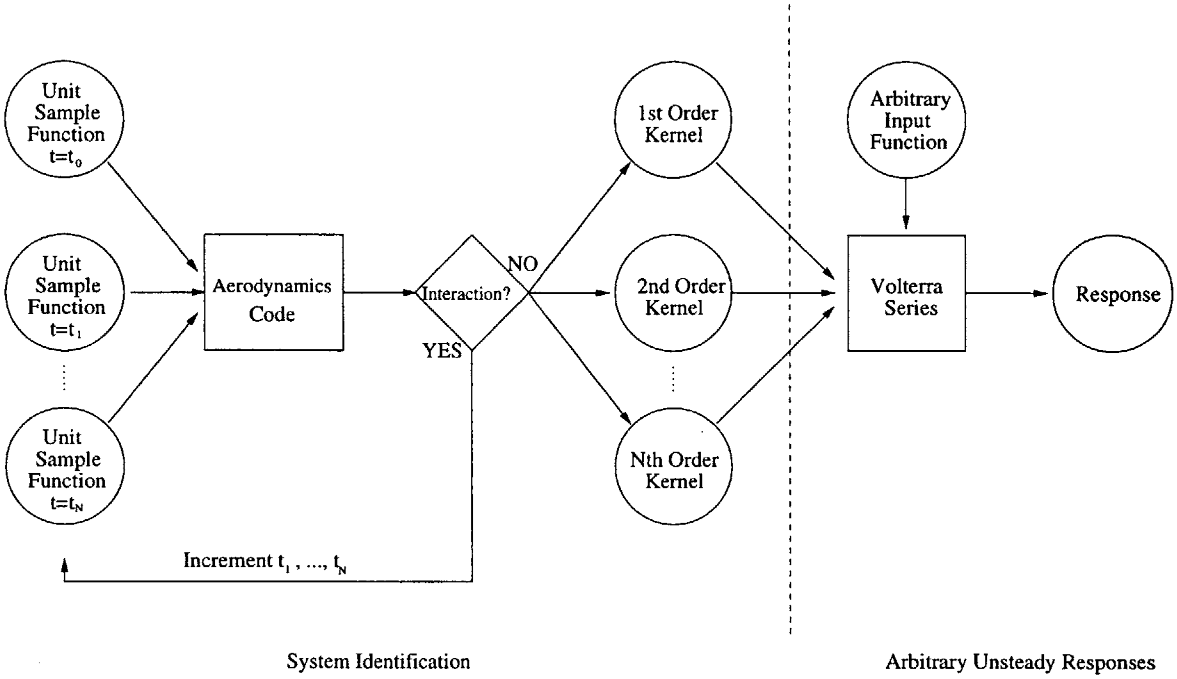


Figure 3-4: Block Diagram of the Nonlinear Aerodynamic Impulse Response Method

Silva examines the Riccati equation and the Burgers equation using the nonlinear impulse response techniques outlined here. In both cases a very good match was found between the response predicted with a combination of first- and second-order kernels and the actual response. An amplitude dependence is, however, apparent. At higher input amplitudes there is a noticable degradation of the fit. This can only be accounted for by using a higher-amplitude impulse to obtain the impulse response kernels.

Besides the amplitude limitations inherent to the nonlinear formulation, it should also be noted that a set of impulse response kernels is strictly valid only at the freestream conditions, e.g., M_∞, for which it was identified. Thus, a single set of impulse response kernels fully describes the unsteady aerodynamic behavior of a given configuration at, say, $M_\infty = 0.8$. To study the response of the same configuration at $M_\infty = 0.6$, however, a new set of impulse response kernels must be identified.

3.3 The Aerodynamic Impulse Response Method

As described in the preceeding sections, the impulse response method is applicable to any mathematical system. Application to particular systems—in this case, aerodynamic systems—is accomplished by way of the boundary conditions of the system. Every aerodynamics code is unique in its treatment of boundary conditions, hence there is no generally applicable method of implementation. Typically, however, some form of what Silva terms the "downwash equation" [8] appears. This equation separates the spatial and temporal components of the body's motion and generally can be expressed in the form

$$w(x, y, z) = \Phi'(x, y)u(t) + \Phi(x, y)\dot{u}(t) \tag{3.7}$$

where $()'$ indicates spatial differentiation and $\dot{()}$ indicates differentiation with respect to time. Additional terms to account for wing thickness or other effects may appear, but only the temporal functions are important.

It is important to notice from Equation 3.7 that there are actually two separate "input channels," u and $\dot{u}$. Consideration of the two channels as a single input would lead to an undefined discrete time-derivative of the impulse, because as $\Delta t \to 0$, $\dot{u}[t_0] \to \infty$. Instead, the two channels must be considered separately. An impulse response is obtained for

$$u[t] = \begin{cases} 1.0 & \text{for } t = t_0 \\ 0.0 & \text{for } t \neq t_0 \end{cases}$$

with $\dot{u}[t] = 0.0$ for all t. Then, a separate impulse response is obtained with

$$\dot{u}[t] = \begin{cases} 1.0 & \text{for } t = t_0 \\ 0.0 & \text{for } t \neq t_0 \end{cases}$$

and $u[t] = 0.0$ for all t. For linear systems the total response is simply the sum of the response of each channel. For nonlinear cases, the method is more complicated, involving consideration of the interaction of the two channels, and is described in more detail in [8].

Depending on the formulation of the aerodynamic model, there may be fewer or more input channels. As will be seen in Chapter 5, there are three input channels in PMARC: pitch, α; pitch rate, $\dot{\alpha}$; and plunge rate, v_z.

As the Aerodynamic Impulse Response formulation stands, it is applicable to rigid-body analyses only; that is, wings or full aircraft configurations undergoing rigid-body pitching and plunging motions. For application to aeroelastic analyses, the response of the system to changes in geometry needs to be determined. Aeroelastic models commonly couple the unsteady aerodynamics model with the structural dynamics model iteratively—the structural dynamics model passes the deflection of the structure due to aerodynamic loading to the aerodynamics model, which then recomputes the aerodynamic loading. This process was illustrated in Figure 2-2. Because it would be impractical to find the impulse response of every degree-of-freedom of the system, a better approach is to use some form of modal analysis, either linear or nonlinear, to find the impulse response of a flexible structure. Because typically only several modes are required to describe the most significant characteristics of a system, the aerodynamics code will only need to be run several times to identify the impulse responses of these modes. Thenceforth only convolution is required to determine arbitrary responses of the flexible structure.

The structural mode shapes describing the deformed state of the structure can be found with a structural analysis model using only the homogeneous form of the aeroelastic governing equation, Equation 2.1. These structural modes can then be written in terms of the spanwise kinematic velocity. As noted earlier, it is generally sufficient to retain only several mode shapes; an impulse response due to each mode shape can be calculated and applied to find the response of the flexible-body system to arbitrary motions. Figure 3-5 illustates this method. The method is discussed in greater detail in Chapter 5.

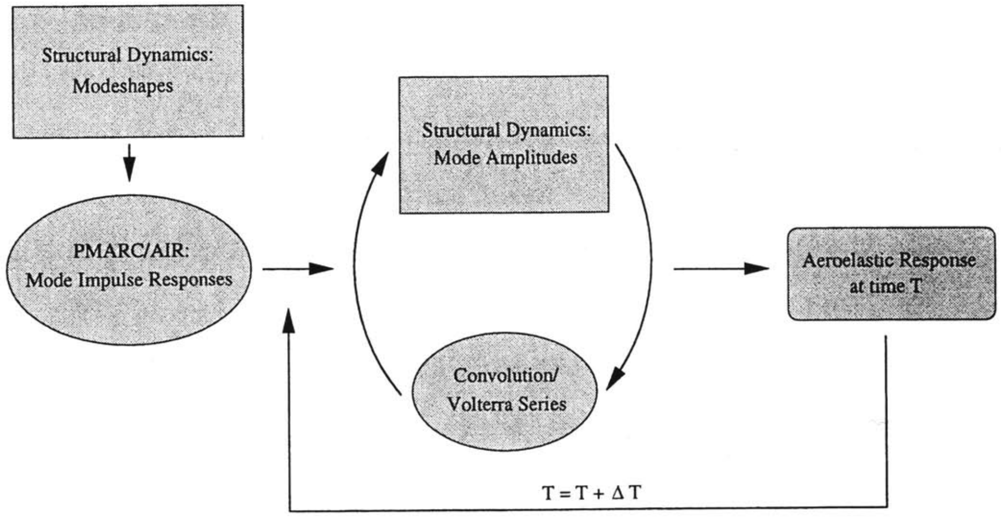


Figure 3-5: Schematic of the Aeroelastic System Using Structural Modal Analysis and the Aerodynamic Impulse Response Method

Chapter 4

An Overview of Aerodynamic Panel Methods

The Aerodynamic Impulse Response method is applicable to any aerodynamic method–either CFD or any of the simplified models mentioned in Chapter 2. In this research, the aerodynamnic platform employed is a low-order aerodynamic panel code similar to that employed by Katz and Maskew [7]. This chapter provides a brief overview of panel methods in general and of the particular panel code chosen for this work, the NASA Ames code PMARC. The chapter begins with a brief history of the development of panel codes and then continues with an overview of the theory of panel methods. Next, the specific benefits of the panel method are detailed, and, finally, PMARC and some specifics of its methodology are presented. References [48] and [49] are invaluable references on the panel method and are referred to throughout.

4.1 History

The aerodynamic panel method is based on the linear potential flow model governed by Laplace's equation. The method was first formulated for two-dimensional lifting and non-lifting airfoils and not long afterwards the technique was expanded to three-dimensional bodies. The earliest work appeared in Germany before the widespread availability of digital computers. The first successful three-dimensional panel code appeared in the early 1960s, a development of the Douglas Aircraft Company called Hess [50]. This method used constant sources only with a Neumann boundary condition and hence solved for only non-lifting three-dimensional flows. Development continued through the 1960s, and by the early 1970s the USSAERO code [51] was available featuring linear source and vortex distributions for sub- and supersonic lifting flows. Douglas's Hess code was improved at the same time to include constant doublets for the solution of lifting flows.

Three paths of innovation appeared at this point, marking the so-called "second generation" of codes. The first was the development of the Dirichlet boundary condition, which eliminated the need for non-physical "lift-carry-over" surfaces at the intersection of wing-type surfaces and body-type surfaces [49]. The second innovation was the development of high-order singularity distributions, primarily quadratic. This provided a more accurate solution at the expense of increased code complexity and more stringent discretization requirements. (Low order methods are substantially less sensitive to gaps in panelling.) Finally, codes capable of simple unsteady analysis appeared. Representative codes of this generation include MCAIR [52], SOUSSA [53], PAN AIR [54, 55], and an updated Hess code often referred to as Hess II [56].

The third generation of panel codes appeared concurrently with the microcomputer. Previous codes required mainframe computers and hence were only practical for large aerospace companies, but the widespread availability of microcomputers in the 1980s allowed for the explosive growth of the market for panel codes. Interest in higher-order methods waned, and a flurry of new techniques appeared including integrated boundary layer effects using sources to adjust the effective body shape (transpiration) and numerical optimization of the solver routines, and improved wake models allowing for roll-up were introduced [48]. VSAERO [39] was the first panel method that was widely available commercially, and this was followed by a Lockheed development, QUADPAN [57], and the NASA Ames code PMARC [40].

4.2 Theory

The foundation of panel methods is potential flow theory, which can be derived from the fluid flow continuity equation. If density is assumed to remain constant, the incompressible form of the conservation of mass equation gives

$$\nabla \cdot \mathbf{V} = 0 \tag{4.1}$$

where $\mathbf{V} = u\hat{\imath} + v\hat{\jmath} + w\hat{k}$ is the velocity vector. Further, if the flow is assumed to be irrotational, a velocity potential ϕ can be defined such that

$$\mathbf{V} = \nabla \phi \tag{4.2}$$

Combining Equations 4.1 and 4.2 gives Laplace's equation,

$$\nabla^2 \phi = 0 \tag{4.3}$$

reducing the set of three equations to a single equation. This equation has been exhaustively studied and its solutions are well understood [33, 48, 58]. This is the governing equation for incompressible potential flows.

For a potential flow analysis about a body and wake as shown in Figure 4-1, there are four boundary conditions that must be satisfied. On the body surface S_b, the non-penetration, or flow tangency, condition is applied. In the body frame-of-reference, this is commonly given for steady flows as

$$(\mathbf{V}_\infty + \nabla \varphi) \cdot \mathbf{n} = v_n \tag{4.4}$$

where v_n is a prescribed normal velocity, usually zero, and φ is a small perturbation potential, so that the local velocity is $\mathbf{v} = \mathbf{V}_\infty + \nabla \varphi$. For unsteady flows, the same boundary condition can be written as

$$(\mathbf{V}_\infty + \mathbf{V}_{plunge} + \mathbf{\Omega} \times \mathbf{R} + \nabla \varphi) \cdot \mathbf{n} = v_n \tag{4.5}$$

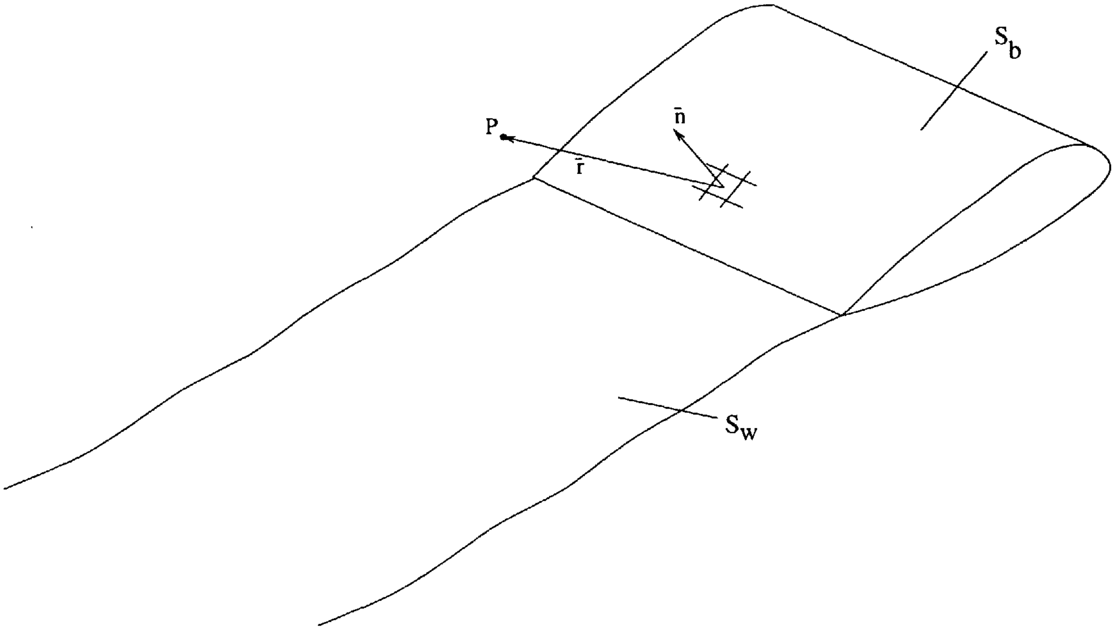


Figure 4-1: Body and Wake for Potential Flow Analysis with a Panel Method

where $\mathbf{V}_{trans}$ is the body translation velocity, $\boldsymbol{\Omega}$ is the rotation rate vector (comprised of pitch, roll, and yaw), and $\mathbf{R}$ expresses the distance from the element to the rotational center. This condition can be satisfied with either a Neumann condition or, as is generally the case in recent codes, a Dirichlet condition, which sets the potential inside the body to zero. The second boundary condition is that the static pressure must be continuous through the wake surface S_w, so

$$\Delta C_p = C_p(S_w^+) - C_p(S_w^-) = 0 \tag{4.6}$$

Thirdly, at the trailing edge, the flow must leave the body smoothly (the Kutta condition), and, finally, the perturbation vanishes at infinity upstream.

The governing equation and the boundary conditions are satisfied with a distribution of two types of elementary solutions: the source, with strength σ; and the doublet, with strength μ. In incompressible flows, sources produce a jump in velocity in the normal direction and doublets cause a velocity jump in the tangential direction.

Thus the body, S_b, is described with a distribution of sources and doublets, while the wake, S_w, which has a velocity jump only in the tangential direction, is described by a distribution of doublets only. Though the wake is physically a vortex system, it can be modeled with doublet elements because, as Lamb shows, the induced velocity from a doublet of constant strength is equivalent to that from a vortex ring with the same strength [58].

The potential at any point P in the flowfield due to the source distribution is

$$\phi_\sigma = -\frac{1}{4\pi} \iint\limits_{S_b} \sigma(\mathbf{x}) \frac{dS}{|\mathbf{r}|} \tag{4.7}$$

where $\mathbf{r}$ is the distance from point P to the element dS. The potential at point P due to the doublet distribution is

$$\phi_\mu = -\frac{1}{4\pi} \iint\limits_{S_b+S_w} \mu(\mathbf{x}) \frac{\hat{\mathbf{n}} \cdot \mathbf{r}}{|\mathbf{r}|^3} dS \tag{4.8}$$

where $\mathbf{n}$ is the unit normal vector from element dS.

Since the normal velocity at the surface is known a priori, the source strengths can immediately be determined. If no flow through the surface is assumed, the source strengths are given by

$$\sigma = -\frac{1}{4\pi}\, \hat{\mathbf{n}} \cdot \mathbf{V}_\infty \tag{4.9}$$

Distributing these sources and doublets on the discretized body and wake surfaces, simplifying with Equation 4.9, and applying the boundary conditions, some manipulation yields an integral equation for the unknown doublet strengths results:

$$\left[\iint\limits_{S_b-P} \mu\, \hat{\mathbf{n}} \cdot \nabla \left(\frac{1}{\mathbf{r}}\right) dS - 2\pi\mu_P \right] + \iint\limits_{S_b} \left(\frac{\sigma}{\mathbf{r}}\right) dS + \iint\limits_{S_w} \mu\, \hat{\mathbf{n}} \cdot \nabla \left(\frac{1}{\mathbf{r}}\right) dS = 0 \tag{4.10}$$

P in this case is an element dS on the body or wake, and $S_b - P$ indicates integration over all elements except P (because at P the integral is singular). μ_P is the, as yet unknown, doublet strength at panel P.

Finally, this equation is written in discretized form. The body and wake are described with a set of small rectangular elements called panels. Low-order panel codes, as will be assumed here, take advantage of the constant singularity strength on each panel to extract the source and doublet strengths from the integral, leaving a matrix equation for the doublet strengths at time t,

$$\sum_{k=1}^{N_s} \mu_k C_{jk} + \sum_{k=1}^{N_s} \sigma_k B_{jk} + \sum_{l=1}^{N_w} \mu_{w_l} C_{jl} = 0 \Big|_{j=1,N_s} \tag{4.11}$$

where N_s is the number of body panels, N_w is the number of wake panels, and the influence coefficient matrices B_{jk} and C_{jk} are defined as follows:

$$B_{jk} = \iint_k \frac{1}{\mathbf{r}} dS \tag{4.12}$$

and

$$C_{jk} = \begin{cases} -2\pi & j = k \\ \iint_k \hat{\mathbf{n}} \cdot \nabla(\frac{1}{\mathbf{r}}) dS & j \neq k \end{cases} \tag{4.13}$$

These coefficient matrices dictate the influence on panel j of a singularity of unit strength at panel k. For rigid bodies, as is assumed in the basic panel code formulation, Equations 4.12 and 4.13 are functions of geometry only and thus need only be evaluated once. (For flexible bodies with small deformations, it can be assumed that the influence coefficient matrix is constant. Highly flexible structures require the coefficient matrix to be updated, however.) Source strengths are known, as given in Equation 4.9, and can be moved to the right hand side of Equation 4.11.

Doublet values in the wake, μ_{w_l} are calculated as a function of body panel doublet strength values. A variety of wake models are available, the simplest being a prescribed rigid wake. Most modern panel codes employ some form of wake timestepping so that the wake grows with the time iteration—a row of wake panels is shed from the trailing edge at each time step and the previous wake panels are convected downstream. Wake relaxation is also common in modern codes. Using this technique, the position of each wake panel is recomputed at each time step based on the local velocity rather than the freestream velocity. This allows the wake to roll up at the edges as is physically observed. It is important to note that if a prescribed rigid wake is used, the problem is entirely linear; wake relaxation, on the other hand, introduces some nonlinearity to the problem.

The Kutta condition fixes the doublet strength values of the first row of wake panels. The Kutta condition simply states that the velocity at the trailing edge must be finite. The trailing edge is defined as the common edge between panels on the lower and upper surfaces of a lifting surface. As indicated earlier, a doublet is equivalent to a vortex ring with the same strength. As shown in Figure 4-2, at the trailing edge, one edge of the two vortex rings coincides and, since they flow in opposite directions, the resulting vortex strength is the difference between the two. The first wake panel shares this edge as well, and hence its doublet strength is equal to the difference of the doublet strengths of the two body panels.

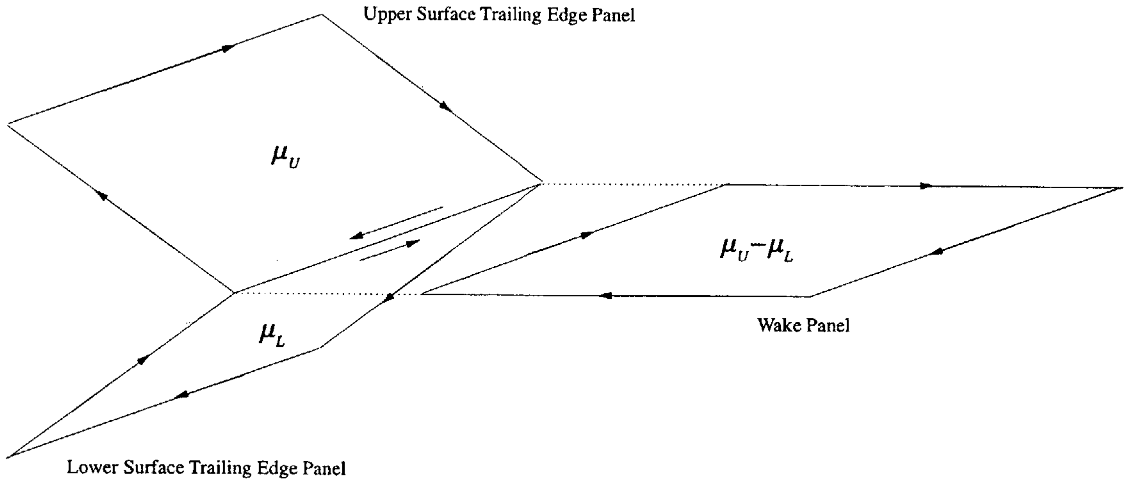

Upper Surface Trailing Edge Panel
μ_U
$\mu_U - \mu_L$
Wake Panel
μ_L
Lower Surface Trailing Edge Panel

Figure 4-2: Determination of Doublet Strength Shed into the Wake

By prescribing an initial wake, a steady state solution beginning at some time $t > 0$ can be achieved. On the initial wake, all rows of wake panels are given the same doublet strength as the first row. If no initial wake is given, an impulsive start at $t = 0$

is simulated, and the wake at the first time step consists only of a single row of wake panels. At subsequent time steps, a new row of panels is added at the trailing edge and the remaining panels are convected downstream, retaining the doublet strength of the previous time step. Hence, the strength of the first row is written in terms of the body doublet strengths and the remaining wake panel strengths are known and can be moved to the right hand side of Equation 4.11. The system is solved using an iterative solver for a system of linear equations.

Once the doublet distribution is found, the potential at any point in the flow can easily be calculated. For aeroelastic applications, only surface forces are of interest. Velocity components at each panel can easily be reduced from the velocity potential, and the pressure at each panel can be found with the unsteady form of Bernoulli's equation,

$$C_{p_k} = 1 - \frac{V_k^2}{V_\infty^2} + \frac{d\phi}{dt} = 1 - \frac{V_k^2}{V_\infty^2} + 2\left(\frac{4\pi}{V_\infty^2}\right)\left(\frac{\mu_k^n - \mu_k^{n-1}}{\Delta t}\right) \tag{4.14}$$

where C_{p_k} is the nondimensional pressure coefficient at panel k, V_k is the local velocity at panel k, and n is the current time step. Forces and moments are calculated by integrating the pressure over the surface.

4.3 Benefits of the Panel Method

Engineering analysis is always a matter of compromise, and this case is no different. Flow simulation via the Navier-Stokes equations would obviously provide the most accurate solution but are prohibitively expensive both in the preparation phase and in the execution phase. For research of fundamental phenomena and for detailed analyses, this cost may be acceptable; however, in the case of preliminary design where multiple design iterations are ordinarily necessary, CFD is still not a realistic tool. The simplified aerodynamic models mentioned in Chapter 2 are more appropriate. Panel methods are a popular choice and were selected for this research for a number of reasons.

Panel codes are fast for reasonably sized problems. Though not very efficient—the cost is proportional to N^2 or N^3, where N is the number of panels [49]—the problem size is orders of magnitiude smaller than CFD control-volume approaches. In three-dimensional panel methods, only the body surface and the wake need to be discretized, and the elements (panels) are two-dimensional. Three-dimensional CFD methods, on the contrary, require discretization of the entire control volume with three-dimensional elements. Inviscid solutions with panel methods require on the order of one thousand panels; solutions coupled to boundary-layer methods typically require several thousand panels. CFD problems typically involve hundreds of thousands or millions of elements. Because of the reasonable problem size, panel methods can be run on a wide variety of platforms, while CFD methods often require high-end workstations or supercomputers.

Another advantage of panel methods is their applicability to general configurations—arbitrary three-dimensional bodies. Lifting-line based methods like the lumped vortex model employed by Drela [5] are limited to two-dimensional sections extended to three-dimensional planar lifting surfaces by way of strip theory. Panel methods, on the other hand, apply to arbitrary three-dimensional bodies, meaning that a highly deflected nonlinear equilibrium position of a wing is an acceptable geometry; with the appropriate modifications, large geometrically nonlinear deformations may be considered.

Finally, panel methods, when employed within their range of applicablity, provide reasonably accurate results [7, 40, 48, 49]—results accurate enough for preliminary design decisions. The region of applicablity of panel methods has the following bounds:

- High Reynolds number flow (inviscid)
- No flow separation
- Vorticity confined to thin layers (i.e., wakes)
- Incompressible flow or small-perturbation compressible flow with no shocks

The linearity of the governing equation restricts the method from prediction flow-related nonlinearities such a shocks, separation, and stall phenomena; however, geometrical and wake nonlinearities are accounted for.

4.4 PMARC

The panel code selected to use with this research is PMARC [40]. This code is a low-order (constant strength) source/doublet method that was developed in the late 1980s at the NASA Ames Research Center. PMARC is one of the most capable modern panel codes, including such features as

- Time-stepping, relaxed nonlinear wake
- Internal flow modeling
- Jet plume modeling
- Simple harmonic pitch and plunge
- Boundary layer modeling

Additionally, the problem size is limited only by hardware; PMARC can readily be adjusted for any number of maximum panels. The source code for versions prior to the latest is readily available and is written in standard FORTRAN 77, hence can be compiled on any platform supporting ANSI FORTRAN. Handy modeling features are included, such as an automatic NACA 4-digit series wing generator.

The unsteady capabilities of PMARC are rudimentary, limited to single-frequency sinusoidal pitching and plunging oscillations. Martin and Kroo compared unsteady results using PMARC with theoretical predictions using Theodorsen theory for two-dimensional airfoils (high aspect ratio wings) [59]. Their findings suggest that PMARC does effectively model unsteady motions. It will thus be effective both as a platform for the impulse response and to provide control data—predictions using the impulse response method in conjuction with PMARC will be compared to predictions for sinusoidal oscillations using PMARC alone.

The methodology followed in the PMARC code is similar to that outlined in the theory section above. Figure 4-3 is a flowchart that traces the operation of the code.

Program input is achieved through a data file that specifies the geometry of the body and wake, if specified, as well as program control options. Upon reading the job information from the data files, PMARC assembles the geometry and locates the corner and control points of all panels. Once the geometry is complete, the source and doublet influence coefficient matrices are constructed—as mentioned earlier, since PMARC assumes only rigid body motion, this needs only to be done once. Initial wake data is read in, and then the time-stepping loop commences. Time step size and total number of time steps are specified in the data file.

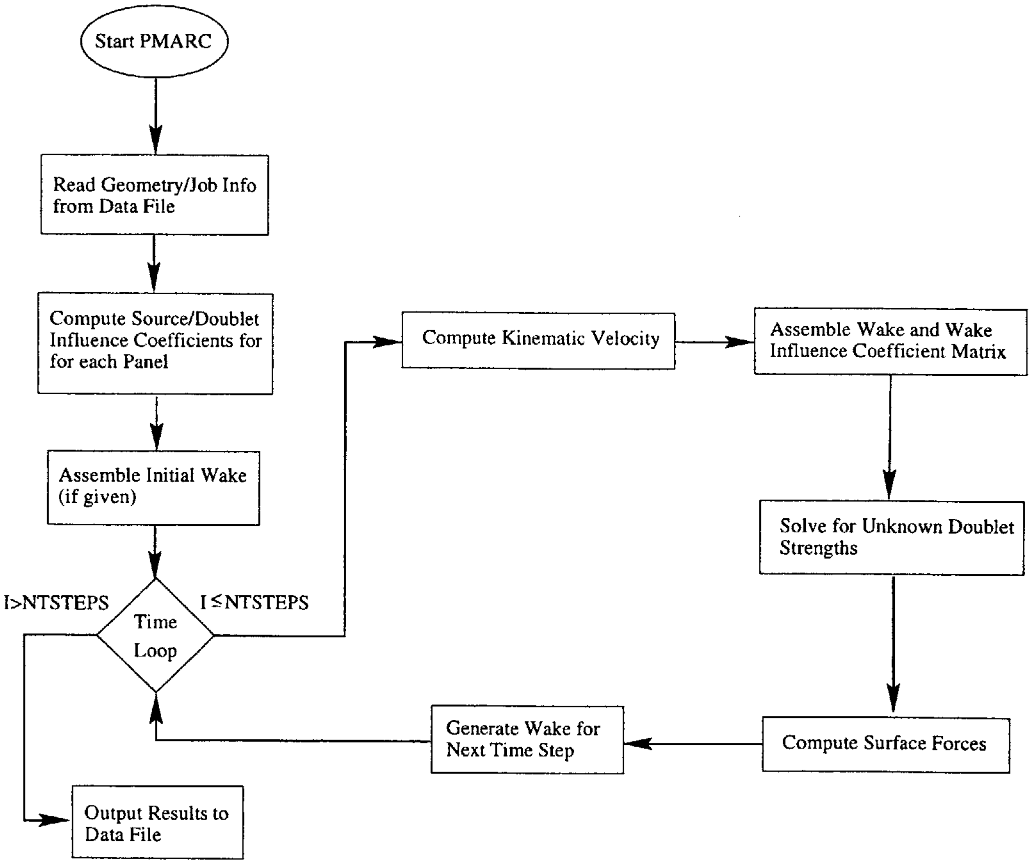

Start PMARC
Read Geometry/Job Info from Data File
Compute Source/Doublet Influence Coefficients for for each Panel
Assemble Initial Wake (if given)
I>NTSTEPS
I≤NTSTEPS
Time Loop
Compute Kinematic Velocity
Assemble Wake and Wake Influence Coefficient Matrix
Solve for Unknown Doublet Strengths
Compute Surface Forces
Generate Wake for Next Time Step
Output Results to Data File

Figure 4-3: Schematic of the Operation of PMARC

The time-stepping loop begins by computing a kinematic velocity, that is, the flow velocity due to the freestream and the motion of the body. In the case of steady flow, this is simply the freestream velocity rotated by any static pitch and yaw angles given in the input file; it remains constant throughout the time-stepping loop. PMARC also is capable of sinusoidal pitch and plunge oscillations; if this unsteady capability is employed, a new kinematic velocity is calculated at each time step. The wake for the current time step is assembled, and a wake influence coefficient matrix is constructed. The system of linear equations is then assembled and solved for the unknown doublet strengths using an iterative scheme. A convergence parameter is specified in the input file to control the iteration. Given the doublet distribution, surface velocities and forces are computed and, if specified, written to an output file. Wake information for the next time step is generated, and the loop repeats until the final time step is reached, at which point the final results are written to the output file and the program ends.

Chapter 5

PMARC Modifications and Other Coding

As indicated in Chapter 3, the Aerodynamic Impulse Response method is not a flow solver itself; rather, it "sits on top" of a traditional flow solver to reduce the computational effort required for repeated unsteady analyses. Any unsteady flow solver is applicable; this research investigates the impulse response method in conjunction with an aerodynamic panel code, as discussed in Chapter 4. Specifically, PMARC Version 12.20 was selected. This chapter outlines the modifications to the source code required for the implementation of the Aerodynamic Impulse Response method for rigid-body responses, followed by the additional changes necessary to obtain flexible-body responses. The resulting code, PMARC with the addition of impulse response capabilities, will be referred to as PMARC/AIR. Implementations using the linear form of the impulse response method will be referred to as Linear PMARC/AIR, while those employing nonlinear impulse response techniques will be referred to as Nonlinear PMARC/AIR. Finally, additional codes such as the convolution program used to generate the response to arbitrary inputs with an impulse response from PMARC/AIR are reviewed. Refer to the Appendix for a full source code listing of programs developed in this research and PMARC subroutines that have been significantly altered.

5.1 Linear PMARC/AIR for Rigid Bodies

Implementation of the linear Aerodynamic Impulse Response method for a rigid-body analysis (pitch and plunge of wings or bodies) is fairly straightforward. Examination of the PMARC source code reveals that the boundary condition that should be perturbed is the unsteady non-penentration condition (as given by Equation 4.5). The freestream flow plus velocities induced by pitch and plunge is referred to as the kinematic velocity, and a single subroutine is responsible for the calculation of the kinematic velocity at the current time step, PATH. The flowchart in Figure 5-1 indicates where this subroutine fits within the entire framework of PMARC. This subroutine in its modified form can be found in Appendix A.1.

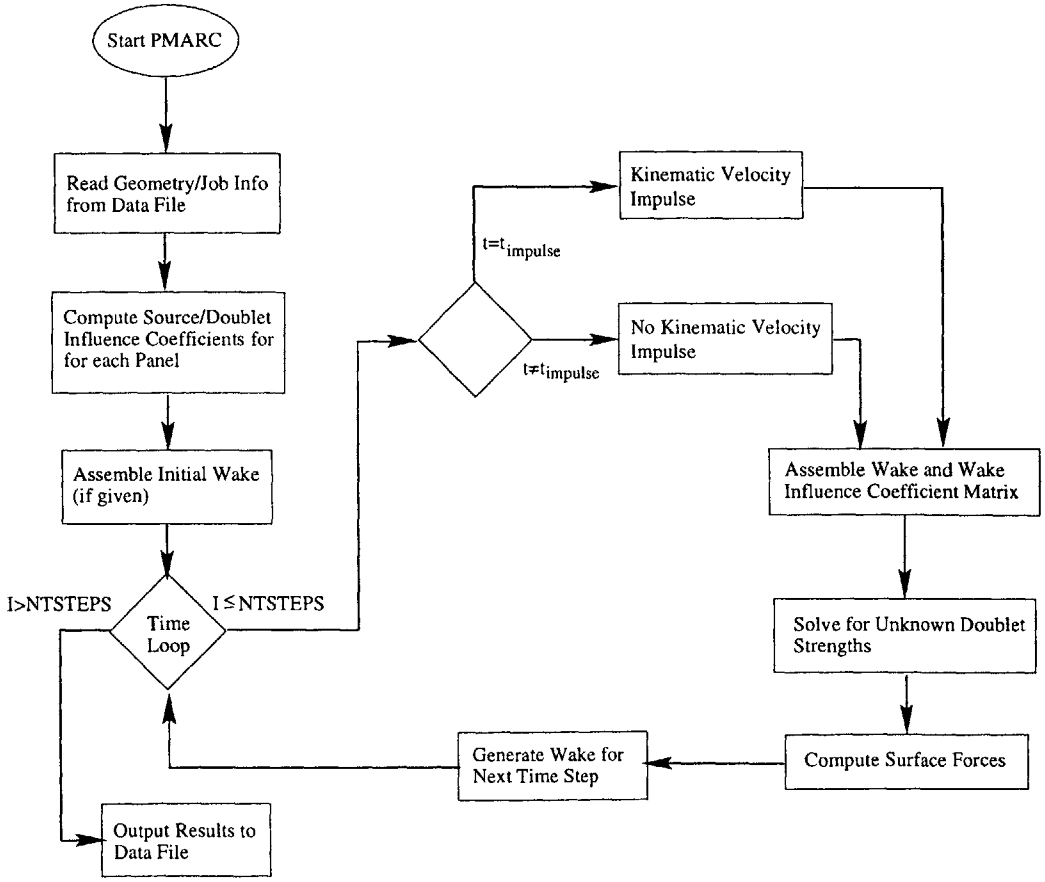

Start PMARC
Read Geometry/Job Info from Data File
Compute Source/Doublet Influence Coefficients for for each Panel
Assemble Initial Wake (if given)
I>NTSTEPS
Time Loop
I≤NTSTEPS
Output Results to Data File
t=t_impulse
t≠t_impulse
Kinematic Velocity Impulse
No Kinematic Velocity Impulse
Assemble Wake and Wake Influence Coefficient Matrix
Solve for Unknown Doublet Strengths
Compute Surface Forces
Generate Wake for Next Time Step

Figure 5-1: Flowchart Outlining the Rigid-Body PMARC/AIR Methodolgy

Because Version 12 of PMARC uses a frame-of-reference fixed on the body, unsteady motions are achieved by adjusting the freestream flow. Because of this, besides implementing the impulse on the body, the impulse affects the wake as well. The wake is updated at each time step in the WAKSTEP subroutine. Modifications to this subroutine were similar to those carried out in the PATH subroutine, except that the impulse actually appears in two time steps because a forward difference derivative is used. However, subsequent testing with both the linear and nonlinear PMARC/AIR implementations showed that carrying the impulse over onto the wake has a nearly negligible effect on the response. Impulse responses are nearly identical as when only the body is impulsed.

In the current implementation of PMARC/AIR for rigid-body responses, an impulse is triggered by a negative integer value for the frequency of oscillation about the z-axis, WTZ. The absolute value of this "frequency" is the time step at which the impulse is applied. This allows the impulse to be applied after the wake has developed over several time steps. However, experimentation shows that impulses applied at all time steps greater than zero produce an identical response. Thus, throughout this research, the impulse is applied at the first time step, WTZ=–1.

As indicated in Chapter 3, PMARC actually has three independent "input channels" that must be accounted for separately when using the impulse response method, a pitch channel, a pitch rate channel, and a plunge rate channel. The variable VZ represents the plunge rate, THETA is the pitch, and Q is the pitch rate. From the code it is evident that, in fact, as many as nine input channels are available for arbitrary motions in three dimensions—translation in the x, y, and z directions, pitch, yaw, and roll, and pitch, yaw, and roll rates. Because they are the most significant for aeroelastic applications, this research only considers vertical plunge and pitching rotations. All nine imput channels may be helpful for advanced flight dynamics analyses.

The implementation of PMARC/AIR for rigid bodies as listed in Appendix A.1 is rather inelegant, in that a separate executable is needed for each input channel, but functional. (This is fixed in the flexible-body implementation, as will be detailed shortly.) To find the impulse response of a given channel, first the remaining channels are set to zero for all time steps. The selected input channel is also set to zero for all time steps except the impulse time step, which is given by the abolsute value of WTZ. At the impulse time step, a pulse is applied to the selected channel. For example, to obtain a plunge rate impulse response, Q and THETA are both set to zero for all time steps and VZ is set to 1.0 for the pulse time step and 0.0 for all other times. Likewise, for a pitch impulse response, VZ and Q are set to zero for all times, and THETA is pulsed at the time step specified by WTZ. The output of the code is a data file containing the time history of the total lift coefficient in the body frame-of-reference—the impulse response. Figure 5-2 is a block diagram showing the process of an analysis using this method. The flow solver is run once for each input channel; once the impulse responses are obtained it is no longer needed. For subsequent responses to arbitrary input functions, only the convolution code is needed.

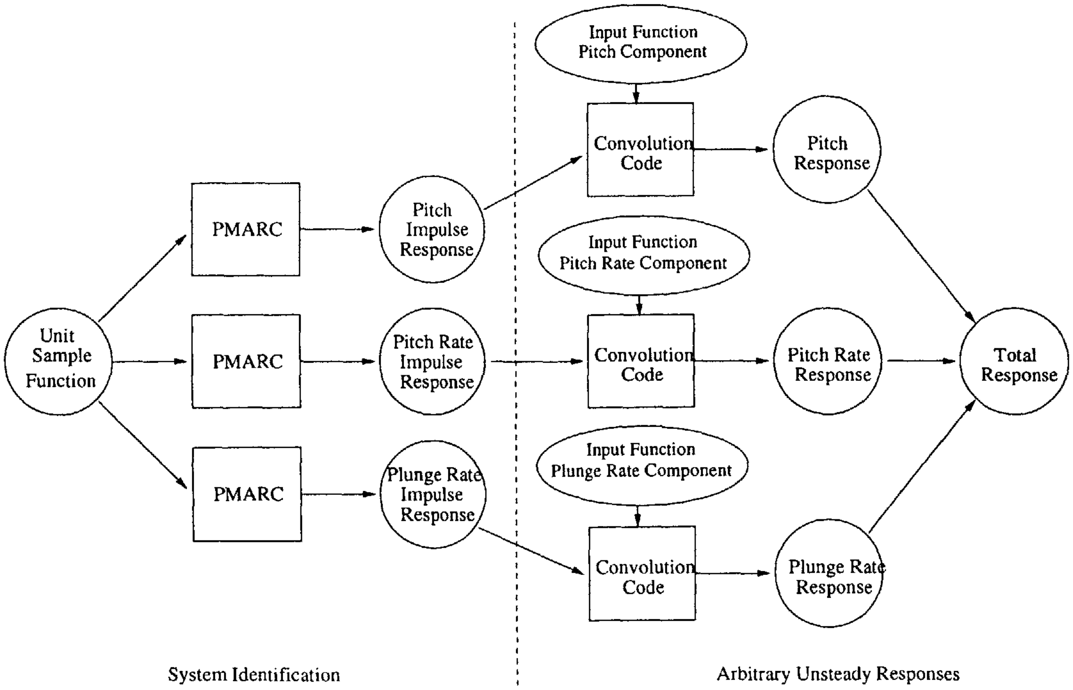


Figure 5-2: Block Diagram of an Unsteady Flow Analysis Using PMARC/AIR

PMARC is generally considered to be a linear code, but, in fact, this is not entirely true. While the governing equation of the flow is linear, the relaxed wake can actually be weakly nonlinear. This nonlinear behavior appears for large amplitude pitching and plunging. Initial results of convolving impulse responses with sinusoidal input functions revealed this amplitude-related nonlinearity. The impulse response method implies that the amplitude of the impulse should be 1.0 since a *unit* response is desired, but this does not always produce the correct result. As shown in Figure 5-3, convolution of the pitch channel impulse response with a sinusoidal function did not reproduce the expected response, while the pitch rate channel results, in Figure 5-4 show a good match. In fact, the units of the input channels must be considered, because, depending on the units, an impulse with magnitude 1.0 may be a large or a small input. The pitch channel is measured in radians, the pitch rate channel in radians per second, and the plunge rate channel in length units per second (where any length unit can be used and is specified in the PMARC input file). Because of the small angle assumption, the impulse must be small. A unit pulse of plunge rate and pitch rate is satisfactory; however, for the pitch channel, a pulse of 0.01 is used because 1 radian is a very large angle. The pitch channel impulse response is then scaled by 100 later to recover a "unit" response. A possible resolution of this pulse amplitude issue may be found by employing the nonlinear form of the Aerodynamic Impulse Response method, as is discussed in the next section.

5.2 Nonlinear PMARC/AIR for Rigid Bodies

As detailed in Chapter 3, the nonlinear form of the Aerodynamic Impulse Response method involves collecting the response of the system to multiple impulses. The first-order kernel of the nonlinear impulse response is exactly the same as the linear form, and the implementation within PMARC is the same as in the preceding section. The nonlinear responses studied in this thesis are limited to second-order kernels. Implementation of the method for a second-order kernel is the same as that described above for the linear impulse response except that two impulse are applied, the first

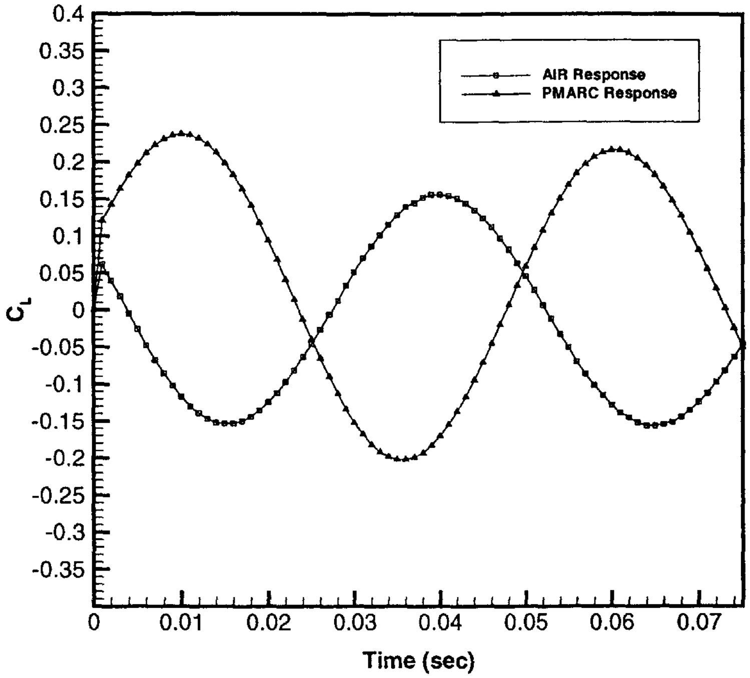


Figure 5-3: Pitch Response to a Sinusoidal Input Using an Impulse Amplitude of 1.0

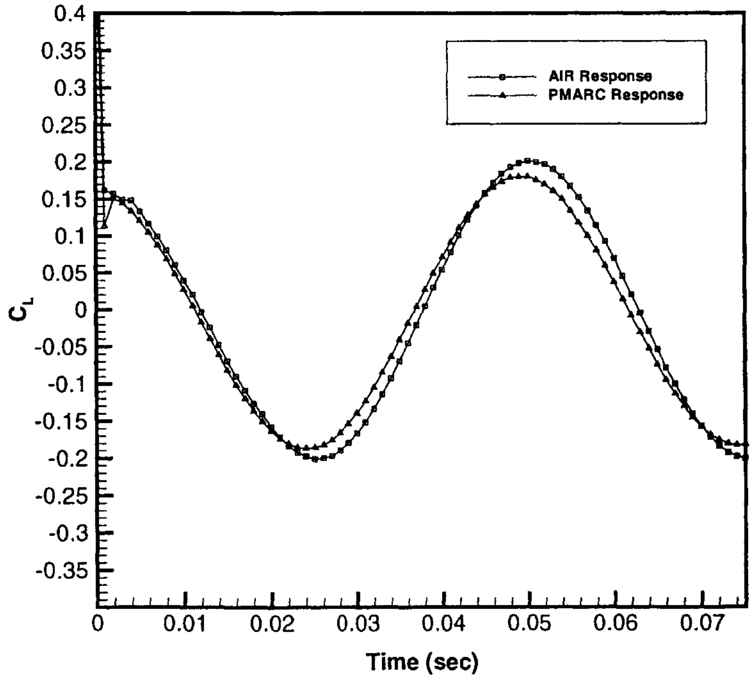


Figure 5-4: Pitch Rate Response to a Sinusoidal Input Using an Impulse Amplitude of 1.0

at time step n and the second at time step $n+T$. The code must be run repeatedly, varying the time $T\Delta t$ between the two impulses, until there is no interaction of the two impulses—when the T-th component of the second-order kernel goes to zero. The nonlinear kernel identification method was described in detail in Section 3.2.3.

5.3 Linear PMARC/AIR for Flexible Bodies

Significantly more complicated to implement is the Aerodynamic Impulse Response method for flexible bodies. This section details this implementation. First, the frame-of-reference and the relation of complex structural deformations to pitch and plunge motions are examined. Relevant assumptions are introduced, then a simple implementation of the method using a simplified aerodynamic formulation, a two-dimensional lumped vortex model, is presented. Results of this simplified formulation are examined to verify the validity of the approach. Finally, the modifications to PMARC are detailed.

5.3.1 Modeling Complex Deformations

There are two ways in which a flexible body can be represented in PMARC. The direct approach is, based on the output of the structural analysis, to update the geometry at each time step. This, however, is complicated and expensive. The other option is to relate the motions of the body to the unsteady flow parameters available, pitch and plunge. This later method is preferred.

PMARC Version 12.20 fixes the frame-of-reference on the body. This means that pitch and plunge of the body are actually represented by rotating the freestream velocity; this rotated freestream is refered to as the kinematic velocity (with the addition of rotation and translation rates in unsteady flows). In the rigid-body formulation, a single impulse amplitude is applied uniformly to the entire body. It is not much of an extension to suppose that if a varying impulse amplitude were applied to the body, some kind of bending or twisting of the body would be modeled. The flexible-body form of PMARC/AIR utilizes structural modal analysis to represent complex geometrical deformations as a series of mode shapes; these same mode shapes are used to define distributions of pitch and plunge to obtain the impulse response of each mode. Only linear flexible-body problems will be considered in this thesis; superposition is valid for linear problems, and the resulting modal responses can be appropriately scaled and superimposed.

This approach may be considered a fixed-panel frame-of-reference. Given the spanwise deformation distribution of a structural mode, the velocity induced by this motion is determined at each panel. Figure 5-5 illustrates linear bending of a wing—a non-physical phenomenon—or rigid body rotation about the wing root (roll). The principle holds for any modal shape; the linear mode was used for simplicity. Assuming that the structural analysis finds linear bending to be the first structural mode, in the body's frame-of-reference, it appears that the wing bends linearly along the span. In the panel-fixed frame-of-reference, however, the geometry remains unchanged. Instead, the wing sees an additional plunge velocity that increases linearly from zero at the wing root to a maximum value at the wing tip, as is shown in Figure 5-6. For an impulse response of this mode, the plunge velocity varies linearly from 0 at the root to 1 at the tip. Linear bending is the simplest of examples. Likewise, if the pitch is varied linearly, linear twist is modeled. Higher-order pitch and plunge distributions give higher-order mode shapes.

5.3.2 Assumptions

The implementation of the flexible-body form of the Aerodynamic Impulse Response presented here has two primary limitations. First, the wing is assumed to have a rigid chord. In other words, the airfoil does not change as the wing is deflected. This is a common assumption that is valid primarily for high aspect ratio wings, for which spanwise bending and twisting are the dominant forms of deflection. Additionally, the implementation presented here is valid only for small deflections about the initial geometry. This limitation is a result of the influence coefficient matrices being calculated only once (see Section 4.2 and Figures 5-1 and 5-11). For rigid bodies, the distance between panels never changes; for flexible bodies with small deformations the

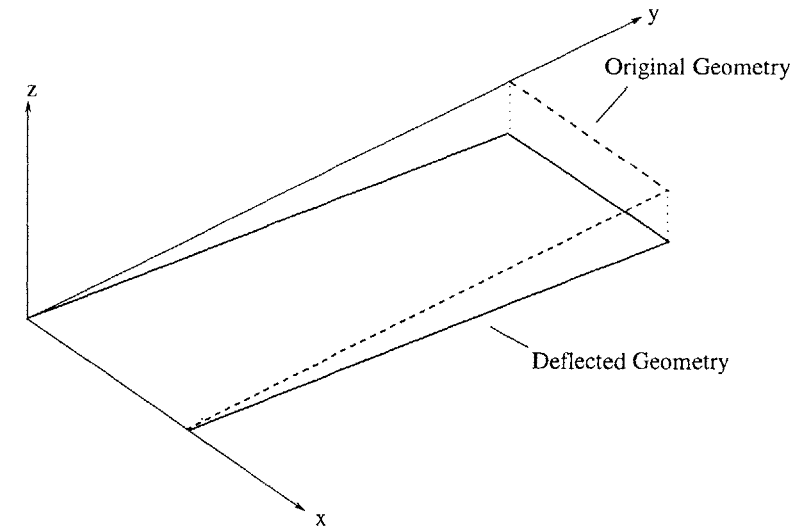


Figure 5-5: Linear Bending of a Rectangular Wing

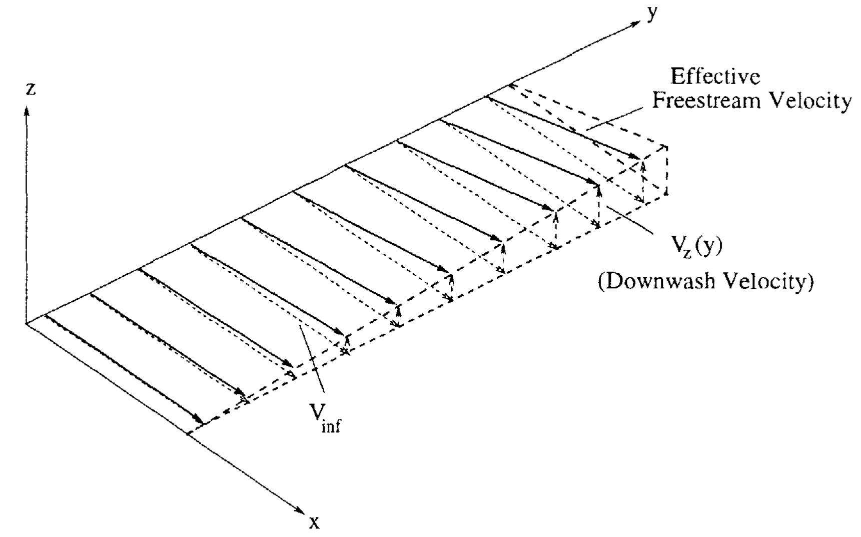


Figure 5-6: Effective Freestream Velocity Equivalent of Linear Bending of a Rectangular Wing

change in the distance between panels is negligible; however, for flexible bodies with large deformations, the influence coefficient matrices need to be recomputed at each time step based on the current geometry. This is not undertaken in this research, hence the flexible-body responses are valid only for small (linear) deformations—deformations about either a static or a nonlinear equilibrium.

5.3.3 Two-Dimensional Lumped Vortex Model

Prior to addressing the complex code modifications to implement the flexible-body impulse response in PMARC, the approach was investigated using a simple two-dimensional lumped vortex formulation. MATLAB was used for this study. A copy of the input file is included in Appendix B.1.

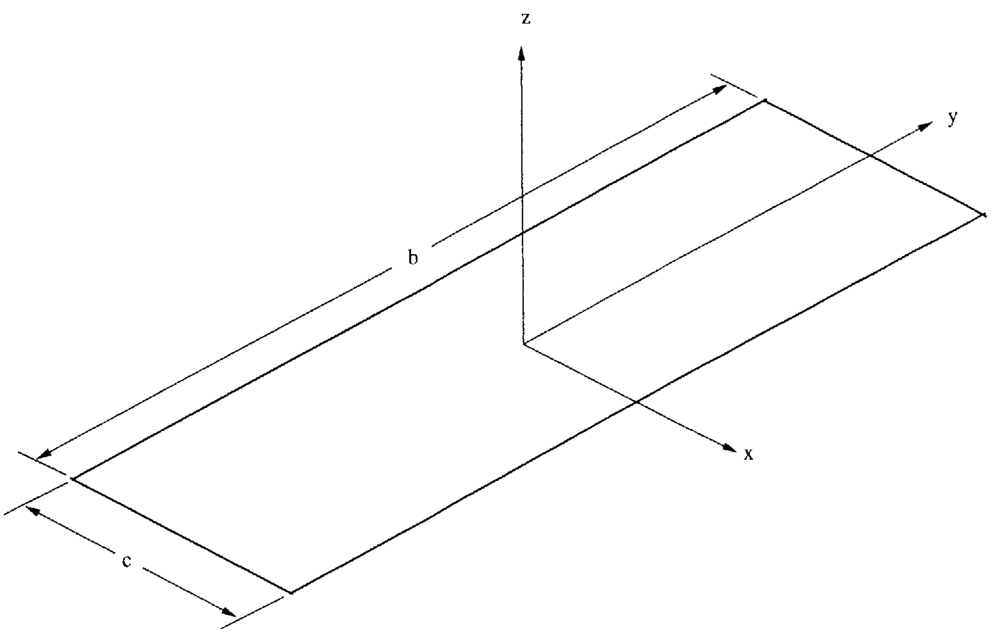


Figure 5-7: Rectangular Wing Modeled with the Lumped Vortex Formulation

Figure 5-7 shows the geometry studied with this model, a flat rectangular wing with span b and chord c. Discrete vortex elements with circulation Γ are distributed along the span at the quarter-chord point as shown in the two-dimensional section in Figure 5-8. The downwash induced by the vortex element is computed at the

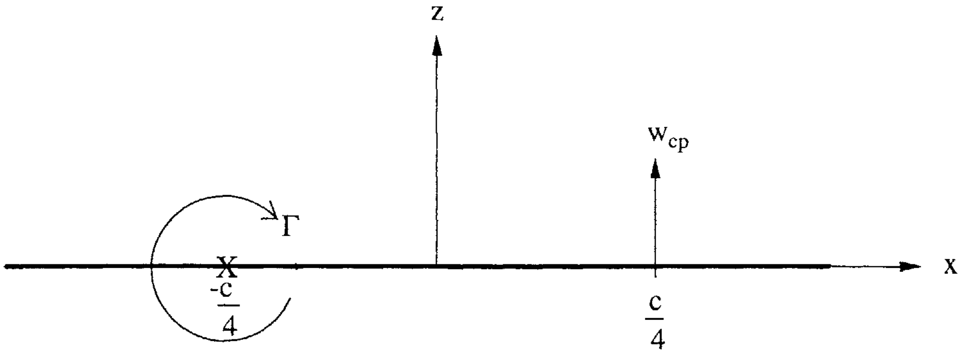


Figure 5-8: 2-D Section of the Wing Modeled with the Lumped Vortex Formulation

three-quarter-chord point according to the equation

$$w_{cp} = -\frac{\Gamma(t)}{\pi c} - \frac{\beta}{U_\infty}\frac{d\Gamma(t)}{dt} \tag{5.1}$$

where c is the airfoil chord and β is a lag parameter found to equal $2/\pi$ by matching to Theodorsen theory for large reduced frequencies, $k \to \infty$ [5]. Strip theory is assumed: each spanwise location is considered independently, and downwash at each spanwise station is induced only by the local vortex element.

The wing surface in Figure 5-7 is defined as

$$F(x, y, z, t) = z_A(x, y, t) - z = 0 \tag{5.2}$$

The flow tangency boundary condition requires that the flow be aligned with the body. This is described mathematically with the substantial derivative operator, giving $\frac{DF}{Dt} = 0$. Using the definition of F from Equation 5.2, manipulating, and dropping higher-order terms,

$$\frac{\partial z_A}{\partial t} + U_\infty \frac{\partial z_A}{\partial x} - w' = 0 \tag{5.3}$$

where w' is a small perturbation velocity in the z-direction. (Note that x- and y-components of the perturbation velocity were dropped as higher-order terms.) Thus, at the control point,

$$w_{cp} = \left(\frac{\partial z_A}{\partial t} + U_\infty \frac{\partial z_A}{\partial x} \right)\bigg|_{cp} \tag{5.4}$$

Now, the surface section z_A is writen in terms of pitch and plunge variables, α and h. This gives

$$z_A = h(t) - x \tan \alpha(t) \tag{5.5}$$

For small angles α and suppressing the explicit time dependence,

$$z_A = h - \alpha x \tag{5.6}$$

and

$$\frac{\partial z_A}{\partial t} = \dot{h} - x\dot{\alpha} \tag{5.7}$$

$$\frac{\partial z_A}{\partial x} = -\alpha \tag{5.8}$$

Combining Equations 5.7 and 5.8 with Equation 5.4 gives

$$w_{cp} = \left(\dot{h} - x\dot{\alpha} - U_\infty \alpha \right)\Big|_{cp} = \dot{h} - \frac{c}{4}\dot{\alpha} - U_\infty \alpha \tag{5.9}$$

Using the definition of the downwash, Equation 5.1, and manipulating gives the governing differential equation for the circulation about a pitching and plunging wing section:

$$\frac{d\Gamma}{dt} + \frac{1}{\beta}\frac{U_\infty}{\pi c}\Gamma = \frac{U_\infty^2}{\beta}\alpha + \frac{U_\infty}{\beta}\frac{c}{4}\dot{\alpha} - \frac{U_\infty}{\beta}\dot{h} \tag{5.10}$$

The solution of this governing equation is found numerically using a first-order forward difference finite difference scheme in MATLAB. The spanwise distribution of circulation is then found to be

$$\Gamma_{i+1,j} = \frac{U_\infty \Delta t}{\beta} \left(U_\infty \alpha_{i,j} + \frac{c}{4}\dot{\alpha}_{i,j} - \dot{h}_{i,j} - \frac{1}{\pi c}\Gamma_{i,j} \right) + \Gamma_{i,j} \tag{5.11}$$

where i is the temporal variable and j is the spatial variable representing the span section.

As in PMARC, three separate input channels are available. This study examines the pitch and pitch rate channels only; plunge is assumed to behave similarly. For the impulse responses, the inputs are as follows:

For pitch:

$$\alpha(y,t) = \begin{cases} \frac{2}{b}y & \text{for } t = t_0 \\ 0 & \text{for } t \neq t_0 \end{cases} \quad (5.12)$$

$$\dot{h}(y,t) = 0 \text{ for all } t \quad (5.13)$$

$$\dot{\alpha}(y,t) = 0 \text{ for all } t \quad (5.14)$$

For pitch rate:

$$\dot{\alpha}(y,t) = \begin{cases} \frac{2}{b}y & \text{for } t = t_0 \\ 0 & \text{for } t \neq t_0 \end{cases} \quad (5.15)$$

$$\dot{h}(y,t) = 0 \text{ for all } t \quad (5.16)$$

$$\alpha(y,t) = 0 \text{ for all } t \quad (5.17)$$

The pitch and pitch rate impulse responses are shown in Figure 5-9. The response to sinusoidal pitching is desired, so the following input functions are used:

$$\alpha_{inp}[i] = \sin\left(\frac{2\pi i\Delta t}{5}\right) \quad (5.18)$$

$$\dot{\alpha}_{inp}[i] = \frac{2\pi}{5}\cos\left(\frac{2\pi i\Delta t}{5}\right) \quad (5.19)$$

where i is the time index. This is a simple sine function and its time derivative with unit amplitude and a frequency of 0.2 Hz. The built-in MATLAB convolution function is used to convolve each impulse response with its corresponding input function. The total response to sinusoidal pitching is obtained by superimposing the pitch channel response and the pitch rate channel response. The result is shown in Figure 5-10.

A full time-marching solution of the differential equation is used as a control case.

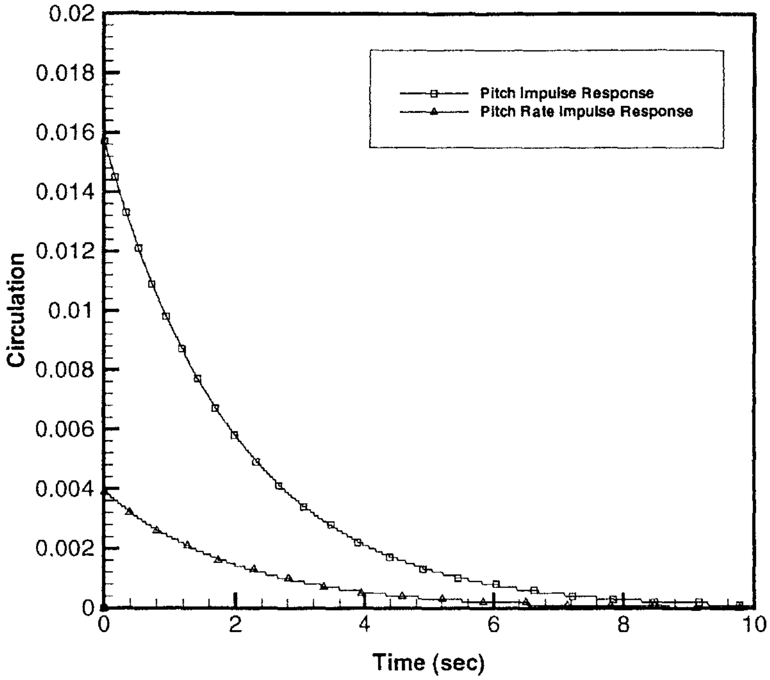


Figure 5-9: Impulse Responses of the Pitch and Pitch Rate Channels, $\Delta t_{impulse} = 0.01$ seconds

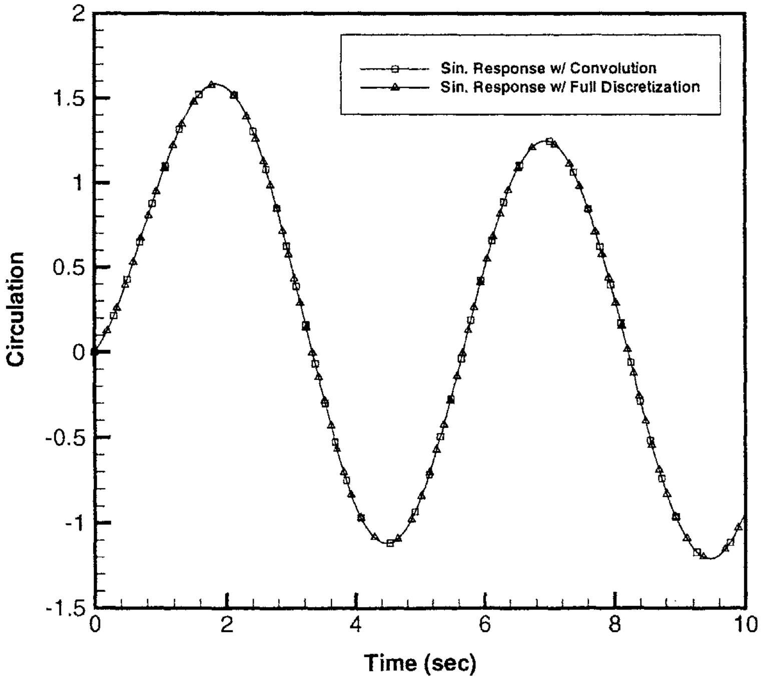


Figure 5-10: Total Response at the Wing Tip to Sinusoidal Pitching Using the AIR Method and by Full Discretization of the Governing Equation

Neglecting the plunge channel, this gives

$$\Gamma_{i+1,j} = \frac{U_\infty}{\beta}\left[\left(U_\infty \Delta t - \frac{c}{4}\right)\alpha_{i,j} + \frac{c}{4}\alpha_{i+1,j} - \frac{\Delta t}{\pi c}\Gamma_{i,j}\right] + \Gamma_{i,j} \tag{5.20}$$

with the input

$$\alpha_{inp}[i,j] = \frac{2}{b} j \Delta y \sin\left(\frac{2\pi i \Delta t}{5}\right) \tag{5.21}$$

where i is the time index, j is the span station index, and Δy is the width of each span station. This equation has the same form as Equation 5.18 except that here the spanwise variation of pitch angle is imbedded in the input function.

The solution of Equation 5.20 for this input is determined with MATLAB; the result is shown in Figure 5-10. Clearly, the impulse response method is equivalent to the full time-marching solution. The result is shown for the wing tip only; examination of the three-dimensional response shows that both responses are equivalent along the entire wingspan. It was thus considered justified to continue with the implementation of this approach for flexible-body response in PMARC.

5.3.4 PMARC Implementation

To implement the flexible-body form of the Aerodynamic Impulse Response in PMARC, changes were required throughout the code. For this reason, Appendix A.2 includes only the most significant portions of the code modifications. Figure 5-11 illustrates the modifications diagrammatically. Comparing to the flowchart of the rigid-body PMARC/AIR methodology in Figure 5-1, it is evident that superficially the only change is the addition of a loop over the wing span within the time iteration loop.

The primary modification is to convert the kinematic velocity arrays VFR and OMEGA from one-dimensional to two-dimensional arrays. In their original form, VFR contains the three components of velocity and OMEGA the three rotation rate components. In modified form, VFR and OMEGA become $(N+1) \times 3$ arrays, where N is the number of columns of panels. Thus a unique kinematic velocity may be used

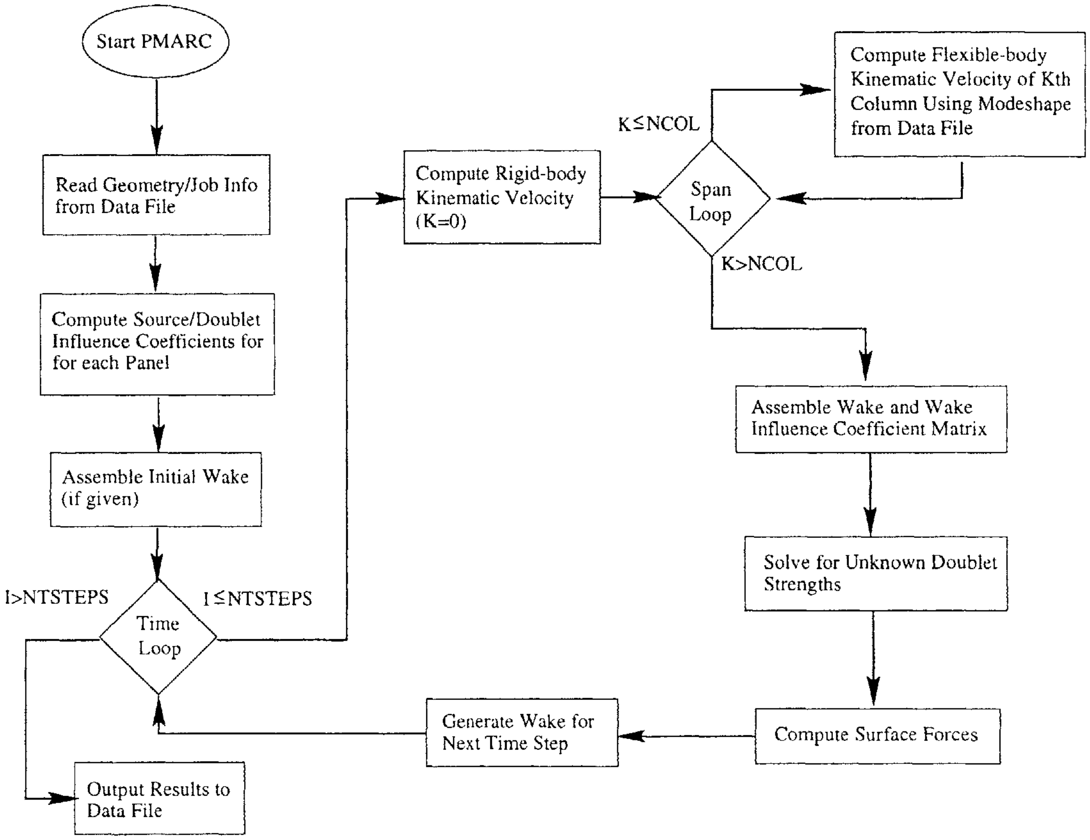

Start PMARC
Read Geometry/Job Info from Data File
Compute Source/Doublet Influence Coefficients for for each Panel
Assemble Initial Wake (if given)
I>NTSTEPS
Time Loop
I≤NTSTEPS
Output Results to Data File
Compute Rigid-body Kinematic Velocity (K=0)
K≤NCOL
Span Loop
K>NCOL
Compute Flexible-body Kinematic Velocity of Kth Column Using Modeshape from Data File
Assemble Wake and Wake Influence Coefficient Matrix
Solve for Unknown Doublet Strengths
Compute Surface Forces
Generate Wake for Next Time Step

Figure 5-11: Flowchart Outlining the Flexible-Body PMARC/AIR Methodolgy

at each span station. (By assuming the wing to be rigid in the chordwise direction, the kinematic velocity is constant at each span station, thus reducing the problem size by obviating the need to compute and store the kinematic velocity at every panel.) The first column (with index 0) of these arrays is reserved for the rigid-body kinematic velocity components, which is used on all panels not specified as wing panels. To fill these arrays, an additional loop is required within the time iteration. With this new scheme, at each time step the program loops over the wing columns and determines the local kinematic velocity based on a mode shape inputted by way of a data file.

This implementation could, in fact, be more efficient. For large problems execution time will increase because of the need to compute the kinematic velocity at each span station for each time step. A marked improvement in the implementation's efficiency would be achieved by noticing that the kinematic velocity is actually uniform for all time steps except at the moment of the impulse, $t_{impulse}$. Thus execution of the additional spanwise loop is actually only required at one time step.

Job control is handled more elegantly in this implementation than in that for rigid bodies. An additional namelist is added to the PMARC input file to specify whether, when, and to which channel an impulse should occur. Three variables are added in this namelist:

CHAN specifies the input channel to be used, with

1 – Plunge

2 – Pitch

3 – Pitch Rate

IMPTYP specifies the type of impulse used, with

0 – No impulse; standard execution of PMARC

1 – Rigid-body impulse

2 – Flexible-body impulse with distribution stored in the file IMPULSE

STEP specifies the time step n at which to apply the impulse

Mode shapes are specified with an input file, IMPULSE. This file lists the impulse magnitude at each span station. All three input channels are listed in the same file. For simplicity, equal spacing of panels in the spanwise direction is assumed for this research. An example of the impulse mode shape file IMPULSE for linear bending is given in Appendix C.4. The mode shape specified in the input file should be obtained from a structural modal analysis.

Because for flexible-body problems the concern is with the spanwise lift distribution, the output is the sectional lift coefficient rather than the total lift coefficient, as was used in rigid-body analyses. This output can then be passed to a structural dynamics module to find the structural response to that lift distribution.

The modifications mentioned here should leave the code "backwards compatible." Namely, the response with no impulse (IMPTYP=0) should be identical to that from the unmodified PMARC code; when IMPTYP=1, the response should be equivalent to that obtained with the rigid-body implementation discussed above; and when the flexible mode is selected (IMPTYP=2), input of a constant impulse distribution should duplicate rigid-body responses. These comparisons were studied as part of the validation of the implementation. The results of these studies are presented in Section 7.4.

5.4 Other Codes Developed

Various utility programs were developed over the course of the research. These are briefly described here, and the source code is reproduced in the Appendix.

5.4.1 Convolution

The linear convolution scheme employed in this program is a simple one based on the definition of convolution in discrete-time,

$$y[n] = \sum_{k=0}^{N} h[n-k]x[k] \tag{5.22}$$

This simple formulation is adequate for moderate sized jobs. (Execution takes less than a second on an Intel Pentium II for $N \approx 1000$.) Very large jobs can be processed more expediently by utilizing one of several more efficient convolution schemes.

The inputs to the convolution code are as follows: the name of the impulse response file, an impulse scaling factor (100 for the pitch channel, 1 for the pitch rate and plunge rate channels), and the names of the input function file and the output file. The total C_L response obtained form the convolution for each time step is written to the specified file in columnar form.

Convolution of flexible-body impulse responses requires a slight modification to the base convolution scheme. Because the impulse response is different at each span station, an additional loop was included. The user is prompted for the number of span stations on the wing, and the convolution scheme is executed at each station using the global input function and the local impulse response function.

More extensive modifications are required for "nonlinear convolution" by way of the Volterra series, Equation 3.4. Two input files are required here, one for the first-order kernel and a second for the second-order kernel. The first input file is identical in structure to the input file for linear convolution, and the data is read into an array in the same way as in linear convolution. The second-order kernel is listed component by component in the second input file. The data is organized into a symmetric two-dimensional array with the first component along the main diagonal, the second component in the next diagonal, and so forth. Any remaining array elements are set to zero. The "convolution" scheme is performed in two parts, the first the same as previously for the first-order kernel, the second requiring an additional loop for the double summation of the second-order kernel.

5.4.2 Deflected Wing Generation

For the steady and unsteady cases discussed in Section 7.2, a program was developed to generate the PMARC input file for a nonlinearly deflected NACA 00xx-series rectangular wing. The user inputs the airfoil thickness ratio, the span, the chord, discretization options, and the wing tip deflection ratio. This wing tip deflection ratio is actually the linear value and not the *true* wing tip deflection ratio; that is, the ratio of the projected tip deflection at $y = b$ to the semi-span, b. Figure 5-12 illustrates this difference. The deflection shape is quadratic. The program generates two output files. The contents of the first, wing.txt, can be pasted into the geometry definition section of a PMARC input file. The second output file, curve.txt, gives the translation and rotation of each span station for use with the lift extrapolation method discussed in Section 7.2.

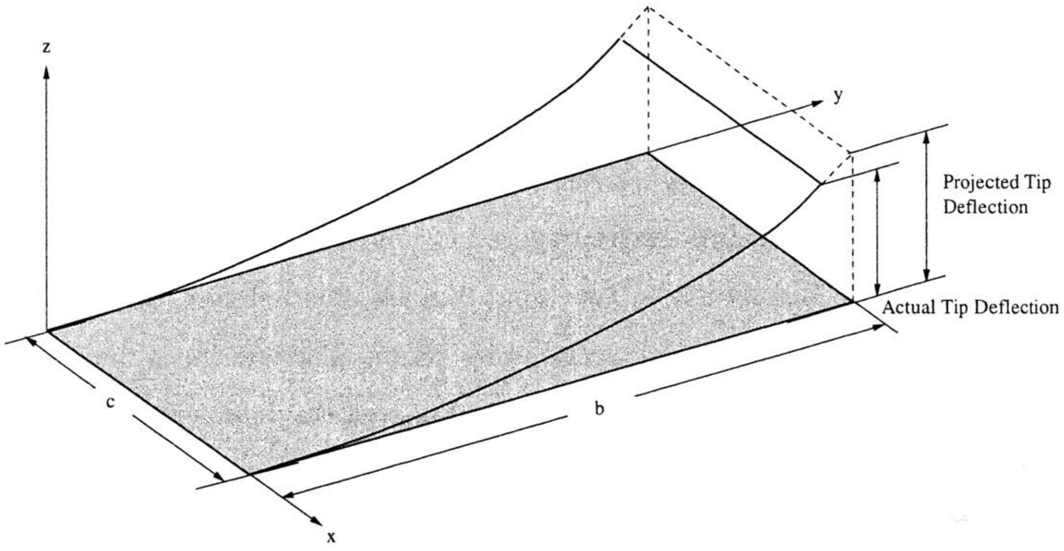


Figure 5-12: Detail of True versus Projected Wing Tip Deflections in Wing Generation Program

5.4.3 Sinusoidal Input Function Generator

A simple program to generate sinusoidal functions to convolve with impulse responses was developed as well. This program takes the user's input of the input type (pitch, pitch rate, plunge rate), the frequency, the amplitude, the time step size, and the duration, and writes the result to the file input.txt.

Chapter 6

Geometry Models

Two aspects of modeling are considered in this work. The first, as has been the focus up to this point, is the modeling of flow features. This chapter, on the other hand, concerns geometric modeling of the aerodynamic body—a wing—subjected to a flowfield. The chapter opens with a brief discussion of the highly flexible, high aspect ratio wings that motivate this work. Comments on the modeling procedure in PMARC follow, and the chapter closes by detailing each geometry model that is studied in this work.

6.1 Highly Flexible, High Aspect Ratio Wings

Though any geometry may be studied with the Aerodynamic Impulse Response method and the implementation of it described in this thesis, the particular application that motivated this work is the design and study of highly flexible, high aspect ratio wings. Such wings are found on a variety of aircraft but are of particular interest for High Altitude, Long Endurance (HALE) unmanned aircraft. What distinguishes the behavior of these wings is their high flexibility and, consequently, large deflection during flight. At the lowest level of analysis, wings are typically assumed to be rigid. At the next level, structural elasticity is accounted for, but deformations are assumed to be small. A highly flexible wing, however, experiences large deformations—bending up to 30% of its semi-span, for instance—and the traditional small displacement/rotation theories are not valid.

In the past, high aspect ratio wings generally were stiffened extensively to prevent large deflections and aeroelastic instabilities. However, doing so incurs a significant weight penalty. To achieve the performance goals set for the next generation of HALE vehicles, highly efficient designs are necessary. High aspect ratio wings are desirable for their high aerodynamic efficiency—the lift-to-drag ratio increases with wing span. The design must also be light and be capable of carrying enough fuel for missions extending to the tens or hundreds of hours. In addition, a large payload capacity is desired. To achieve these goals, one solution is to cut back on the wing stiffening, allowing the wings to bend and twist more, controlling the deformations through material tailoring and active control systems if necessary [60]. Substantial savings may be achieved but at the cost of greater complexity of the engineering analysis of the vehicle.

The Aerodynamic Impulse Response method in combination with PMARC is an effective and efficient aerodynamic analysis tool for the study of this type of wing. Combined with a modal (linear or nonlinear) structural analysis module, the aeroelastic characteristics of such a wing may be studied at the design level. Figure 6-1 illustrates a representative Highly Flexible, High Aspect Ratio (HFHAR) Wing design that will ultimately be the subject of study with the combined suite of aerodynamics, structural, and controls tools. This wing has NACA 0014 cross section, an effective chord, $\bar{c}$, of 0.169 m, and a semi-span aspect ratio of 11.8; the inboard section is rectangular, and outboard the wing tapers to half the root chord. For this research, the geometry is slightly simplified. The ensuing sections describe the geometries used and how they are modeled with PMARC.

6.2 Geometry Modeling with PMARC

Geometry modeling in PMARC is accomplished via a data input file. Data is read by PMARC by way of a series of namelists, each of which contains a set of job control variables. The first section of the the input file contains basic job control data and

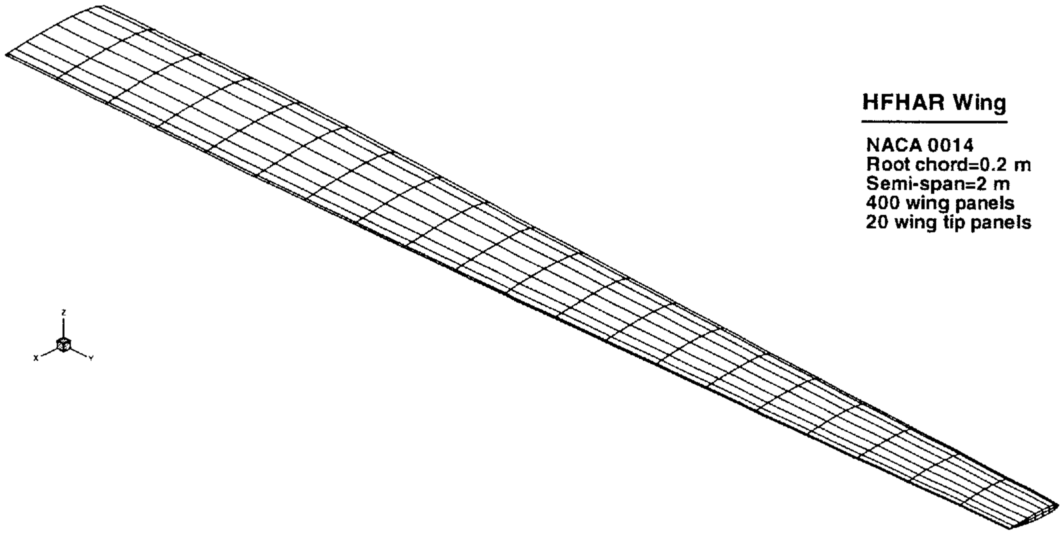


Figure 6-1: Highly Flexible, High Aspect Ratio Wing Modeled in PMARC

global geometry information. The following are the most pertinent of the namelists contained in this section of the input file:

- Basic job control and print options
- Solver parameters
- Time step parameters
- Symmetry and computation parameters
- Freestream conditions
- Angular position and constant rotation rates
- Sinusoidal rotation and translation
- Reference dimensions

In the flexible-body implementation of PMARC/AIR, a new namelist is added to this section to control the impulse, as was described in Section 5.3.4.

The next section contains the geometry definition. Complex geometries can be handled by taking advantage of the hierarchy of coordinate systems in PMARC. Large, uniform surfaces such as a wing or fuselage section are defined as *patches* by two or more *sections*, which describe cross sections of the patch. Patches are then discretized into small rectangular *panels*. Patches are generally rectangular, but they may be curved, and they may have a side with zero length or be folded such that two sides are coincident. Wings are modeled this way, a single patch folded such that two originally opposite sides form the trailing edge. Because PMARC is a low-order code, exact matching of panels at the patch boundaries is not necessary. Patch and section options allow the user to control the number of panels on each patch as well as the spacing of the panels. Four spacing options are available,

- Equal spacing
- Full-cosine spacing, which places the smallest panels near both edges
- Half-cosine spacing, with smallest panels near the first edge
- Half-cosine spacing, with smallest panels near the opposite edge

For wings, typically full-cosine spacing is employed in the chordwise direction to concentrate panels near the leading and trailing edges, and half-cosine spacing is used in the spanwise direction to concentrate panels near the wing tip. NACA 4-series wings can be automatically generated by PMARC in this section, and PMARC also can automatically generate a closing wing tip patch, either flat or rounded.

Wake definition follows the geometry definition. Here the user specifies all edges on the body from which a wake is shed. Wakes can be set to either rigid or relaxed mode, and initial wake shapes can be specified if desired. In rigid mode the wake is time-stepped with the freestream velocity, while in relaxed mode the wake is time-stepped with the local velocity.

The final sections of the data file are for special options including off-body velocity scans and off-body streamlines. These options are not used in this research.

6.3 Simple High Aspect Ratio Wing

Studies with PMARC and the Aerodynamic Impulse Response detailed in this thesis are limited to wing models only. The wing model used, the Simple High Aspect Ratio Wing (referred to as the SHAR Wing), is a slightly simplified version of the HFHAR Wing—principally, the wing is rectangular with no tapered section. Again, the airfoil is a NACA 0014. Various configurations of this wing design are studied throughout this research. For the initial studies with PMARC described in Section 7.2, a wing with a chord of 0.2 m and semi-span of 2 m (hence an aspect ratio of 10) is used, as illustrated in Figure 6-2. The wing is discretized in PMARC with 750 wing surface panels and 30 wing tip panels. Panels are equally spaced in the spanwise direction and are full-cosine spaced in the chordwise direction, which results in smaller panels at the leading and trailing edges. A relaxed wake model is used with no initial wake, and the freestream flow is 10 m/s. The freestream density is set by specifying a speed of sound. Here, the sea-level value, 340 m/s, is used. This configuration is referred to subsequently as "SHAR1" or the "baseline" configuration; a listing of the PMARC input file for this model is included in Appendix C.1.

Flat NACA 4-series wings are simple to model in PMARC because they can be automatically generated. For the initial studies, however, it was desired to study this same SHAR1 Wing in a highly deflected state. The deflected wing generation program described in the previous chapter was used for this purpose. Figure 6-3 shows a resulting wing model in a quadratic bending mode, where the tip is deflected (nonlinearly) 29.3% of the semi-span. This configuration is referred to as the "deflected" SHAR1 configuration.

For the impulse response method studies, the wing configuration is similar, though with a semi-span aspect ratio of 50. Additionally, the length units are feet rather than meters. As before, a relaxed wake is used. Freestream flow in this case is 100 ft/s,

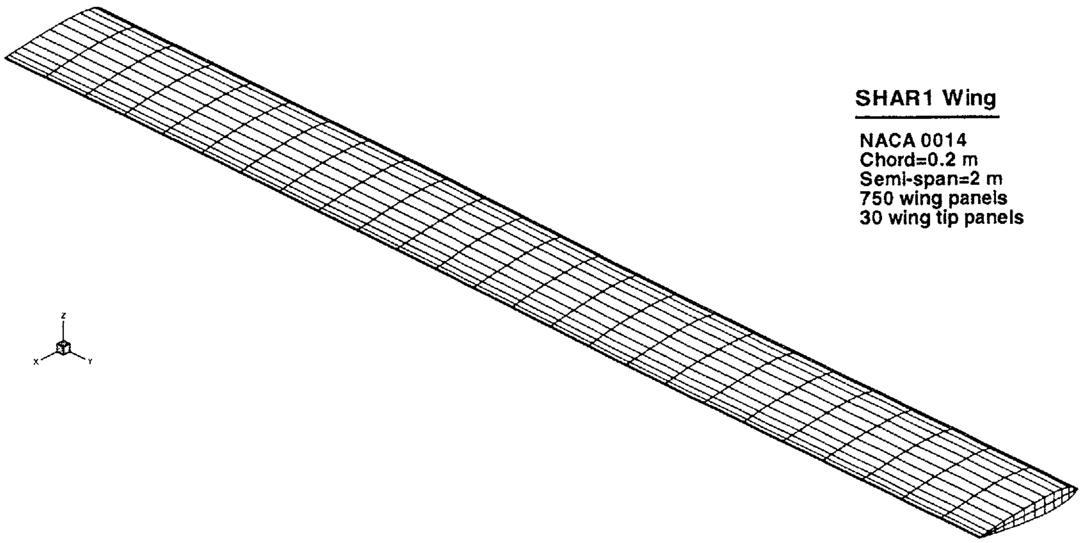


Figure 6-2: Simple High Aspect Ratio Wing (SHAR1) Modeled in PMARC

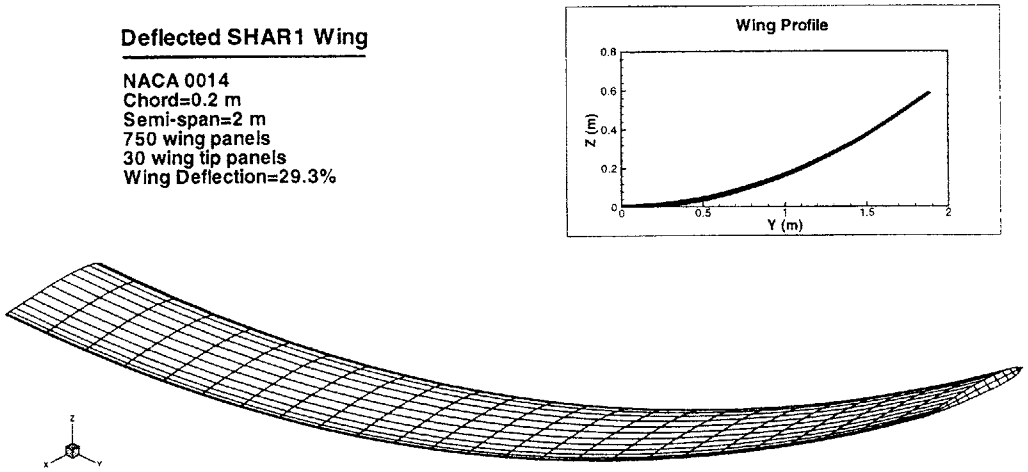


Figure 6-3: Simple High Aspect Ratio Wing (SHAR1) with 29.3% Bending

and the speed of sound is set to its sea-level value, 1116 ft/s. Figure 6-4 illustates the primary wing configuration used with PMARC/AIR, which is referred to as the "SHAR2" Wing. The PMARC input file listing for this configuration can be found in Appendix C.2. Additionally, studies were performed varying the aspect ratio of the wing to examine sensitivity to three-dimensional flow effects. Impulse response method results for these configurations are presented in Sections 7.3 and 7.4.

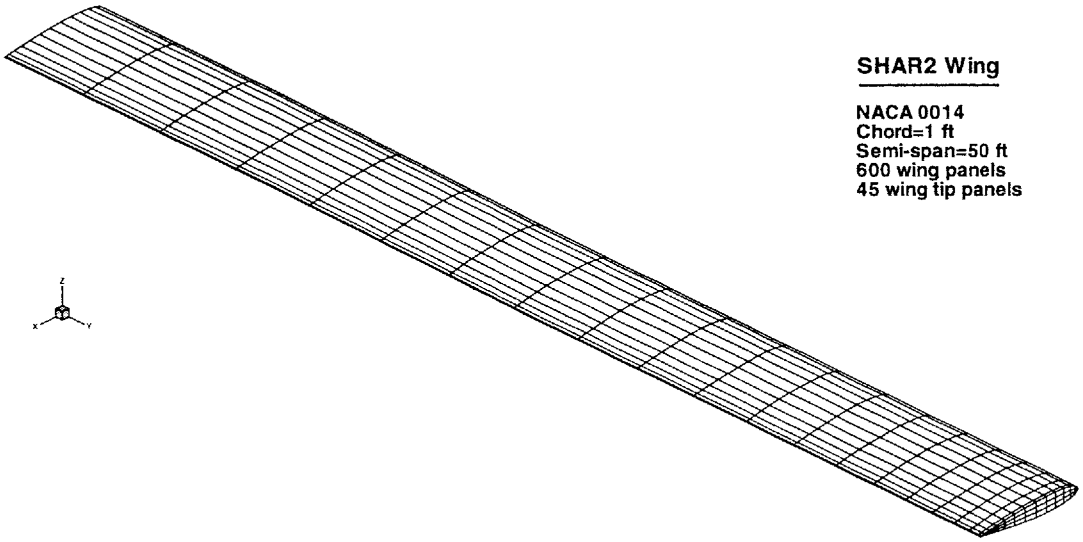


Figure 6-4: Simple High Aspect Ratio Wing (SHAR2)

6.4 BACT Wing

With an aspect ratio of 2, the BACT (Benchmark Active Controls Technology) Wing is not considered a highly flexible, high aspect ratio wing; however, it was chosen for initial testing with PMARC/AIR because of its simplicity and to study the possible aspect ratio dependence of the method. Additionally, experimental data is available from wind tunnel testing at NASA Langley, though only at elevated

Mach numbers [61]. The wing is a rectangular NACA 0012 with a chord of 16 inches and a semi-span of 32 inches. The wing tip is rounded. In PMARC, 600 panels are distributed on the wing surface and 60 on the wing tip. The wing surface is discretized with full-cosine spacing of the rows and half-cosine spacing of the columns, so that panels are smallest at the leading edge and trailing edge and towards the wing tip. The relaxed wake model is employed with no initial wake. Freestream velocity is 100 ft/s. Figure 6-5 shows the PMARC model of the BACT Wing, and the geometry input file is listed in Appendix C.3.

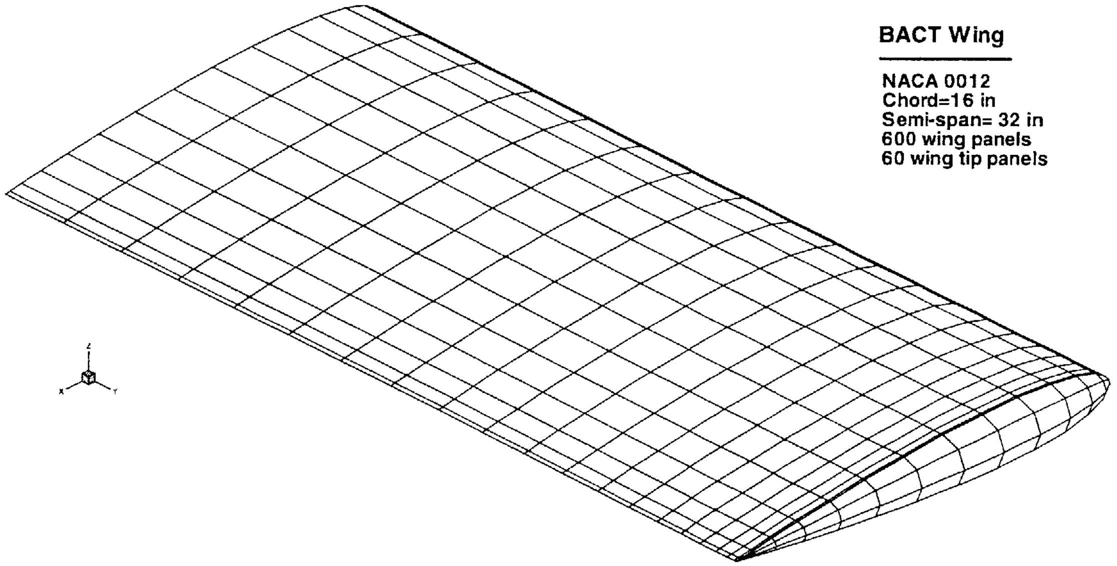


Figure 6-5: BACT Wing

Chapter 7

Results and Discussion

Two codes resulted from the modifications to PMARC discussed in Chapter 5, the rigid-body PMARC/AIR code and the flexible-body PMARC/AIR code. This chapter presents some of the results obtained with the final versions of each of these codes for the purpose of validation of the method. The unmodified form of PMARC was used, when possible, to generate control data against which to compare the results predicted with the Aerodynamic Impulse Response method. This chapter begins by briefly describing the error analysis method employed where needed, then the numerical studies are presented, organized as follows:

- Preliminary testing
    - Steady baseline versus deflected configurations
    - Unsteady baseline versus deflected configurations
    - PMARC unsteady results for various time steps
    - Separation of input channels
- Impulse response tests for rigid bodies
    - Linear impulse responses
    - Linear response to sinusoidal motions
    - Linear response to step motions

  - Linear response to arbitrary motions
  - Nonlinear impulse responses
  - Nonlinear response to sinusoidal motions
- Impulse response tests for flexible bodies
  - Linear response with no impulse
  - Linear response with a constant impulse
  - Linear response to distributed impulses and various inputs

7.1 Error Analysis Method

Throughout this chapter, results obtained using the AIR method are presented in comparison with those generated by PMARC. In some cases a qualitative comparison of the results is adequate; however, when the error needs to be quantified, Fast Fourier Transforms (FFTs) of the data are used. As implemented in MATLAB, Fourier transforms are a sine/cosine series expansion of the original function with the form

$$f(n) \approx a_0 + \sum_{k=1}^{N/2} \left(a_k \cos\left(\frac{2\pi kt}{N\Delta t}\right) + b_k \sin\left(\frac{2\pi kt}{N\Delta t}\right)\right) \tag{7.1}$$

where k is the discrete frequency index. This can be rewritten as

$$f(n) = a_0 + \sum_{k=1}^{N/2} A_k \sin\left(\frac{2\pi kt}{N\Delta t} + \phi_k\right) \tag{7.2}$$

With this result, three properties of a signal can be easily quantified for comparison with those of another signal: the frequency or frequencies of the signal, k, the amplitude of these frequencies, A_k, and the phase angle of each frequency, ϕ_k. The MATLAB 'fft' function was used to compute the Fourier transform. Care must be taken to ensure that the input signal is exactly an integer number of cycles. For the results presented in this chapter, the second cycle of the response was used for the FFT analysis because the first few data points from both PMARC and PMARC/AIR can be erratic. With this input, MATLAB returns a complex-valued frequency domain transform. The real values of the transform give the cosine coefficients, a_k, and the imaginary values give the sine coefficients, b_k. The total amplitude coefficients A_k and phase angles ϕ_k are obtained by simple trigonometric manipulation. Error results for the response amplitude are presented in relative form with respect to the PMARC results, so

$$\%Error = \frac{A_{k,AIR} - A_{k,PMARC}}{A_{k,PMARC}} \times 100\% \tag{7.3}$$

Because of the very small phase angles of the responses, phase error is shown in absolute form, where

$$\Delta\phi_k = \phi_{k,PMARC} - \phi_{k,AIR} \tag{7.4}$$

and is measured in radians.

7.2 Preliminary Testing

Two primary purposes motivate this preliminary set of tests: first, to evaluate the PMARC unsteady routines; and second, to quantify the aerodynamic effects of a highly deflected lifting surface. Preliminary testing was done prior to any implementation of the Aerodynamic Impulse Response; the only modification to the code is the separation of input channels for an unsteady PMARC test.

7.2.1 PMARC Unsteady Tests

Before proceeding with the implementation of the impulse response method, though, two validation tests of PMARC's unsteady capabilities were performed. The intent of these tests was to establish whether PMARC is an effective platform for the AIR method and if it may be used to generate control data against which to compare the results from the AIR method. Both tests use the SHAR2 model described in the previous chapter.

First, the time step dependence of the unsteady responses was investigated. Responses from PMARC at various values of Δt were compared at different pitch amplitudes and frequencies. Figures 7-1 and 7-2 show the results for moderate pitch amplitude (3° was the amplitude tested). Figures 7-3 and 7-4 present the results at high pitch amplitudes (15°). For the range of time steps tested, it is evident that the response is nearly independent of time step size. The significance of this conclusion is that control data generated with the original, unmodified PMARC code does not need to have the same time step size as the responses generated with the AIR method. This allows for a substantial reduction in the time required to collect the necessary responses. A small degree of time step size dependence is noted for the large amplitude, 10 Hz case in Figure 7-4. Care will need to be taken in selecting control data to use with high amplitude, high frequency test cases.

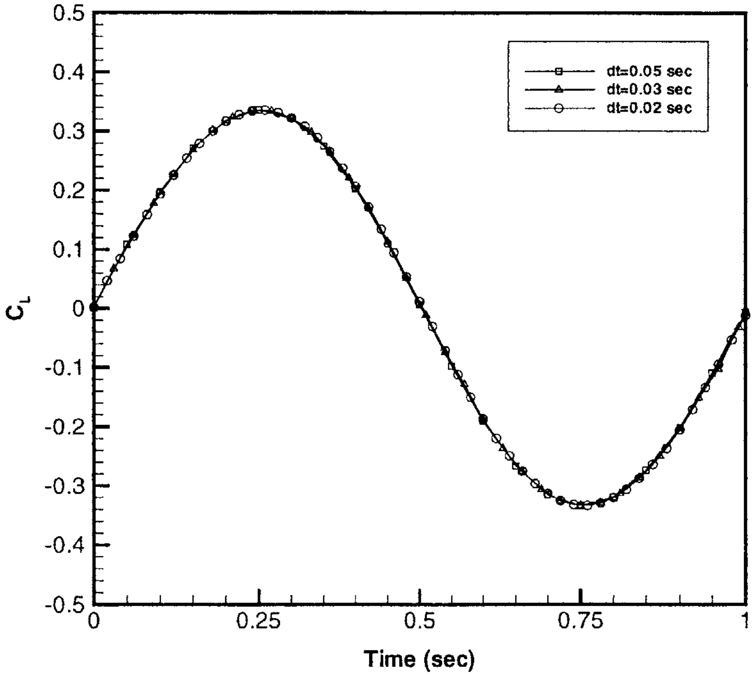


Figure 7-1: 1 Hz, 3° Pitch Response of the SHAR2 Wing from PMARC at Various Values of Δt

The second validation test confirms that the total unsteady response may be broken into pitch, pitch rate, and plunge rate components as is required by the AIR method. The total response of the SHAR2 Wing to 1 Hz, 3° pitching oscillations was found using the PMARC unsteady routines. The response of the pitch channel only was found by setting the pitch rate variable, Q, to zero, recompiling the code, and

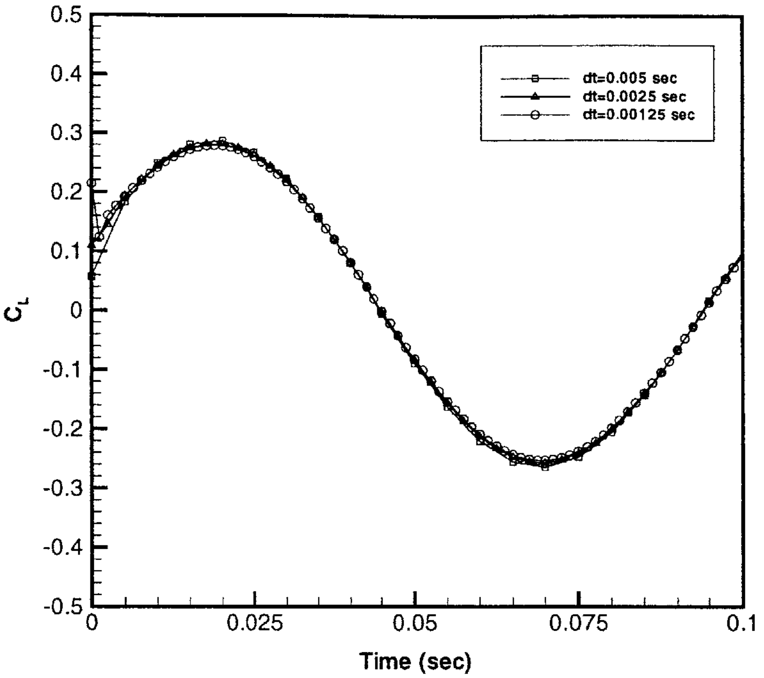


Figure 7-2: 10 Hz, 3° Pitch Response of the SHAR2 Wing from PMARC at Various Values of Δt

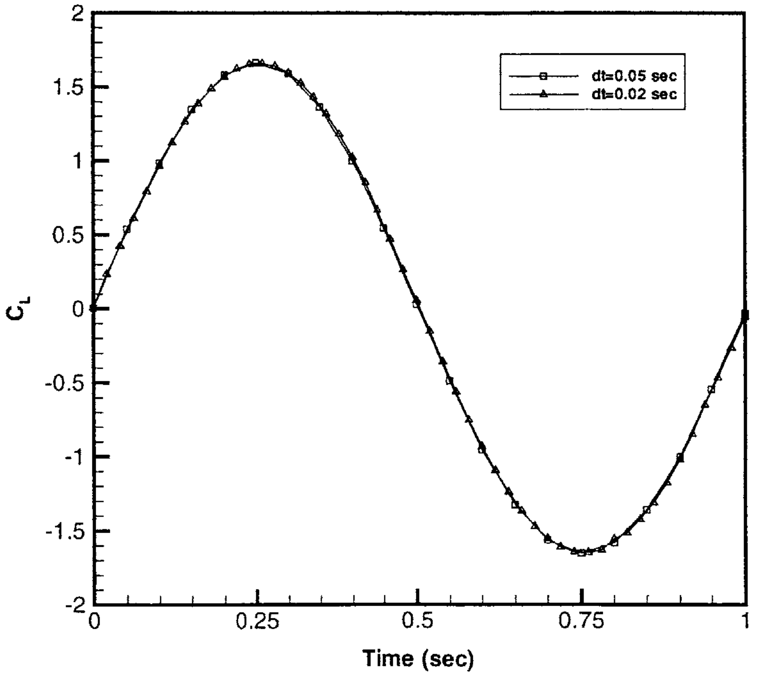


Figure 7-3: 1 Hz, 15° Pitch Response of the SHAR2 Wing from PMARC at Various Values of Δt

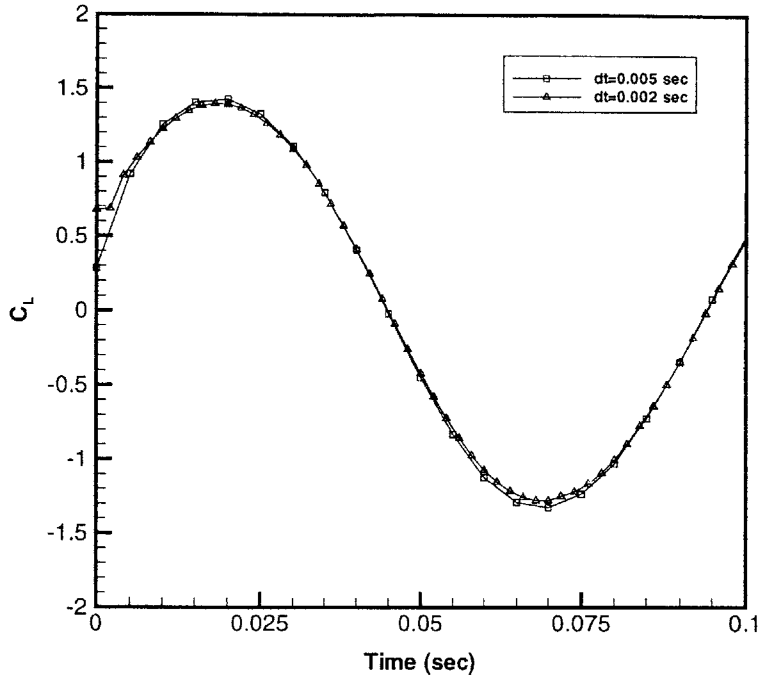


Figure 7-4: 10 Hz, 15° Pitch Response of the SHAR2 Wing from PMARC at Various Values of Δt

executing it with the same input file as previously. The same procedure was repeated for the pitch rate channel response. Figure 7-5 shows that adding the pitch response and the pitch rate response produces a response equivalent to the total response of the wing. (The deviations evident at $t \approx 0.045$ sec and $t \approx 0.055$ sec are due to PMARC not converging at those time steps. This happens occasionally and is the result of some numerical instability at that particular point.) Additional tests of 15° pitching confirm the same behavior at large oscillation amplitudes. This result confirms that the separation of input channels is valid and is properly implemented.

7.2.2 Nonplanar Wing Tests

The work in this thesis rests largely on the assumption that non-trivial aerodynamic effects result from large deflections of the lifting surface. A key early study was the aerodynamic behavior of rigid, nonplanar wings with PMARC (in its original, unmodified form).

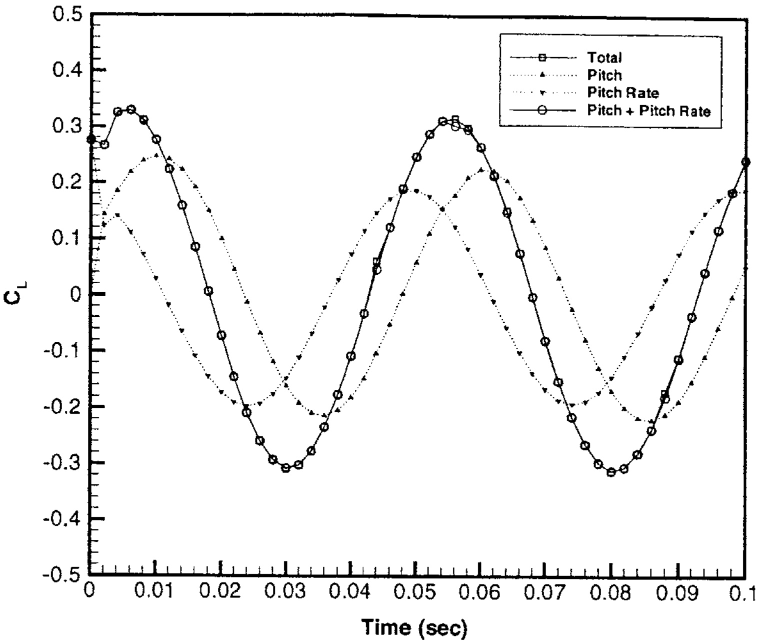


Figure 7-5: Pitch Response of the SHAR2 Wing with Separate and Combined Input Channels (20 Hz, 3°)

The first test was a steady PMARC analysis comparing the lift generated at various angles of attack of the SHAR1 Wing in its undeflected ("baseline") configuration to the lift generated with the wing deflected 29.3% of its semi-span. Two methods of simulating the deflected geometry were studied. The first uses the actual deflected wing geometry as generated by the wing generation program (referred to in subsequent figures as "Exact"). The second method uses the results of the baseline configuration analysis to extrapolate the lift generated by a deflected wing (referred to subsequently as "Extrapolated"). The process is illustrated in Figure 7-6. One of the output files of the wing generation program gives the translation and rotation of each span station on the deflected wing. Lift of the deflected wing can be extrapolated from the baseline analysis using this rotation data. If lift at each station is assumed to be normal to the surface, then the component of the lift in the vertical direction is simply

$$L_{ext} = L \cos\theta \tag{7.5}$$

where L is the sectional lift of the current span station and θ is the rotation of that span station due to the spanwise bending. The vertical lift component at all the span stations are summed for the total lift of the wing.

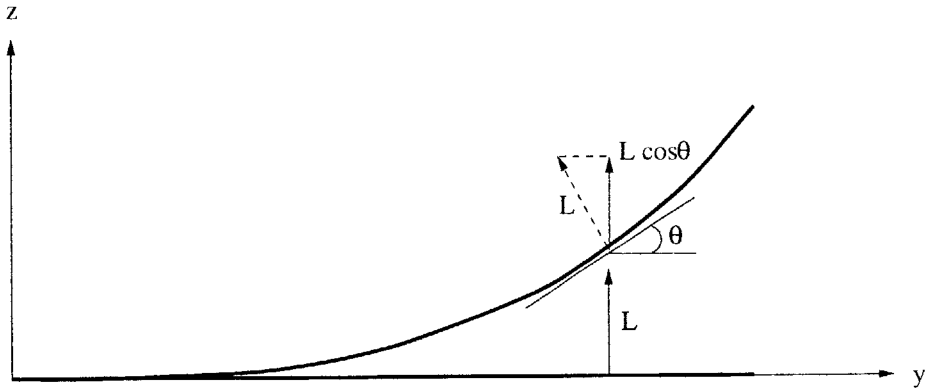


Figure 7-6: Diagram of the Lift Extrapolation Method

The results of this test are shown in Figure 7-7. It is evident that there is a significant lift loss associated with the deflection of the wing—the deflected configuration develops nearly 10% less lift than does the baseline configuration. It is thus clear that the wing deflection must be considered in the aerodynamic analysis. The extrapolation method described above seems to account for the lift loss well, particularly at low angle of attack.

The second of these preliminary tests considered the unsteady response of the rigid, highly-deflected wing. The responses of the SHAR1 Wing at pitching frequencies of 1 Hz, 10 Hz, and 20 Hz were considered; four bending configurations were used—0%, 29%, 37%, and 49%. Deflection was achieved with the same two methods as previously described, through generating a deformed geometry for the exact response and by extrapolating the response of a deformed shape from the response of the baseline, undeflected configuration. Figure 7-8 shows some representative results. It is clear from the figure that the extrapolation method breaks down as the frequency and the amount of deflection are increased. Table 7.1 summarizes the error analysis results of the complete set of test cases. The tabulated

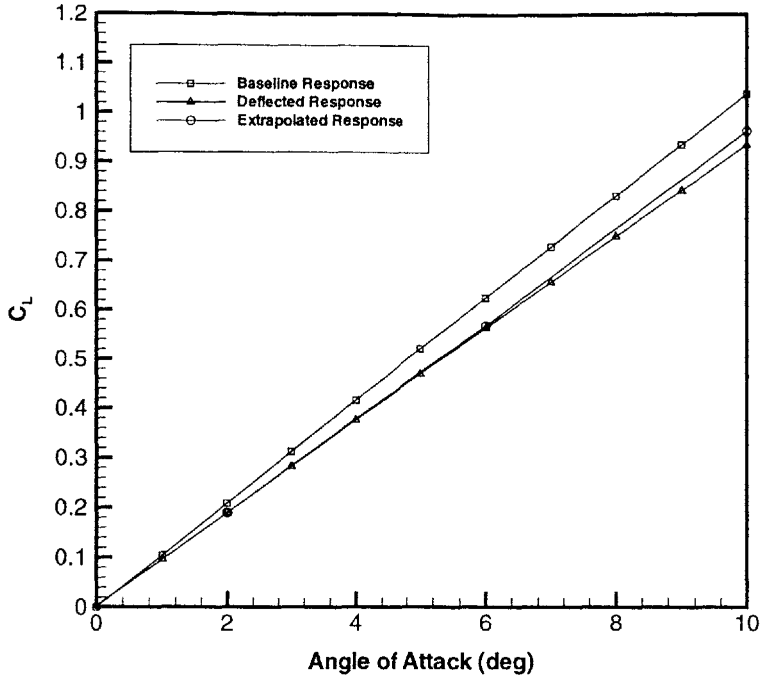
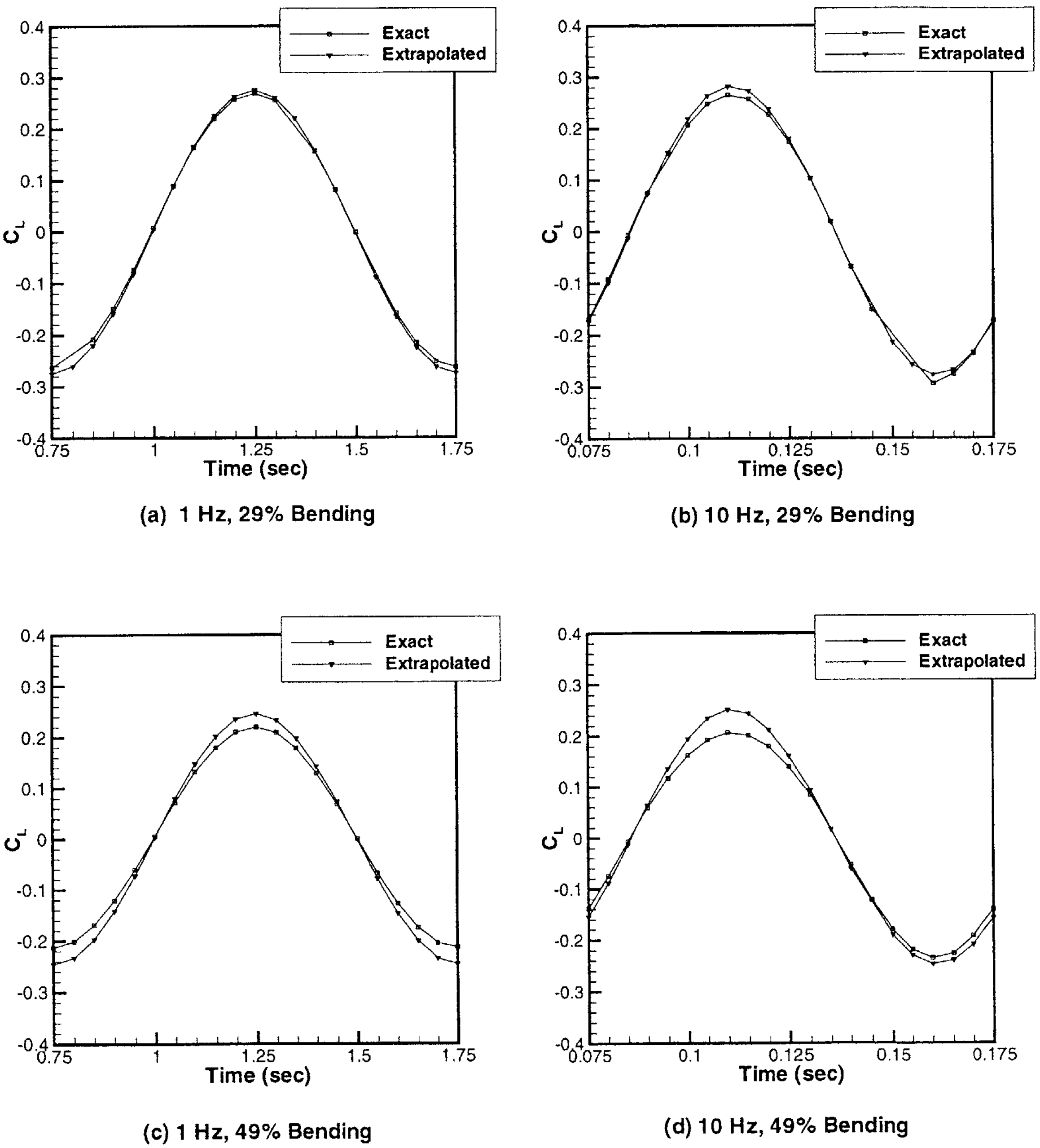


Figure 7-7: Lift of the SHAR1 Wing in Baseline and Deflected Configurations

data clearly shows the large deviation in amplitude—as high as 25%. In nearly all cases, phase differences are minimal. (The 1 Hz, 37% bending case seems to be anomalous.) These tabulated data consider only the primary frequency of the response; in some cases, the FFT analysis shows there to be other frequencies of lesser importance. Figures 7-9 and 7-10 show the full FFT results for the exact and extrapolated responses of two cases, 20 Hz pitching at 29% bending and 1 Hz pitching at 49% bending. In both cases there are clear differences between the exact and extrapolated results.

It is clear that the deflection of the wing must be accounted for in some manner. The extrapolation method works reasonably well for steady flows and unsteady flows with low frequency and low to moderate deflections. However, for general unsteady flows over flexible wings, a more accurate solution is desired. This solution is to be achieved by employing the Aerodynamic Impulse Response method with PMARC.



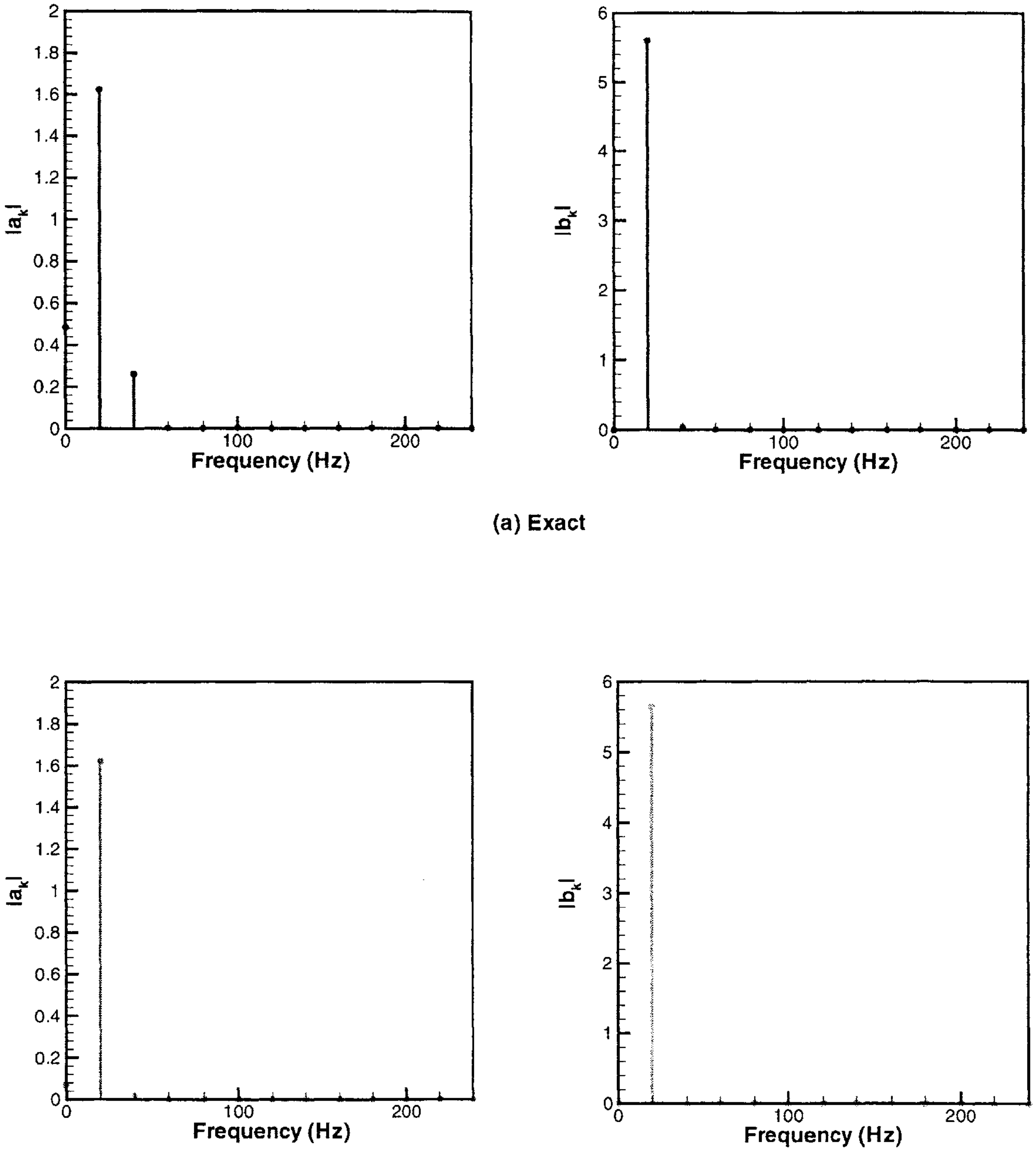
(a) 1 Hz, 29% Bending

(b) 10 Hz, 29% Bending

(c) 1 Hz, 49% Bending

(d) 10 Hz, 49% Bending

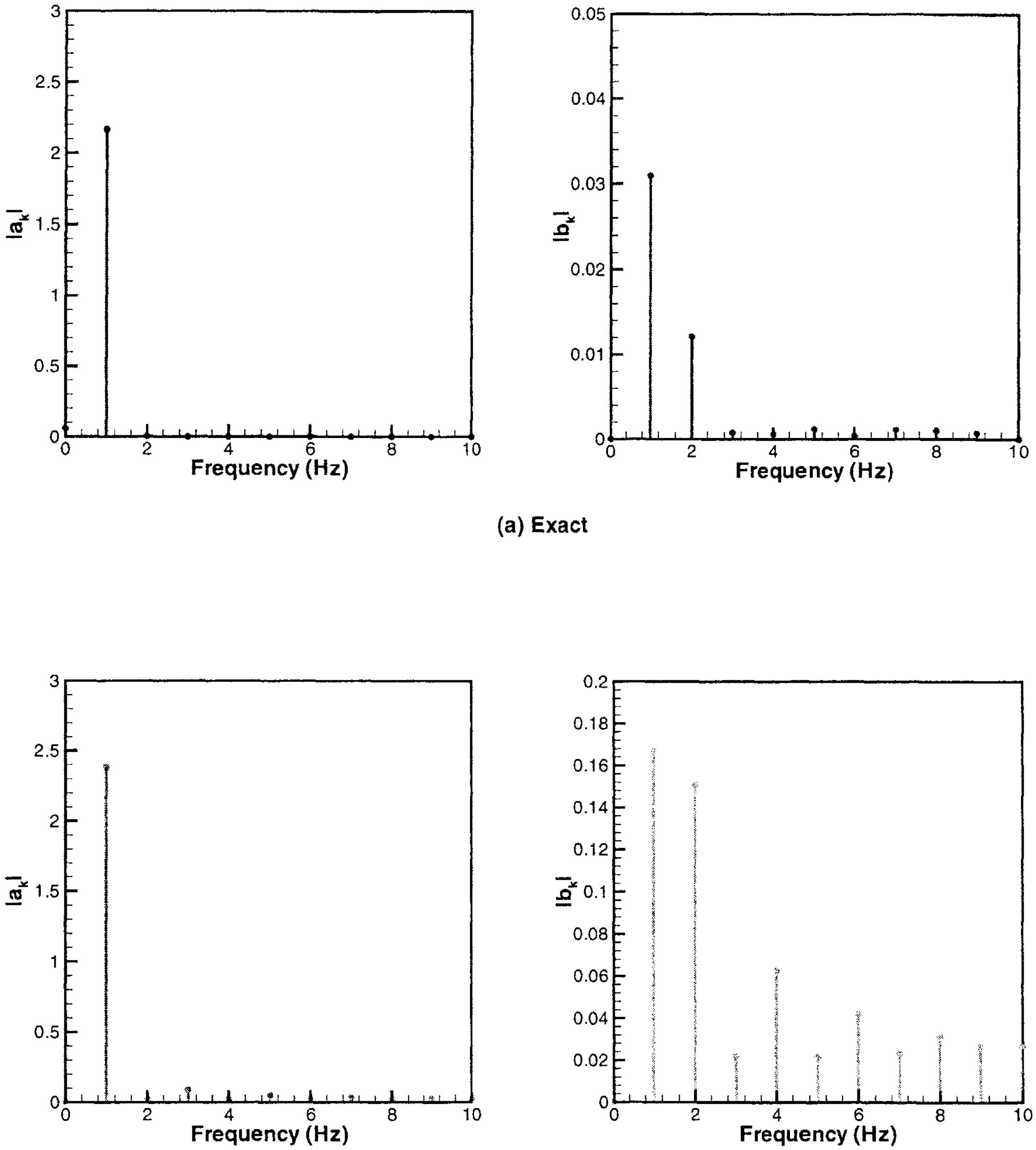
Figure 7-8: Unsteady Lift of the Deflected SHAR1 Wing Undergoing 3° Pitching Oscillations at Various Frequencies



(a) Exact

(b) Extrapolated

Figure 7-9: FFT Results of the Unsteady Lift Response of the Deflected SHAR1 Wing, 29% Bending, 20 Hz



(a) Exact

(b) Extrapolated

Figure 7-10: FFT Results of the Unsteady Lift Response of the Deflected SHAR1 Wing, 49% Bending, 1 Hz

Table 7.1: Error Results for Pitching of the Nonplanar SHAR1 Wing

Pitch Frequency	*% Bending*	*Amplitude Error*	*Phase Error*
1 Hz	0%	-2.14%	-0.0002 rad
1 Hz	29%	7.08%	-0.0166 rad
1 Hz	37%	25.5%	-0.4844 rad
1 Hz	49%	16.0%	0.0558 rad
10 Hz	0%	-1.45%	-0.0893 rad
10 Hz	29%	5.35%	0.0678 rad
10 Hz	37%	8.20%	0.0160 rad
10 Hz	49%	11.5%	-0.0700 rad
20 Hz	0%	-2.15%	0.0000 rad
20 Hz	29%	0.622%	0.0026 rad
20 Hz	37%	8.18%	0.0322 rad
20 Hz	49%	11.57%	0.0023 rad

7.3 Rigid-Body PMARC/AIR

The rigid-body form of the Aerodynamic Impulse Response method with PMARC was the first step towards an effective aerodynamics component for an aeroelastic analysis tool. The rigid-body analysis alone is insufficient for an aeroelastic analysis because the very nature of aeroelastic analysis is to study flexible bodies; however, the rigid-body formulation is an important stepping stone and offers significant contributions itself: principally, decreased computational costs and the extension of PMARC's unsteady capabilities to include arbitrary motions. Of all the validation studies of the Aerodynamic Impulse Response method, the bulk were performed with the rigid-body code, primarily because it can most easily be compared with control PMARC responses. Except where noted, all studies of the rigid-body formulation of PMARC/AIR were performed with the SHAR2 Wing per the specifications in Chapter 6.

7.3.1 Linear Impulse Responses

The fundamental result of the Aerodynamic Impulse Response method is the linear impulse response of a body. Given this impulse response, the body's response to any motion can quickly be determined. Impulse responses for the three PMARC input channels, pitch, pitch rate, and plunge rate, are shown in Figures 7-11 through 7-13 for $\Delta t = 0.005$ seconds. An important observation is that the impulse response *is* dependent on time step size. Figure 7-14, for instance, compares an impulse response of the pitch channel with $\Delta t = 0.005$ sec to one with $\Delta t = 0.05$ sec. The amplitude of the response is inversely proportional to the time step size. Experimentation indicates that as $\Delta t \to 0$, the response amplitude goes to infinity.

It is important to notice, however, that the variation of the impulse response with Δt has little effect on the convolution response, as will be seen in the following sections. Early in this reseach, a better defined impulse response curve was sought by reducing the time step size; however, it quickly became clear that no "converged" impulse response would be found. However, but for small inaccuracies, the convolution process eliminates apparent differences in the impulse responses; seemingly quite different impulse responses, when convolved with the same input function (note, though, that the time step size of the input function must match that of the impulse response), produce the same response.

It is also evident from the impulse responses that this system is one with very little memory, because the responses die quickly to zero. The rapid damping to zero of the response is one of the advantageous features of the method. PMARC computes the impulse response much more quickly than it does a sinusoidal unsteady response—198.05 CPU seconds were required on an Intel Pentium II for twenty time steps of the sinusoidal response, while twenty time steps of the impulse response were computed in 81.8 CPU seconds. In addition to the computational benefits, the nonlinear kernel should only require several components for this system because of its short memory.

7.3.2 Linear Responses to Sinusoidal Motions

Because of the ease of generating control data using the built-in PMARC unsteady capabilities, the bulk of the testing of the PMARC/AIR implementation was performed for sinusoidal pitching and plunging motions. The robustness of the method was scrutinized by varying the following parameters:

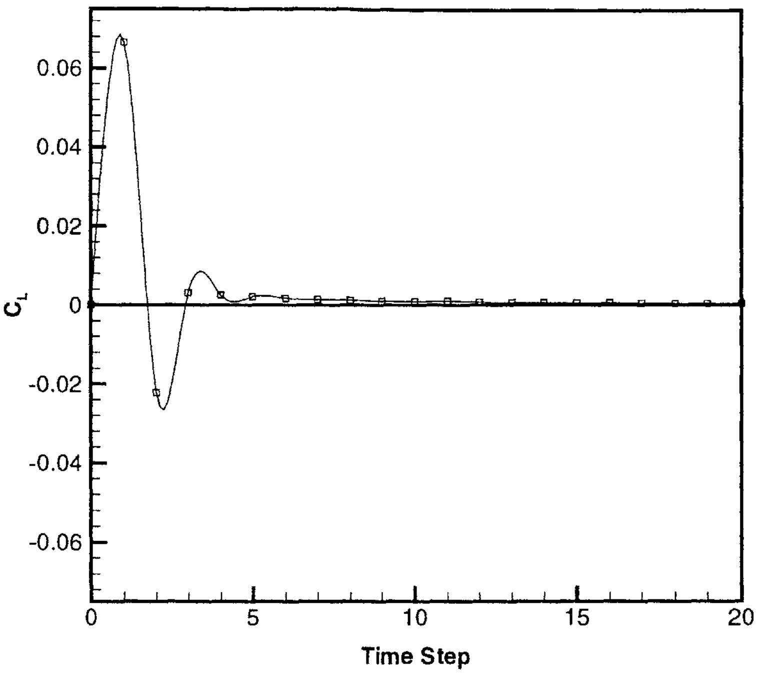


Figure 7-11: Pitch Channel Impulse Response with $\Delta t = 0.005$ sec, SHAR2 Wing

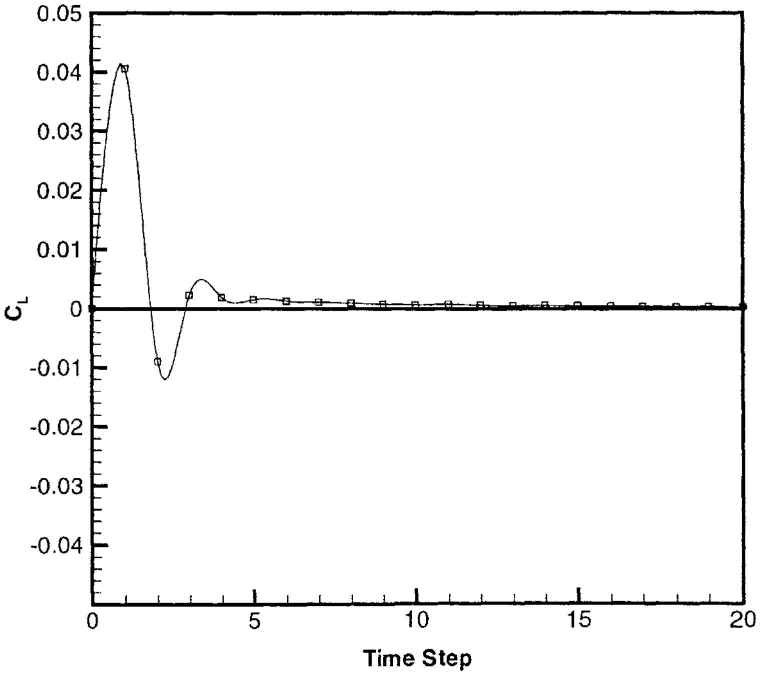


Figure 7-12: Pitch Rate Channel Impulse Response with $\Delta t = 0.005$ sec, SHAR2 Wing

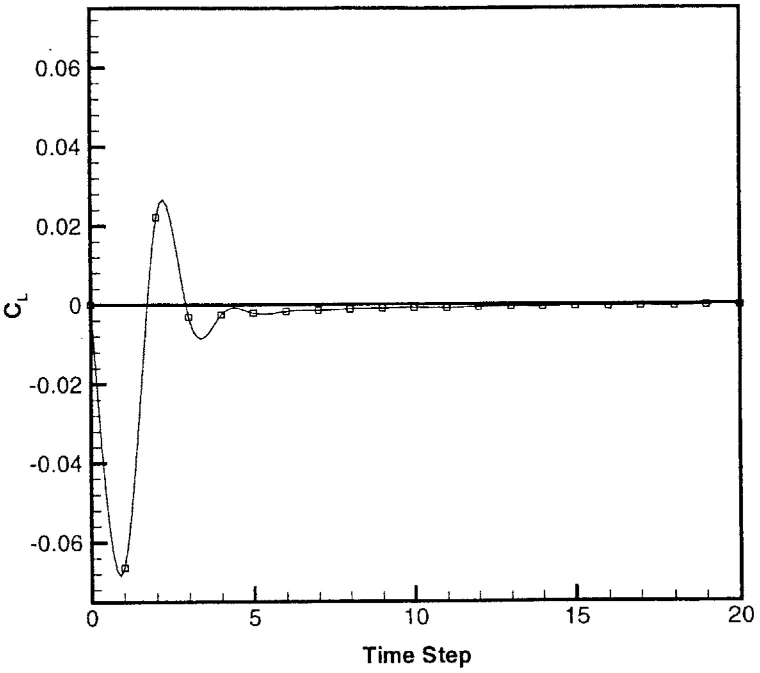


Figure 7-13: Plunge Rate Channel Impulse Response with $\Delta t = 0.005$ sec, SHAR2 Wing

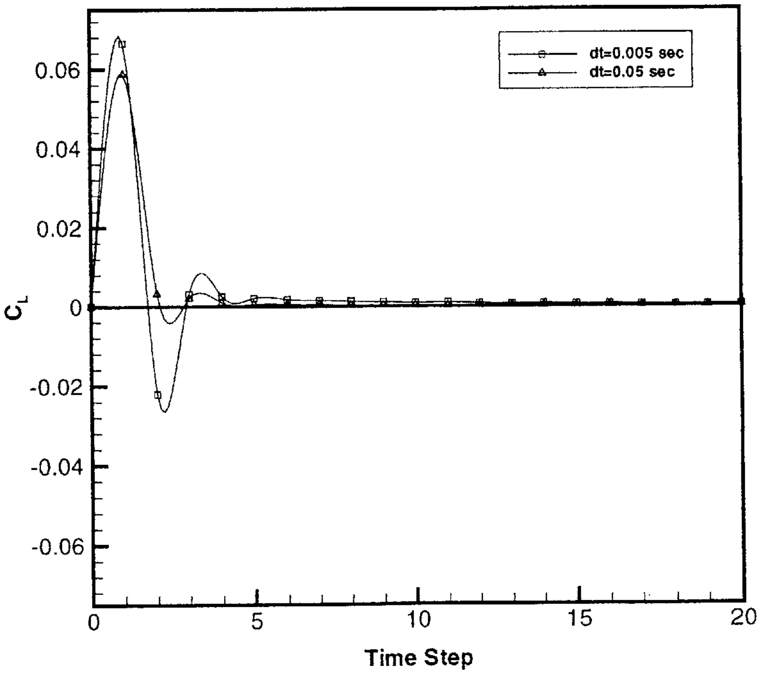


Figure 7-14: Pitch Channel Impulse Response with $\Delta t = 0.005$ sec and $\Delta t = 0.05$ sec, SHAR2 Wing

- Input function amplitude
- Input function frequency, f
- Impulse response time step size, $\Delta t_{impulse}$
- Freestream velocity magnitude
- Wing aspect ratio

Pitch and Plunge Amplitude

If PMARC were purely linear, the unit impulse response would be sufficient to predict the response to *any* input. However, as has already been seen, there is some nonlinearity and the AIR responses are expected to deviate from the PMARC responses at high amplitudes. The results presented in this section compare PMARC control data to responses generated with the AIR method for 10 Hz pitching with amplitudes varying from 1° to 10° and for 10 Hz plunging with amplitudes varying from 0.1 ft to 1.0 ft—that is, 0.1 to 1.0 chord lengths. (Note that the plunge amplitude is related to the plunge rate amplitude by the oscillation frequency, so the maximum plunge rate for this case is 62.83 ft/s. Plunge amplitude rather than plunge rate amplitude is referred to throughout for two reasons: first, the built-in PMARC unsteady routine takes the plunge amplitude as an input, and secondly, it is more physically meaningful than plunge rate.) Pitch channel results are shown in Figures 7-15 through 7-18. The pitch channel exhibits little amplitude dependence within this range; however, 10° is still well within the linear small angle range. The deviation apparent for the 1° amplitude case at 0.14 seconds are the result of difficulty in obtaining a converged solution at several time steps with PMARC.

Plunge rate channel results are shown in Figures 7-19 through 7-22. A noticable deviation from the control data appears in the plunge rate channel response beginning with a plunge amplitude of 0.5 ft, and the deviation becomes significant for the 1.0 ft amplitude case. For both of these cases, the plunge amplitude is on the order of the wing chord-length. This suggests that the chord-length is a good reference of the

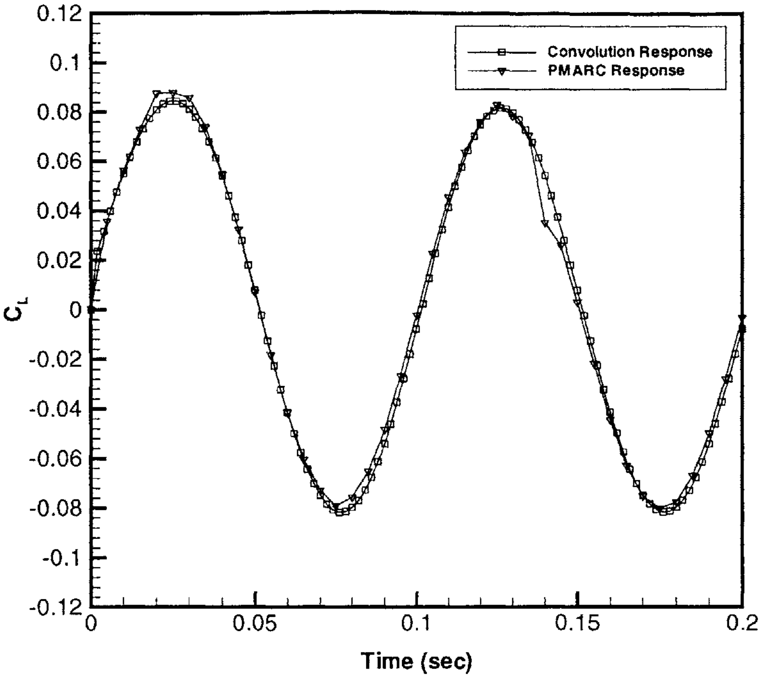


Figure 7-15: AIR and PMARC Pitch Channel Responses of the SHAR2 Wing for 10 Hz Sinusoidal Pitching Oscillations with Amplitude 1°

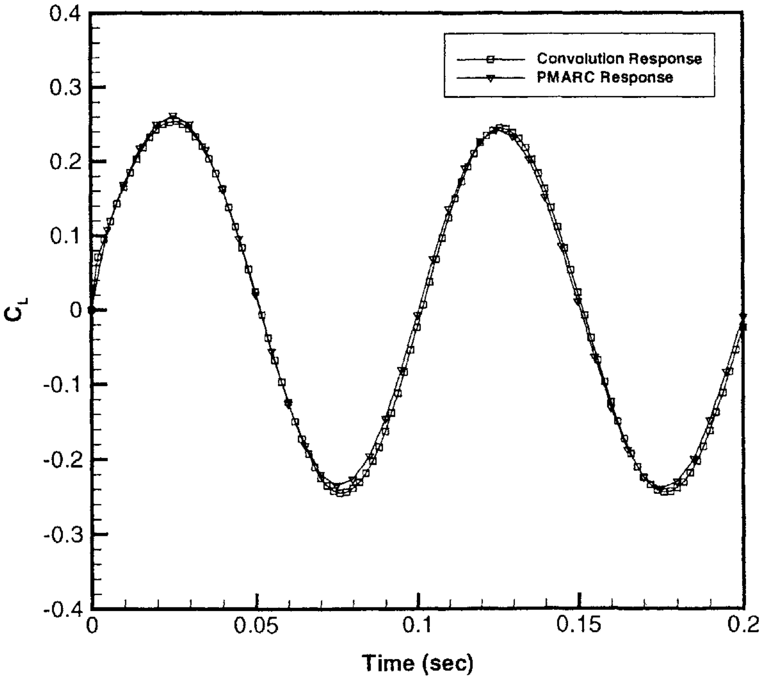


Figure 7-16: AIR and PMARC Pitch Channel Responses of the SHAR2 Wing for 10 Hz Sinusoidal Pitching Oscillations with Amplitude 3°

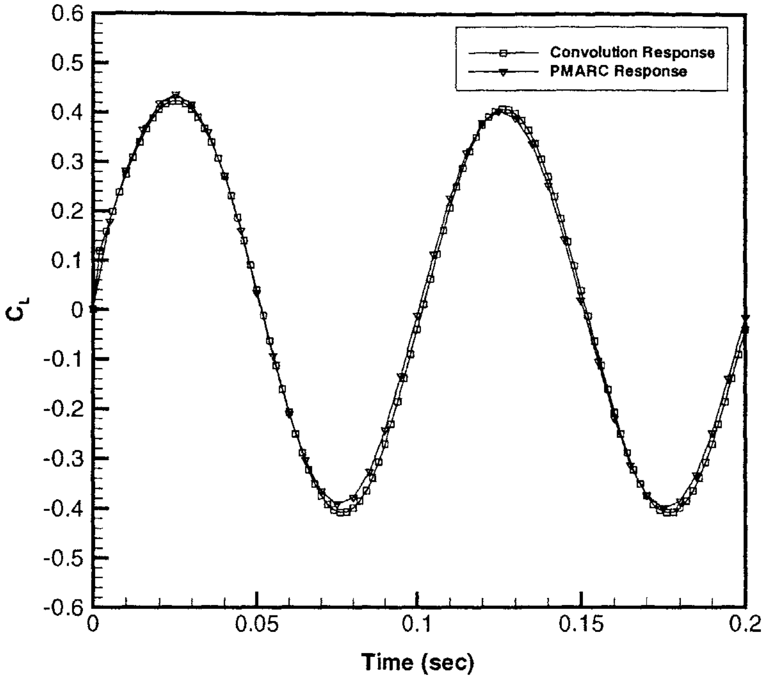


Figure 7-17: AIR and PMARC Pitch Channel Responses of the SHAR2 Wing for 10 Hz Sinusoidal Pitching Oscillations with Amplitude 5°

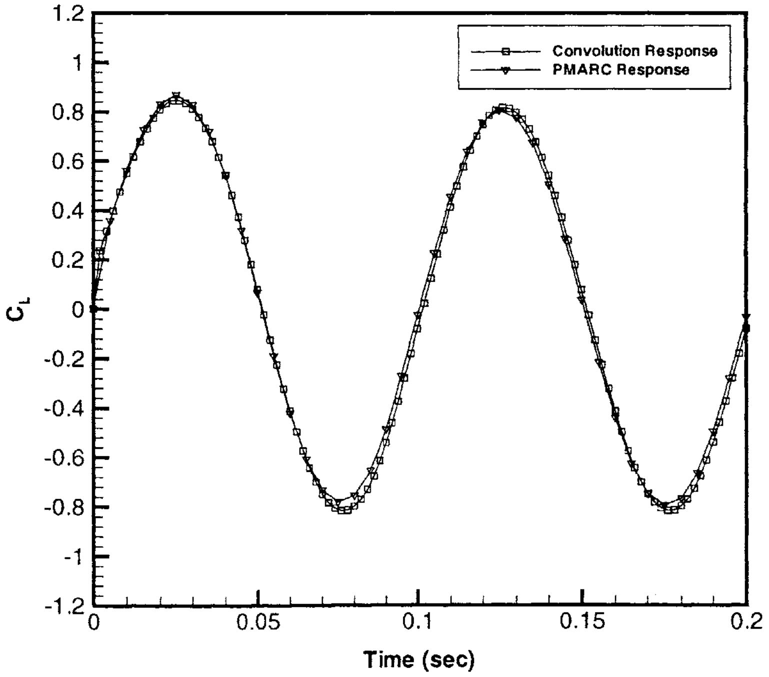


Figure 7-18: AIR and PMARC Pitch Channel Responses of the SHAR2 Wing for 10 Hz Sinusoidal Pitching Oscillations with Amplitude 10°

boundary between large- and small-amplitude motions. This case is repeated with a nonlinear analysis in Section 7.3.6, which shows the difference here to be attributable to nonlinearity of the solution.

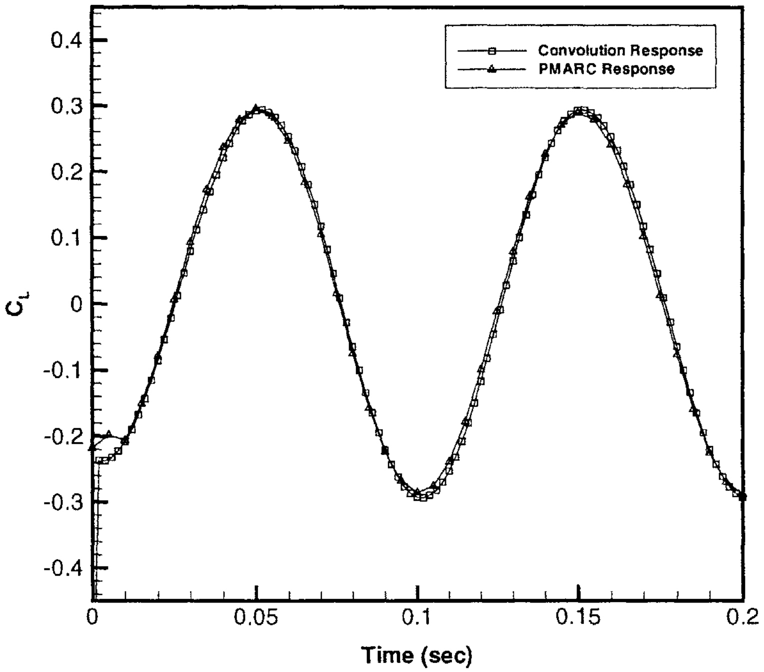


Figure 7-19: AIR and PMARC Plunge Rate Channel Responses of the SHAR2 Wing for 10 Hz Sinusoidal Plunging Oscillations with Amplitude 0.1 ft

Pitching Frequency

The pitch and pitch rate channel responses were studied with sinusoidal input functions of varying frequencies. A single impulse response for each input channel was generated with PMARC/AIR with $\Delta t_{impulse} = 0.002$ seconds, then sinusoidal responses were generated by convolving the impulse response with three sinusoidal input functions, each with an amplitude of 3° and with frequencies 1 Hz, 10 Hz, and 20 Hz. Results for the pitch channel are shown in Figure 7-23, and those for the pitch rate channel are shown in Figure 7-24. It appears that the frequency of the input function does have an effect on the accuracy of the predicted response. Interestingly, there does not appear to be a consistent trend to this behavior—while there is a

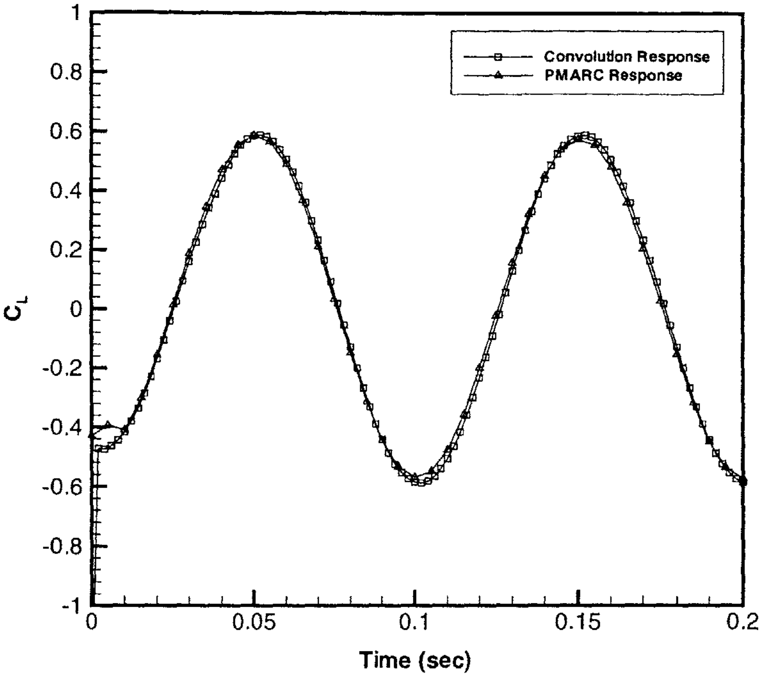


Figure 7-20: AIR and PMARC Plunge Rate Channel Responses of the SHAR2 Wing for 10 Hz Sinusoidal Plunging Oscillations with Amplitude 0.2 ft

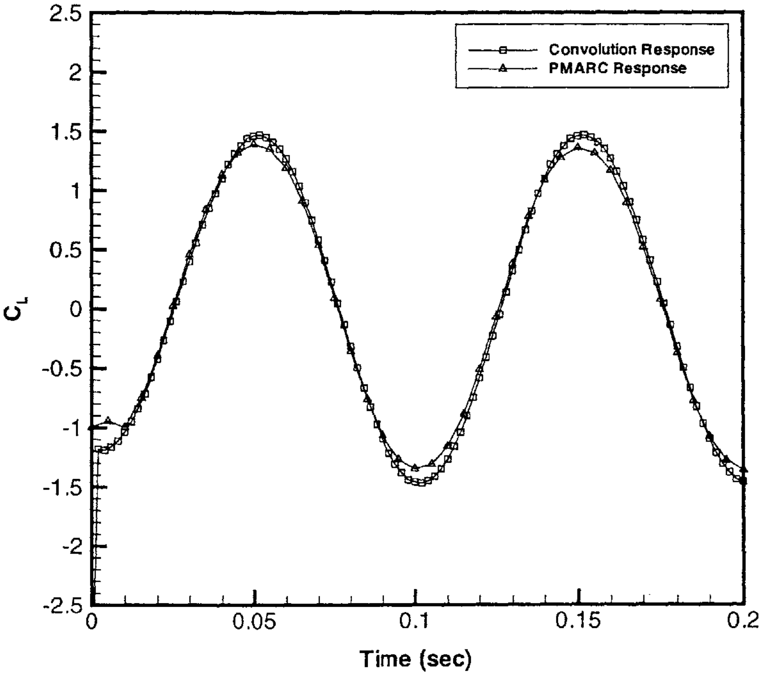


Figure 7-21: AIR and PMARC Plunge Rate Channel Responses of the SHAR2 Wing for 10 Hz Sinusoidal Plunging Oscillations with Amplitude 0.5 ft

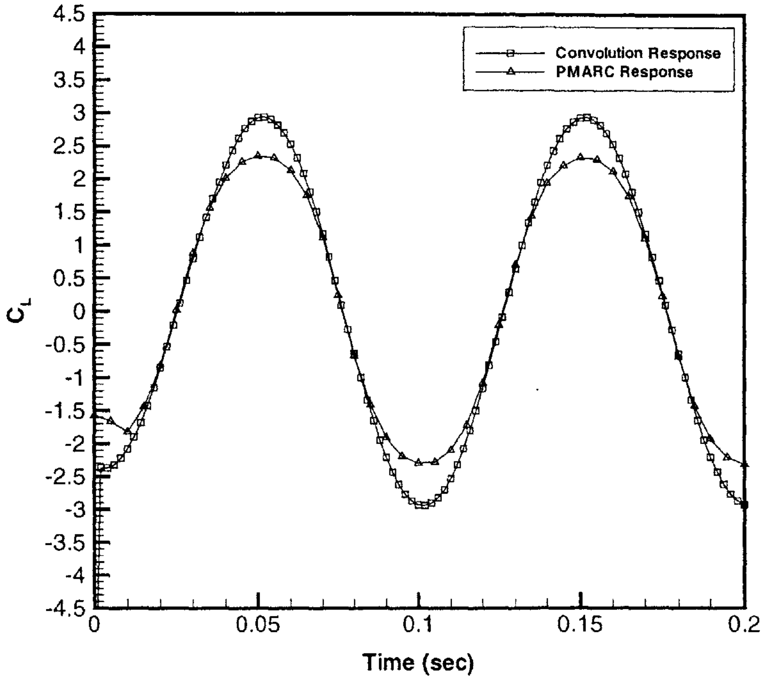
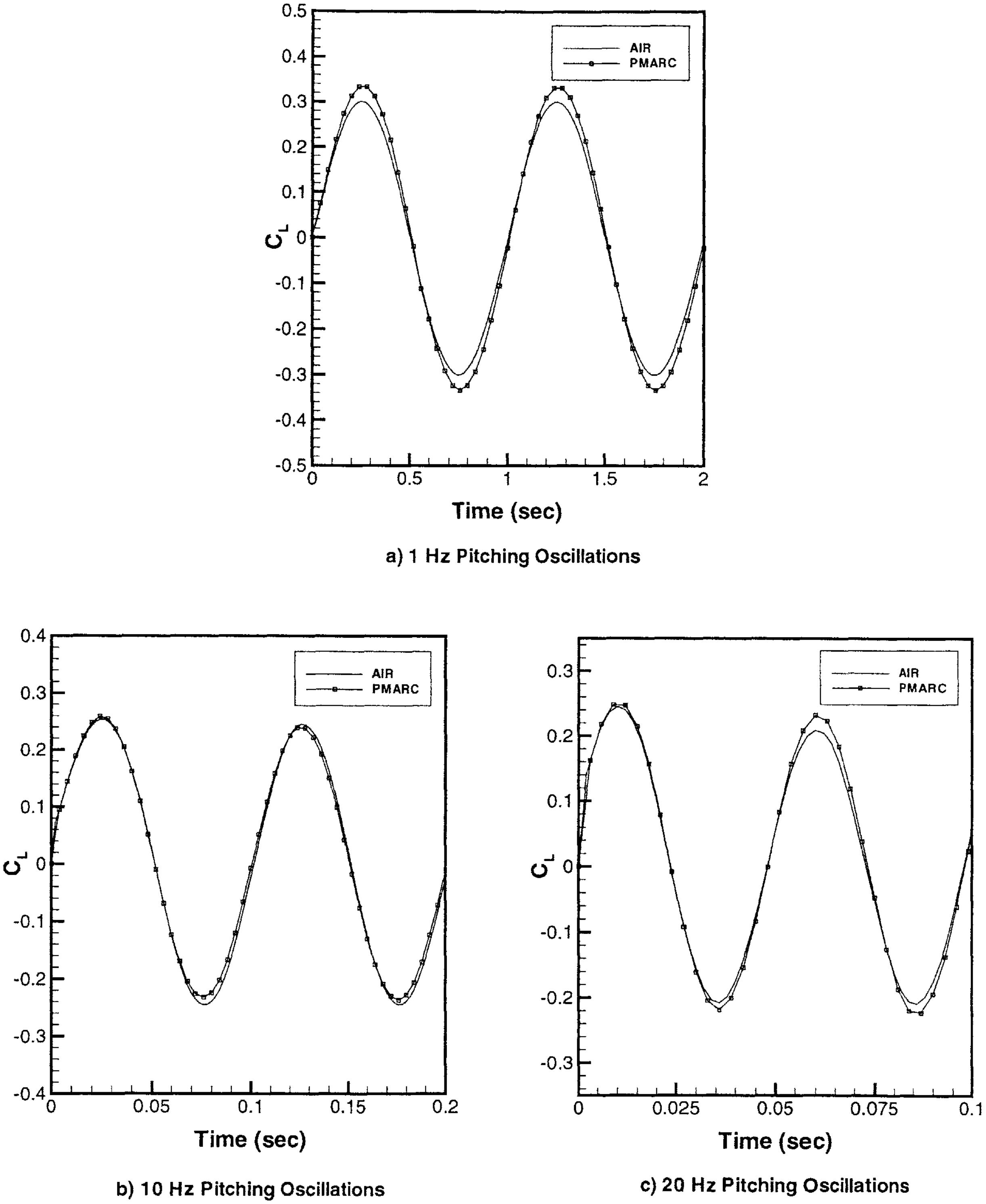


Figure 7-22: AIR and PMARC Plunge Rate Channel Responses of the SHAR2 Wing for 10 Hz Sinusoidal Plunging Oscillations with Amplitude 1.0 ft

close fit for the 10 Hz case, the AIR method significantly underpredicts the response amplitude for both 1 Hz and 20 Hz cases. This behavior is exhibited in both the pitch and pitch rate channel responses. The plunge rate channel was not rigorously studied but was observed to have similar behavior. Since the time step size was held fixed at 0.002 seconds for all frequencies, some relationship between time step choice and excitation frequency seems to be exhibited. This will be studied further in the following section.

The magnitude of the error for each time step choice was quantified using the FFT method described in Section 7.1. Table 7.2 summarizes the FFT analysis results for this test. As one can see from the table, the match is reasonable for both amplitude and phase. As will be seen in the following section, error can be minimized by properly tuning the time step size to the excitation frequency. Here, a constant time step size is utilized for each frequency. As will be seen, the time step size, $\Delta t_{impulse} = 0.002$ sec, is actually too small for all cases studied here. Examination of the full FFT results show no significant contribution due to additional frequencies.

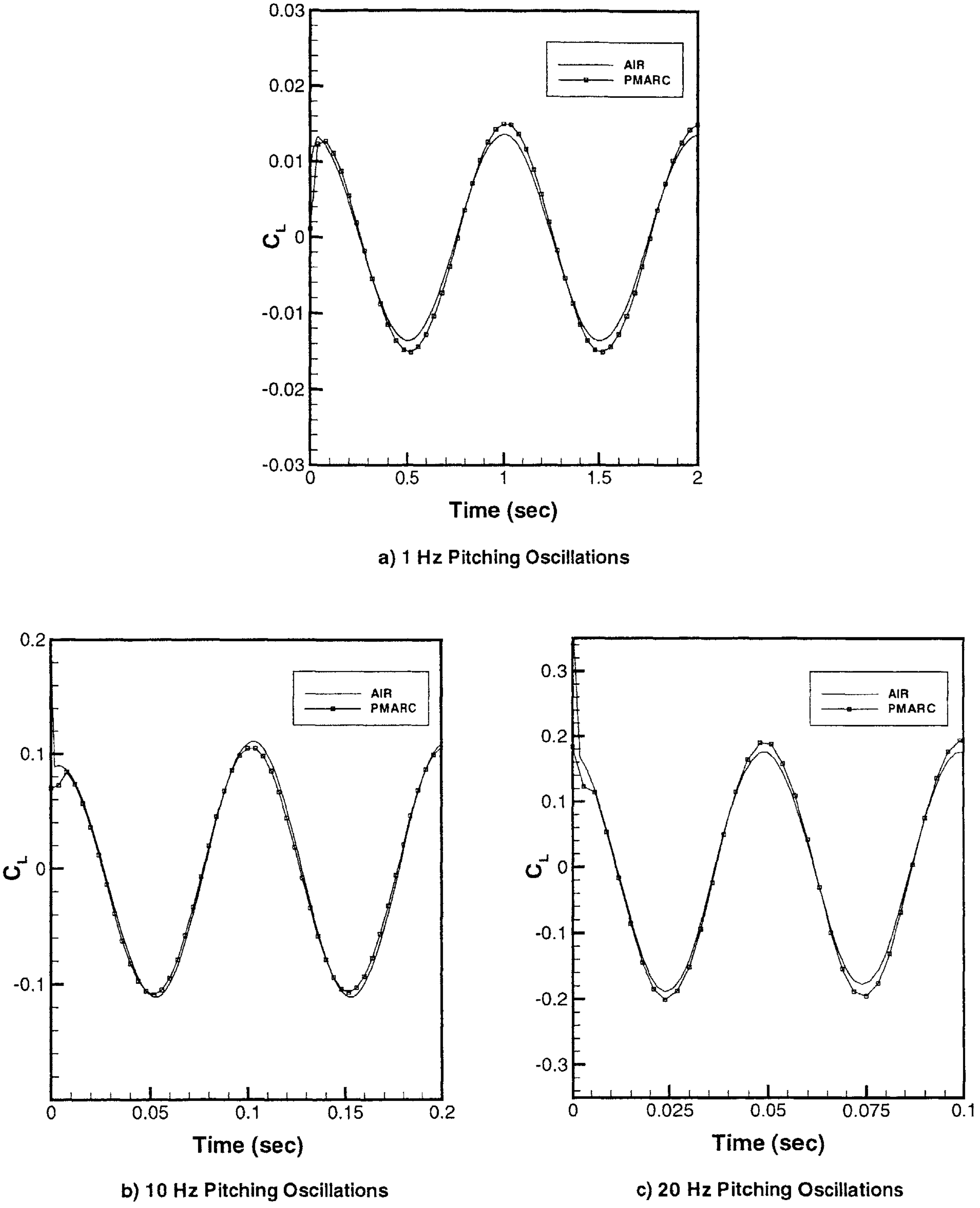


a) 1 Hz Pitching Oscillations

b) 10 Hz Pitching Oscillations

c) 20 Hz Pitching Oscillations

Figure 7-23: Pitch Channel Responses of the SHAR2 Wing for Sinusoidal Pitch Input at 1 Hz, 10 Hz, and 20 Hz with Amplitude 3° and $\Delta t_{impulse} = 0.002$ sec



a) 1 Hz Pitching Oscillations

b) 10 Hz Pitching Oscillations

c) 20 Hz Pitching Oscillations

Figure 7-24: Pitch Rate Channel Responses of the SHAR2 Wing for Sinusoidal Pitch Input at 1 Hz, 10 Hz, and 20 Hz with Amplitude 3° and $\Delta t_{impulse} = 0.002$ sec

Table 7.2: Error Results for 3° Pitching at Various Frequencies Using the FFT Method

Channel	*Frequency*	*Amplitude Error*	*Phase Error*
Pitch	1 Hz	-10.06%	0.0471 rad
Pitch	10 Hz	2.63%	-0.0467 rad
Pitch	20 Hz	-7.79%	-0.0521 rad
Pitch Rate	1 Hz	-9.93%	0.0579 rad
Pitch Rate	10 Hz	4.13%	0.0189 rad
Pitch Rate	20 Hz	-9.88%	0.0418 rad

Time Step Size

To further study the deviation of the AIR method results from the PMARC responses that was noted in the previous section, a similar test was carried out, only this time the frequency was held constant at 1 Hz and the time step size used for the generation of the impulse response, $\Delta t_{impulse}$, was varied. The input amplitude was again 3°. The plots in Figures 7-25 and 7-26 show representative results of this test. Time steps range from 0.002 seconds to 0.1 seconds. Here it is evident that, in general, accuracy increases with time step size. There is a significant amplitude error for the smallest time step, while there is no discernable difference between the control data and the AIR method predicted response at $\Delta t_{impulse} \geq 0.05$ seconds. Table 7.3 summarizes the FFT error analysis of these results. A large degree of uncertainty was found in the computation of phase angles, and the result was heavily dependent on the starting point of the signal. For the cases marked with (*), accuracy of the result cannot be assured because the starting point of the AIR response could not be matched to that of the PMARC response. Regardless, a trend is evident: in general, accuracy of the result increases as the time step size increases.

Given the results of varying the time step size and the frequency independently, there seems to be a relationship between the two. Results clearly show that a large time step is desirable. Combining the results of the two tests, it is postulated that best accuracy can be achieved by selecting a time step such that $f\Delta t_{impulse} \sim 0.05$. Figures 7-27 through 7-29 show pitch, pitch rate, and plunge rate channel responses for sinusoidal pitching and plunging at 1 Hz, 10 Hz, and 25 Hz with $\Delta t_{impulse} =$

Table 7.3: Amplitude Error for 3°, 1 Hz Pitching at Various Time Steps Using the FFT Method

Time Step Size	*Amplitude Error*	*Phase Error*
0.002 s	-10.07%	0.0471 rad
0.003 s	-5.51%	0.0321 rad
0.004 s	-2.52%	0.0195 rad
0.01 s	3.00%	-0.0239 rad
0.02 s	-0.12%	-0.0502 rad
0.03 s	-3.12%	-0.0330 rad (*)
0.04 s	-1.17%	-0.0048 rad
0.05 s	-0.39%	0.0044 rad
0.06 s	0.45%	-0.0660 rad (*)
1.0 s	0.21%	0.0188 rad

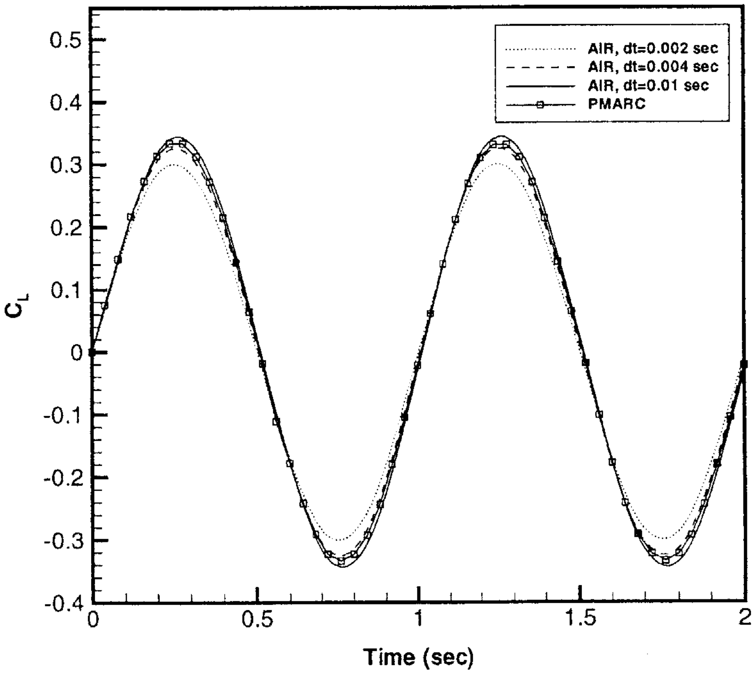


Figure 7-25: Pitch Channel Responses of the SHAR2 Wing for Sinusoidal Pitching Input at 1 Hz with Amplitude 3° and Varying Time Steps

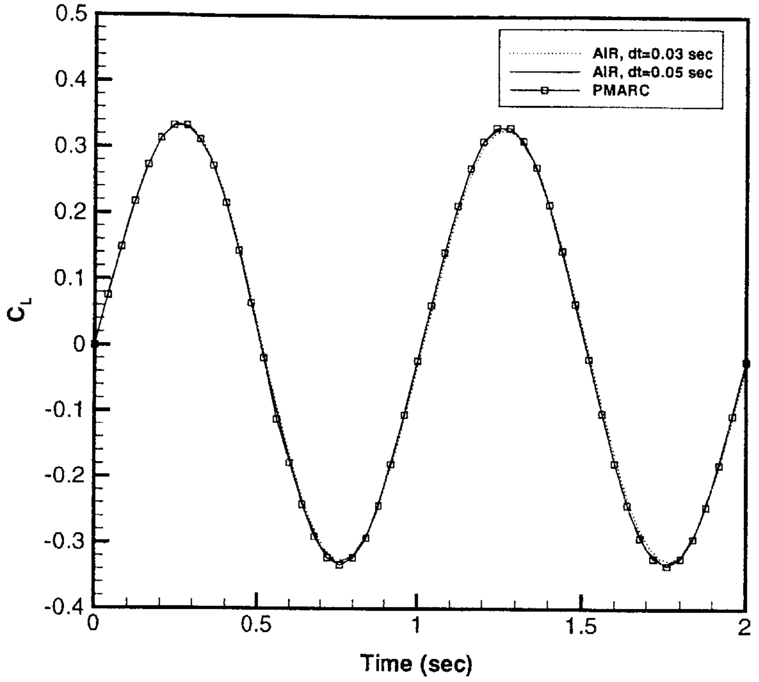
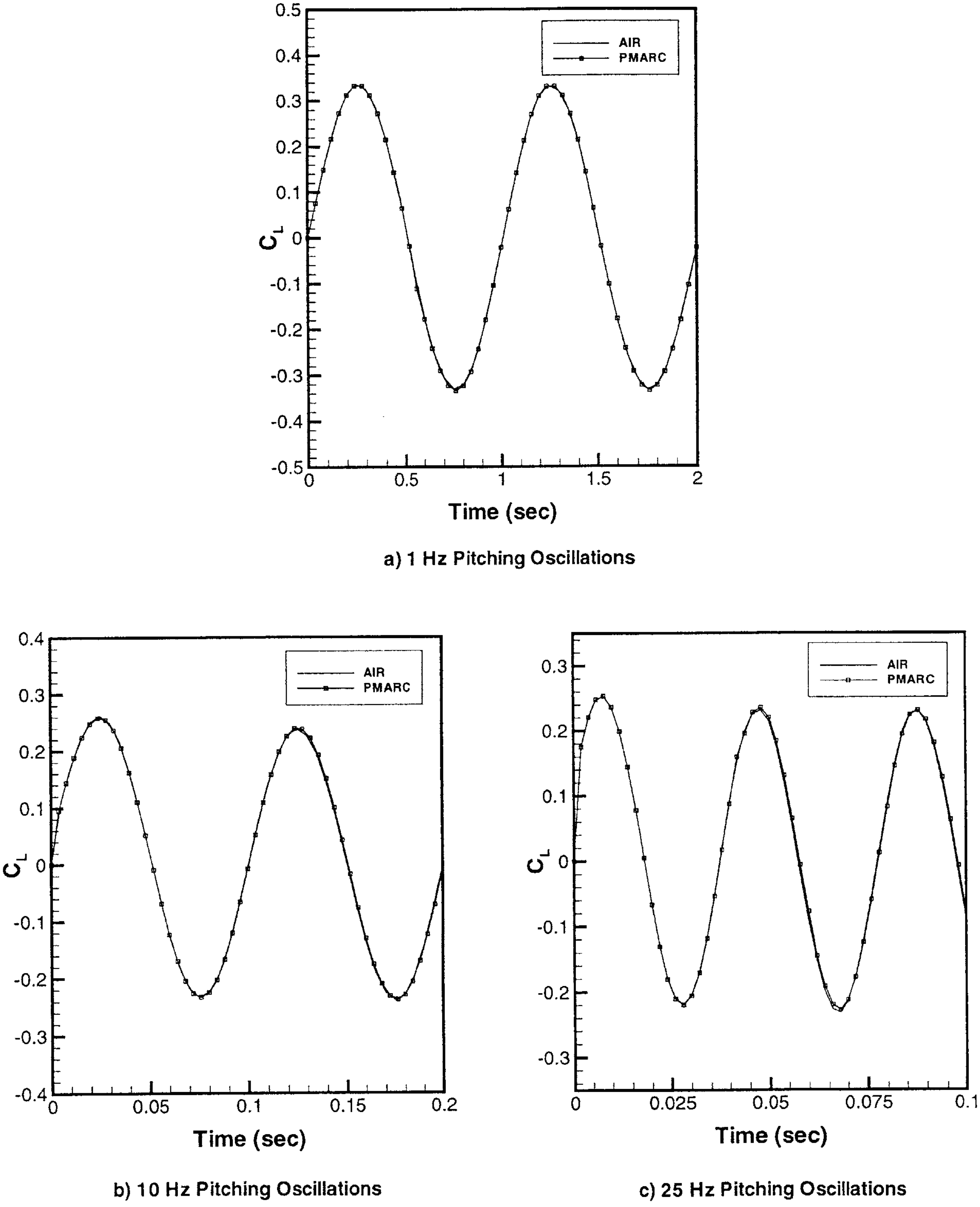


Figure 7-26: Pitch Channel Responses of the SHAR2 Wing for Sinusoidal Pitching Input at 1 Hz with Amplitude 3° and Varying Time Steps (continued)

0.05, 0.005, and 0.002 seconds, respectively. In nearly all cases a close fit between the control data from PMARC and the responses using the AIR method is found; only the high frequency pitch rate and plunge rate responses show any discernable error. However, regardless of the close match that was obtained, putting these results into a broader perspective, this behavior limits the method's applicability. Though near perfect accuracy can be assured for a specific time step-frequency combination, only adequate accuracy can be promised for a different frequency at the same time step. Since it is desired to divorce the arbitrary response analysis from the aerodynamic code by obtaining one impulse response from which all other responses can be derived, this is a potentially severe limitation. This can clearly be seen when the input is not sinusoidal, and the $f\Delta t$ factor becomes ambiguous because there is not a single frequency upon which to base the time step choice.

A better understanding of the time step–frequency dependence of the predicted response is desired so as to improve the applicability of the method to general cases. It is believed that the late-time behavior of the impulse response may be responsible

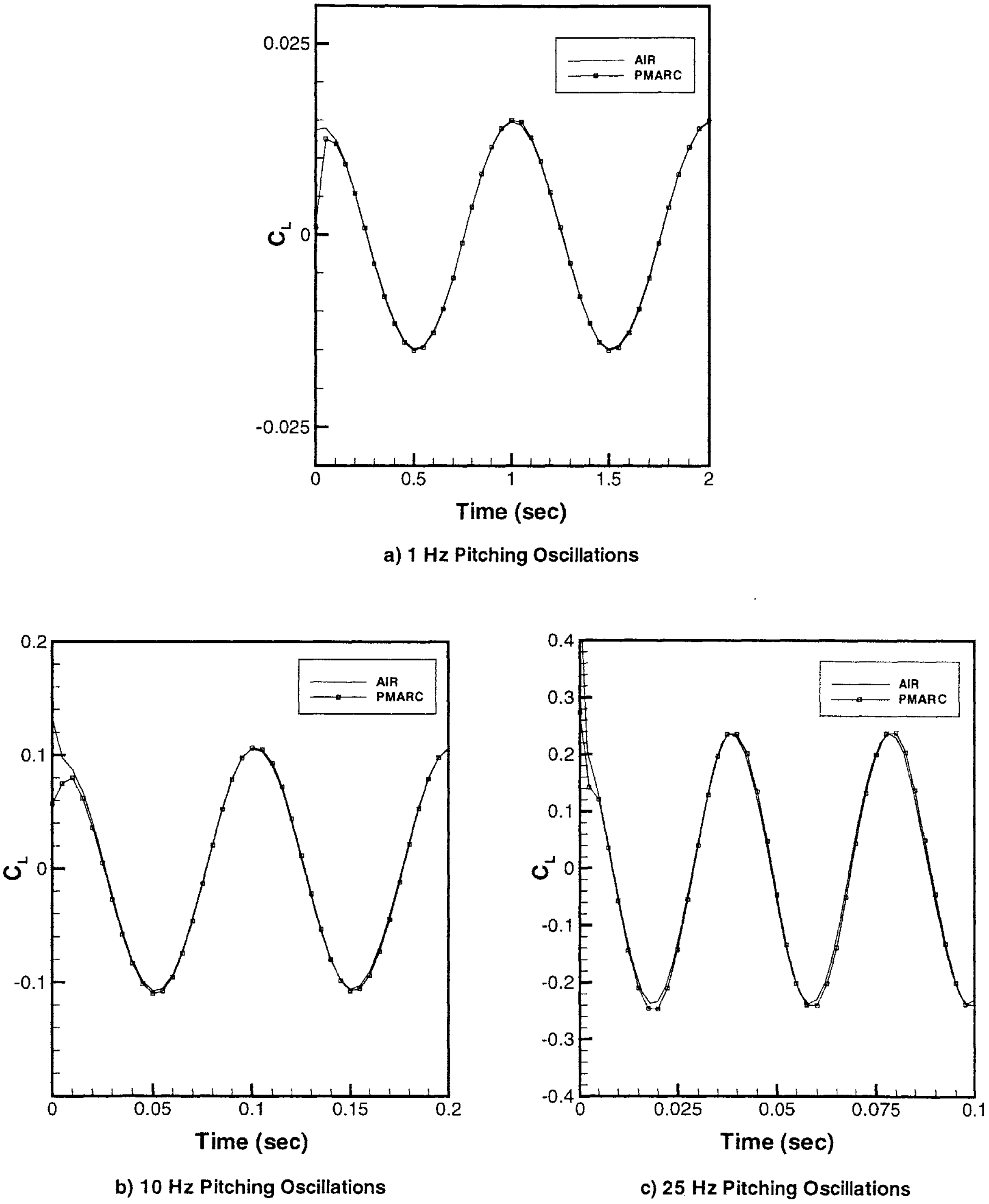


a) 1 Hz Pitching Oscillations

b) 10 Hz Pitching Oscillations

c) 25 Hz Pitching Oscillations

Figure 7-27: Pitch Channel Responses of the SHAR2 Wing for Sinusoidal Pitch Input at 1 Hz, 10 Hz, and 25 Hz with Amplitude 3° and $f\Delta t_{impulse} = 0.05$

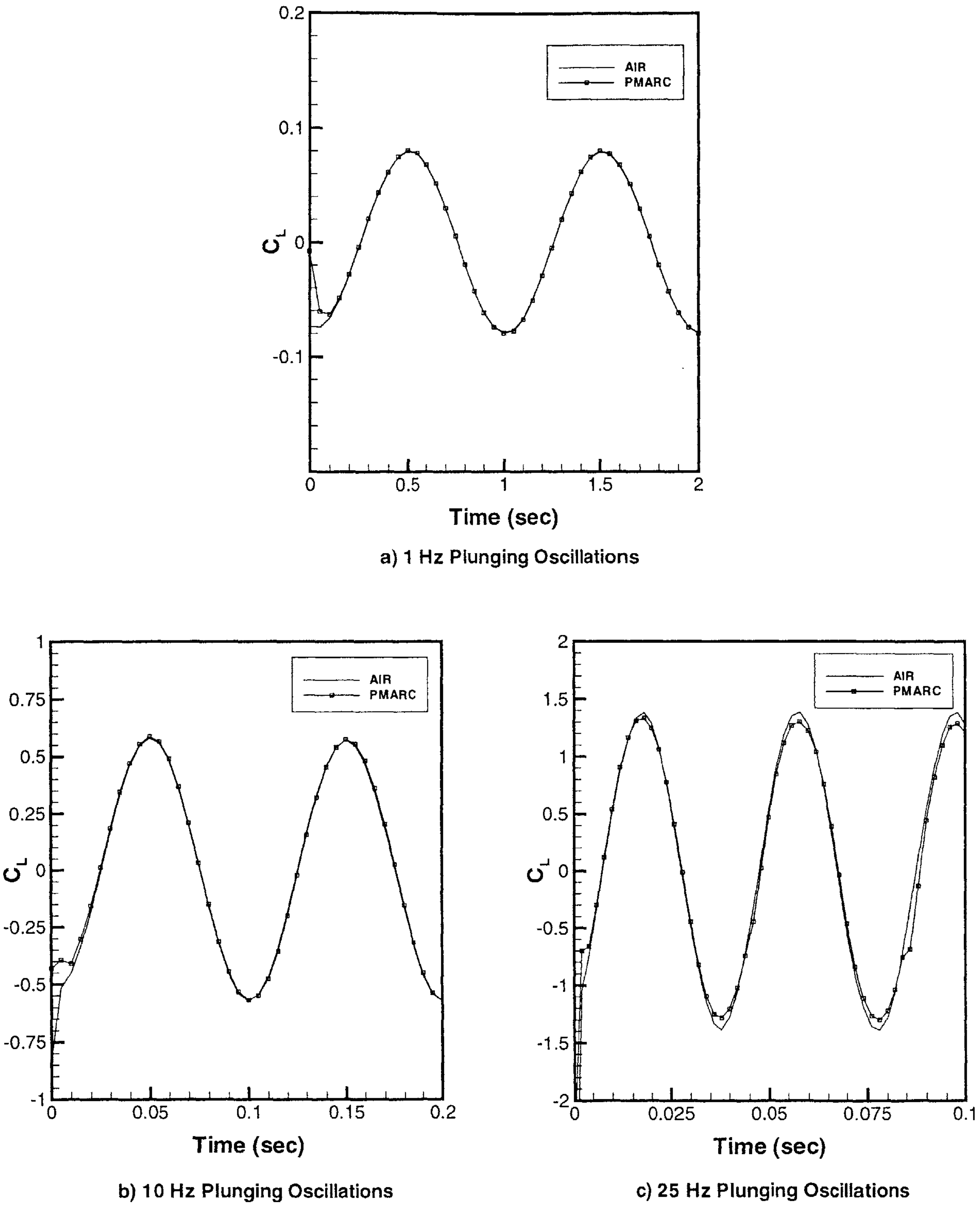


a) 1 Hz Pitching Oscillations

b) 10 Hz Pitching Oscillations

c) 25 Hz Pitching Oscillations

Figure 7-28: Pitch Rate Channel of the SHAR2 Wing Responses for Sinusoidal Pitch Input at 1 Hz, 10 Hz, and 25 Hz with Amplitude 3° and $f\Delta t_{impulse} = 0.05$



a) 1 Hz Plunging Oscillations

b) 10 Hz Plunging Oscillations

c) 25 Hz Plunging Oscillations

Figure 7-29: Plunge Channel Responses of the SHAR2 Wing for Sinusoidal Plunging Input at 1 Hz, 10 Hz, and 25 Hz with Amplitude 0.2 ft and $f\Delta t_{impulse} = 0.05$

for the error that is seen in these results. Close study of the impulse responses shows that the response does not go to exactly zero but to a small, non-zero value (roughly two to three orders of magnitude smaller than the initial response). The superposition of many such responses as occurs in the convolution process at a small time step may cause a residual large enough to affect the overall response. The significant portion of the response seems to occur before the fourth time step. Figure 7-30 compares control data from PMARC with responses generated from impulse responses truncated at various points. A response generated with an impulse response retaining the residual for twenty time steps, as has been used in all previous analyses, is also shown and is seen to give the best fit to the control data. Consequently, further study will have to be directed towards understanding the frequency-time step size interaction to improve the accuracy of the method.

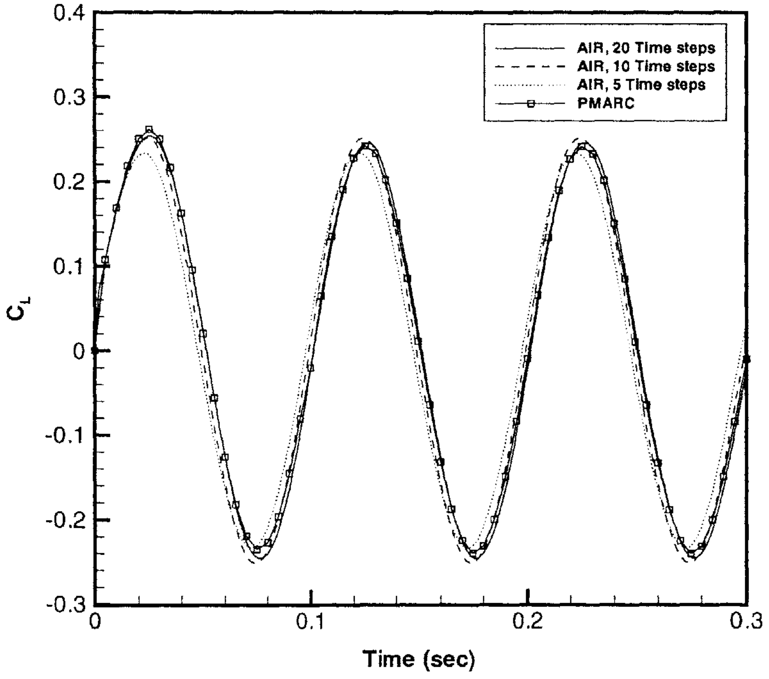


Figure 7-30: Comparison of Various Pitch Channel Impulse Response Lengths for 10 Hz, 3° Pitching, $\Delta t_{impulse} = 0.002$ seconds

Freestream Velocity

The default freestream velocity used to obtain the impulse response of the SHAR2 Wing is 100 ft/s. The question then stands whether this impulse response can be used for arbitrary responses at any velocity, if a simple scaling factor for other velocities exists, or if it can only be used for arbitrary responses also at a freestream velocity of 100 ft/s. If the flow were steady, the response should be the same regardless of the freestream since the lift coefficient is independent of velocity; however, the unsteadiness of the flow introduces rate-dependent effects. As Figure 7-31 clearly shows, results using the AIR method incur a loss of accuracy when the freestream velocity of the impulse response does not match the freestream velocity of the input function. For this test, two pitch channel impulse responses were generated, one with a freestream velocity of 100 ft/s, the other at 50 ft/s. Both impulse responses were convolved with the same input function, 5 Hz sinusoidal pitching with amplitude 3° and $\Delta t = 0.002$ seconds, and the results were compared to the PMARC response for the same unsteady conditions with a freestream flow at 50 ft/s. Though the impulse response at 50 ft/s does not produce a perfect result (as seen earlier, greater accuracy would be achieved by choosing $\Delta t_{impulse} = 0.01$ seconds), the impulse response at 100 ft/s generates a worse response, and there is not clear scaling factor between the responses. It is thus apparent that impulse responses are only strictly valid for responses at the same freestream conditions.

Aspect Ratio: the BACT Wing

The pitch channel response of the BACT Wing, with an aspect ratio of 2, was studied to see if any wing aspect ratio dependence appears in the results. Figure 7-32 shows the response of the BACT Wing undergoing 10 Hz, 3° amplitude pitching oscillations as predicted by PMARC and by the impulse response method. The time step used was 0.005 seconds, satisfying the $f\Delta t_{impulse}$ factor discussed above. These results show no evidence of any aspect ratio dependence of the AIR method: the method appears to be as accurate for low-aspect ratio wings as it is for high-aspect ratio

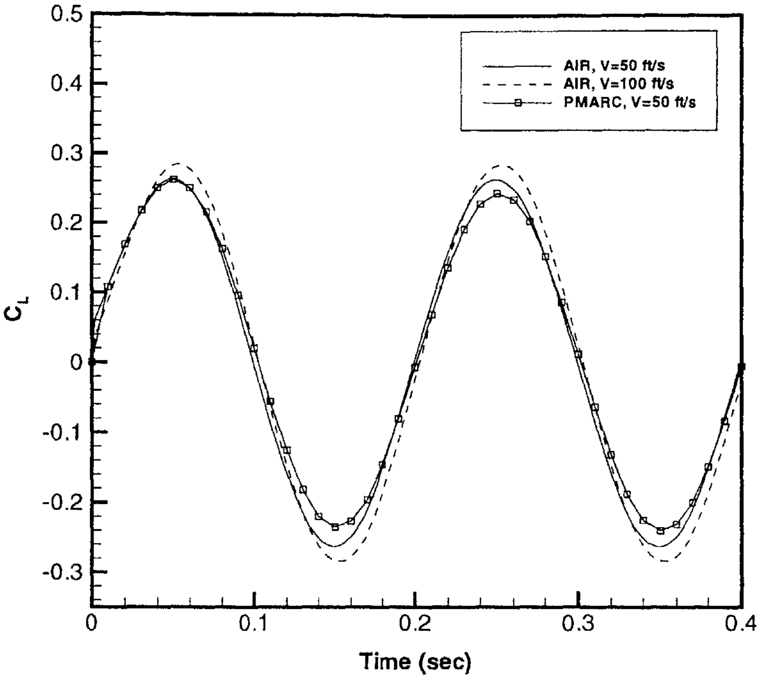


Figure 7-31: Pitch Channel Response of the SHAR2 Wing to 5 Hz, 3° Pitching with Impulse Responses Computed at 50 ft/s and 100 ft/s, $\Delta t_{impulse} = 0.002$ seconds

wings. A distinguishable error does appear at the bottom for the first cycle of the response; however, this error disappears in subsequent valleys.

7.3.3 Linear Response to Step Input

Useful input types for the validation of the AIR method are limited to those that can be duplicated in PMARC to provide control data. Sinusoidal inputs are simplest because the capability is built into PMARC. In addition, a step input response can be studied with a simple modification to the PMARC/AIR code. For a step change in pitch, the pitch channel impulse amplitude, THETA, is changed so that instead of being 0.01 rad at $t = t_0$ and 0.0 for all other times, it is set to a non-zero step amplitude for $t \geq t_0$. For the total response, the derivative must be considered as well. This is approximated by setting the pitch rate amplitude, Q, to the step amplitude divided by the time step size at $t = t_0$ and zero for for all other times. (The derivative of a discrete-time step function is a discrete impulse function.) However, this study

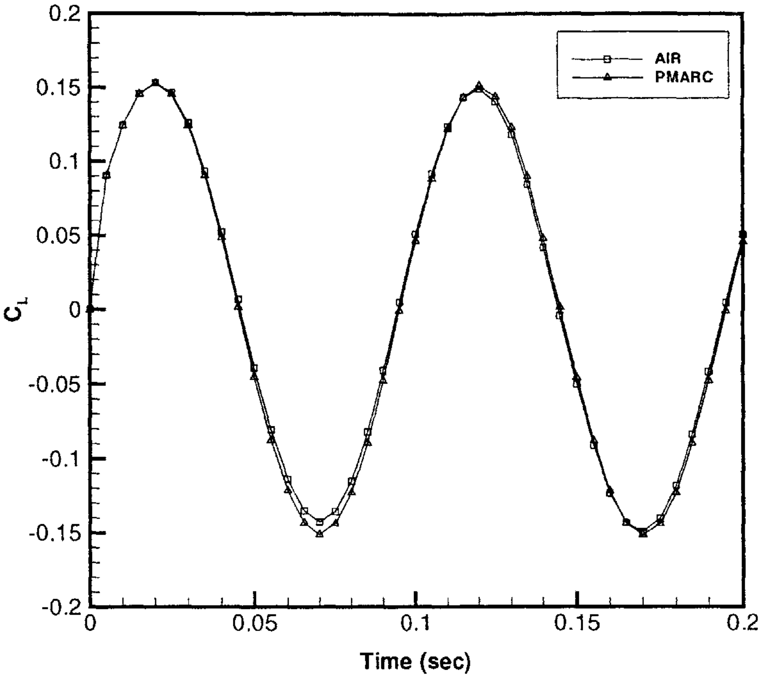


Figure 7-32: Pitch Channel Response for 10 Hz, 3° Pitching Oscillations of the BACT Wing, $\Delta t_{impulse} = 0.005$ seconds

only looks at the pitch channel response, so pitch rate channel is left at zero for all times.

Figure 7-33 shows the response generated by this modified code and the response predicted using the pitch channel impulse response with $\Delta t_{impulse}$ =0.05 seconds. The step amplitude is 0.1 radians, or 5.73°. The response predicted using the AIR method is fairly accurate; however, there is a noticeable error that increases as time increases. This is likely an artifact of the non-zero residual of the impulse response, in which case the error should asymptote to a constant value. For the response shown here, the error at $t = 1$ second should be near this constant value because the impulse response was truncated after twenty time steps, and $t = 1$ second is at the twentieth time step of the response. Further study into this residual response will be necessary to improve the accuracy of the method.

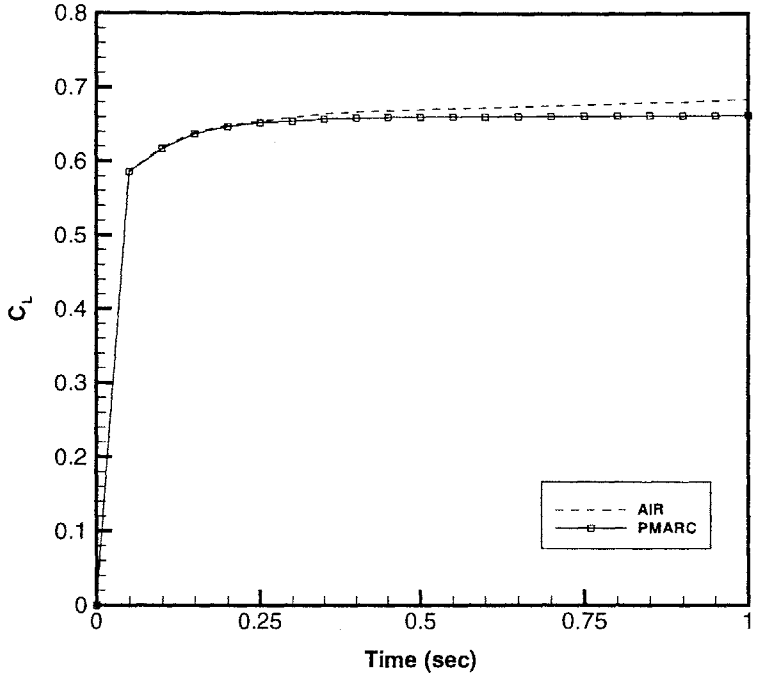


Figure 7-33: Pitch Channel Responses of the SHAR2 Wing to 0.1 rad (5.73°) Step Input with $\Delta t = 0.05$ Seconds

7.3.4 Linear Response to Arbitrary Input

A particularly useful feature of PMARC/AIR is the ability to predict the response to an arbitrary input. For the purpose of testing this feature, a Fourier sine series with three components, as is given in Equation 7.6 and is shown in Figure 7-34, was used. This results in an input signal with three frequencies: 1 Hz, 2 Hz, and 3 Hz. This input function was selected because modification of the PMARC code to generate control data was relatively simple—additional frequency terms had simply to be added to the unsteady terms in the PATH subroutine.

$$\theta[t] = 3^\circ \sin(2\pi t) + 3^\circ \sin(4\pi t) + 3^\circ \sin(6\pi t) \tag{7.6}$$

The pitch channel response to this pitch input is shown in Figure 7-35. The response was calculated with the AIR method using impulse responses generated with three time steps, $\Delta t_{impulse} = 0.01$ seconds, 0.015 seconds, and 0.02 seconds, and with the modified PMARC code at $\Delta t = 0.015$ seconds. All three time step choices

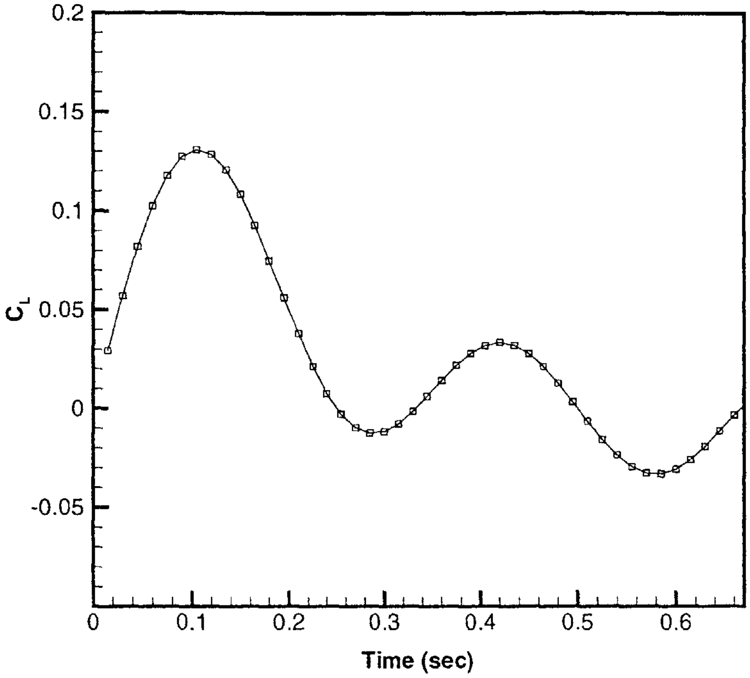


Figure 7-34: Arbitrary Pitch Input

produce a fairly accurate prediction of the response. The time step–frequency factor, as was discussed in Section 7.3.2, is ambiguous in this case, as it was for the step input, because multiple frequencies are involved. Typically, the highest frequency restricts the choice of time step. In this case, if 3 Hz is used in the time step–frequency factor, an optimal time step of $\Delta t \sim 0.0167$ seconds is predicted. Surprisingly, in this case the optimal results are found with the smallest of the time steps, suggesting that the time step-frequency factor may have little significance for arbitrary responses.

Regardless of the the time step choice ambiguity, it is clear from Figure 7-35 that the AIR method predicts the response to this arbitrary input fairly reliably. These results indicate that the method is a useful one for predicting the response to arbitrary inputs that cannot so easily be modeled in PMARC, making a significant improvement over the capabilities of PMARC alone.

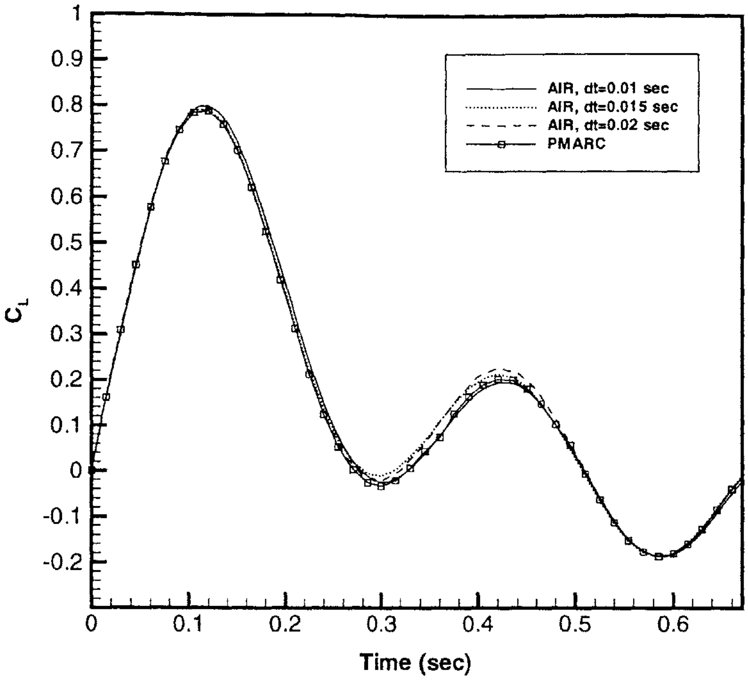


Figure 7-35: Pitch Channel Responses of the SHAR2 Wing to Arbitrary Pitching Motions given in Figure 7-34

7.3.5 Nonlinear Impulse Responses

The linear AIR method was found, as shown in Figure 7-22, to have quite poor accuracy for high amplitude oscillatory motions. This suggests that nonlinearity may be a factor. Because of the relaxed wake used, PMARC is weakly nonlinear. At small impulse amplitudes, the nonlinearity does not appear. For the original 1 ft/s plunge rate channel impulse, nonlinearity was negligible. However, by increasing the impulse amplitude to 20 ft/s, slight nonlinearities appeared. For the kernels shown here, a plunge amplitude of 40 ft/s was used. Best results are expected for a plunge amplitude near the excitation amplitude. The excitation amplitude for the study presented in the following section is 1 ft plunge at 10 Hz, or a plunge rate of, maximally, 62.83 ft/s. The first-order kernel is shown in Figure 7-36 compared to the purely linear impulse response. Figure 7-37 shows the first seven components of the second-order kernel—clearly, the degree of nonlinearity is quite small; the first-order kernel deviates only slightly from the linear impulse response, and the magnitude of the $T = 0$ component is an order of magnitude smaller than the first-order kernel. Qualitatively, the second-order kernel shows rapid damping to zero as T increases, as was expected—in fact, Figure 7-36 shows that the third and higher components are nearly negligible compared to the first two. The second-order kernel is predominantly positive as expected; strangely, however, the second component behaves oppositely to the remaining components. This behavior remains unexplained.

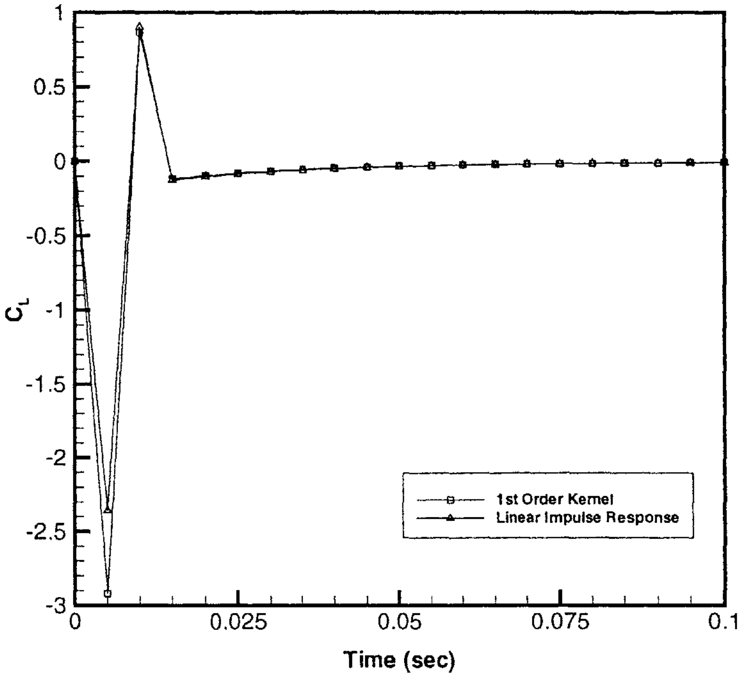


Figure 7-36: First-Order Kernel for High-Amplitude Plunge, $\Delta t_{impulse} = 0.01$ seconds, SHAR2 Wing

Obtaining the responses from PMARC needed to assemble the first- and second-order nonlinear kernels turned out to be significantly more computationally expensive than the linear impulse response identification. Besides the fact that seven responses for just the plunge rate channel were required, the solver converged very slowly for high-amplitude, multiple impulses. While 77 CPU seconds were required for the computation of the linear impulse response, the average time required for the multiple impulse response was 392 CPU seconds.

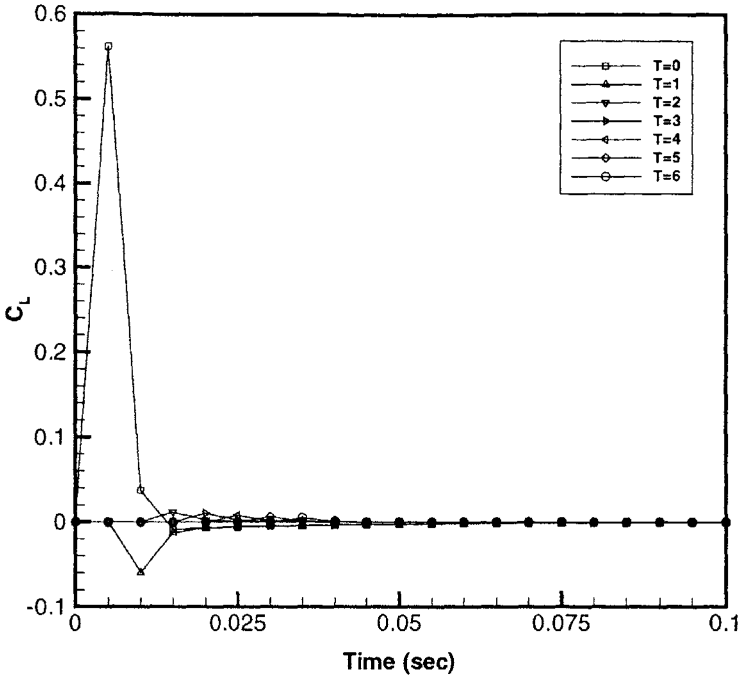


Figure 7-37: Seven Components of the Second-Order Kernel for High-Amplitude Plunge, $\Delta t_{impulse} = 0.01$ seconds, SHAR2 Wing

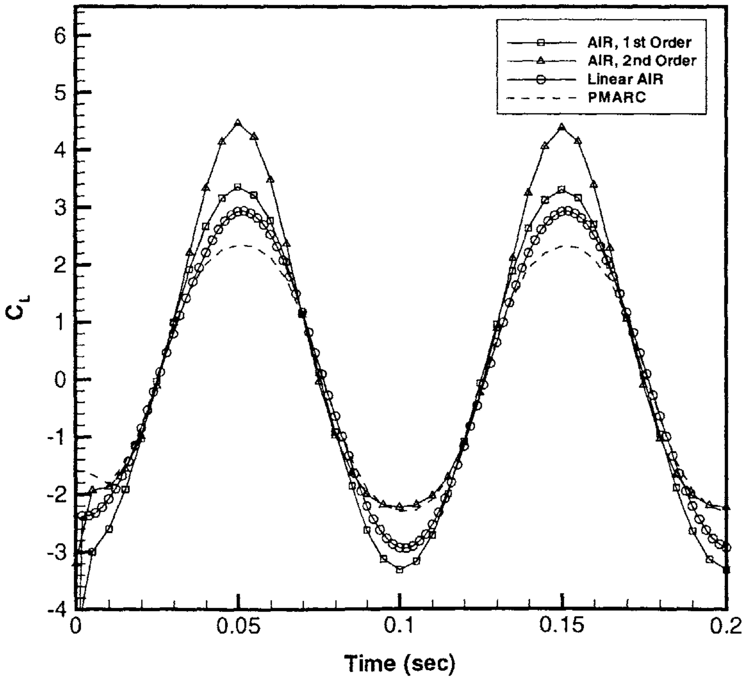


Figure 7-38: Linear and Nonlinear AIR Responses of the SHAR2 Wing to 1.0 ft, 1 Hz Plunging Oscillations Compared to the PMARC Response

7.3.6 Nonlinear Responses to Sinusoidal Motions

The first- and second-order plunge kernels shown above were "convolved," using the Volterra series, with a 10 Hz sinusoidal input function with various amplitudes as was done with the linear response and illustrated in Figures 7-19 through 7-22. The results of this nonlinear analysis for the 1.0 ft plunge amplitude (62.83 ft/s) case are shown in Figure 7-38. The figure compares the linear AIR response, the first-order nonlinear AIR response, and the second-order nonlinear AIR response with the response predicted by PMARC. As might be surmised from Figure 7-36, the linear impulse response gives a better fit to the control data than does the nonlinear first-order kernel. Both significantly overpredict the amplitude of the response. The behavior of the response due to the second-order kernel is peculiar. The lower half of the response matches the control data well; however, the upper half is overpredicted by a greater amount than both the linear and first-order responses. This is the result of the second-order term in the Volterra series being nearly always positive. Further study will be required to investigate this behavior; it remains unclear whether this is the result of a conceptual error or an implementation error.

7.3.7 Summary of Rigid-Body Results

In general, the results obtained for rigid-body motions using the AIR method were very satisfactory. The method works well for all the types of motions and configurations examined and provides a considerable extension to PMARC's unsteady capabilties. The validation tests presented in this section lead to the following conclusions about the method:

- With the appropriate choice of $\Delta t_{impulse}$, the method predicts sinusoidal responses with near perfect accuracy
- Responses to step and arbitrary input functions are predicted with reasonable accuracy, thus extending PMARC's unsteady capabilities
- The Aerodynamic Impulse Response method markedly improves the

computational efficiency of PMARC, especially for long-time inputs

- The $f\Delta t_{impulse}$ frequency-time step factor restricts the method's general applicability
- Input function amplitude limitations with the linear AIR formulation were observed
- For best accuracy, the method is only valid at the freestream conditions at which the impulse response was computed
- The accuracy of the method does not appear to be aspect ratio dependent; this indicates the method should work equally well for arbitrary configurations
- Slight nonlinearity appears only with large impulse amplitudes, not unexpectedly since PMARC is weakly nonlinear
- Inclusion of nonlinear kernels appears to improve the prediction of responses to high-amplitude oscillations, but the method remains unperfected
- However, nonlinear responses may be unwieldy because of the need to generate many impulse responses; in this case, seven per input channel were needed, and the average time required to generate each response was much greater than for linear impulse responses

7.4 Flexible-Body PMARC/AIR

Limited studies using the flexible-body implementation of the AIR method were performed and the results are presented here. All results are for the SHAR2 geometry, and only linear responses were considered. PMARC in its original state is a rigid-body code; because of this, control data was available only for select cases.

7.4.1 Initial Testing

The first step in testing the flexible-body implementation was to ascertain that it reproduces steady and unsteady rigid-body responses. PMARC/AIR for flexible bodies has been designed to accommodate a range of impulse types, controlled by the IMPTYP variable that has been added to the PMARC input file. Setting IMPTYP to 0 specifies steady, rigid-body flow—no impulse. The code then executes similarly to the original code and should exactly duplicate its results. For this test, the SHAR2 Wing with a fixed angle of attack of 5° was used. Both the original PMARC code and the PMARC/AIR code were run using the same data input file—the original PMARC code simply skips the added namelist. As is shown in Figure 7-39, the code modifications are transparent, and, in fact, the execution times are nearly identical, 50.13 CPU seconds for the flexible-body case and 50.06 CPU seconds for the rigid-body case.

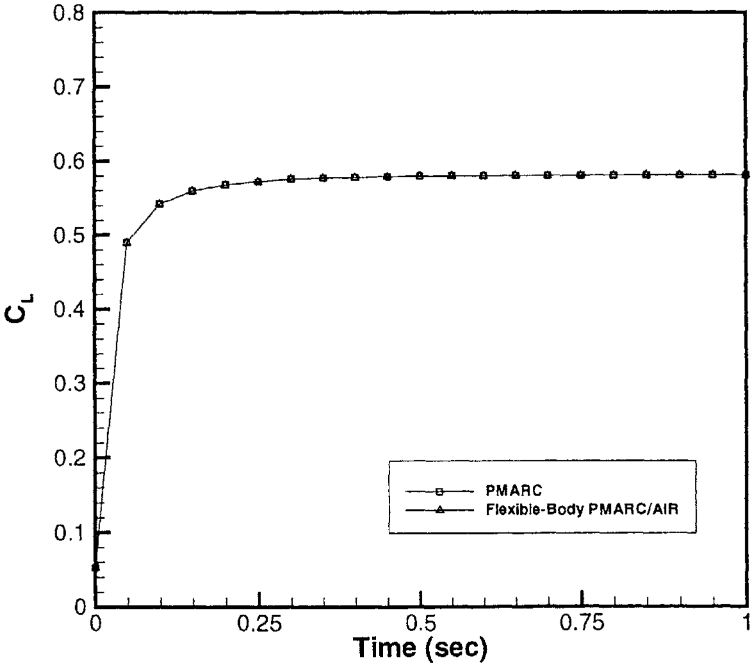


Figure 7-39: Comparison of Flexible-Body PMARC/AIR with PMARC for Rigid-Body, Steady Flow with the SHAR2 Wing

By setting IMPTYP to 1, rigid-body impulse execution is selected. Rigid-body impulses use the same impulse magnitude along the entire span of the wing. The program outputs two impulse response data files, one of the total C_L response and one of the sectional C_L response. For this test, the total C_L response of the pitch channel was compared with that predicted by the rigid-body PMARC/AIR implementation. The flexible-body code reproduces the impulse response from the rigid-body code, and execution times are nearly equal, 65.13 seconds and 64.43 seconds for flexible- and rigid-body implementations, respectively.

7.4.2 Linear Impulse Responses

Impulse responses were obtained for two types of impulse distributions: constant and linear. As shown in the previous section, a constant impulse distribution recovers the rigid-body response. Figure 7-40 is an example of the pitch channel impulse response for a constant impulse distribution. This response curve is simply the three-dimensional representation of multiple sectional impulse responses. The execution time for the flexible-body code for this constant impulse response was nearly equal to its execution time in the previous section for a rigid-body response.

Simple flexible-body motion is modeled with a linear impulse distribution, which approximates a linear deformation, either linear twisting or linear bending. This is the same deformation mode that was studied using the lumped vortex formulation in Section 5.3.3. Though not a physical deformation mode, it is the simplest to study because limited comparisons can be made with results from PMARC. This will be discussed in greater detail in the next section. Figure 7-41 shows the response of the SHAR2 Wing to a linearly distributed pitch impulse for $\Delta t_{impulse} = 0.05$ seconds. Execution time for the linear impulse distribution was slightly longer than the preceeding cases, 68.12 CPU seconds. Pitch rate and plunge rate channel impulse responses are similar.

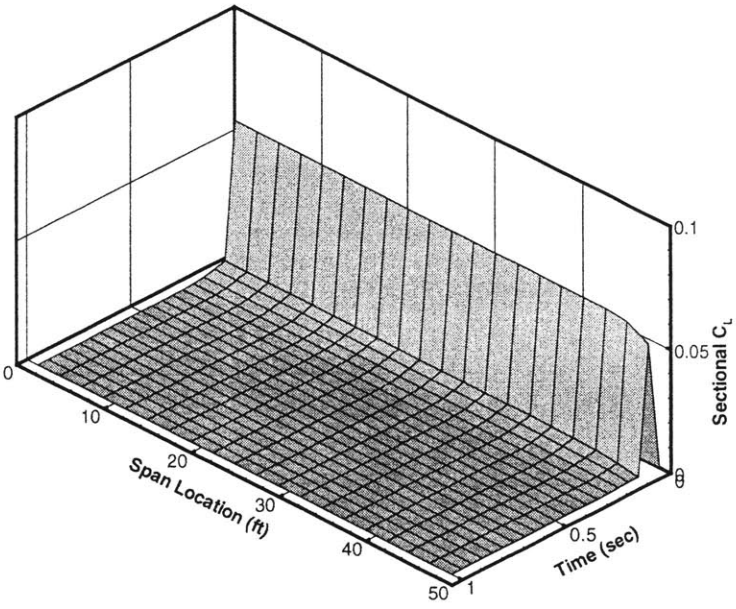


Figure 7-40: Impulse Response of the Pitch Channel to a Constant Impulse Distribution with $\Delta t_{impulse} = 0.05$ seconds, SHAR2 Wing

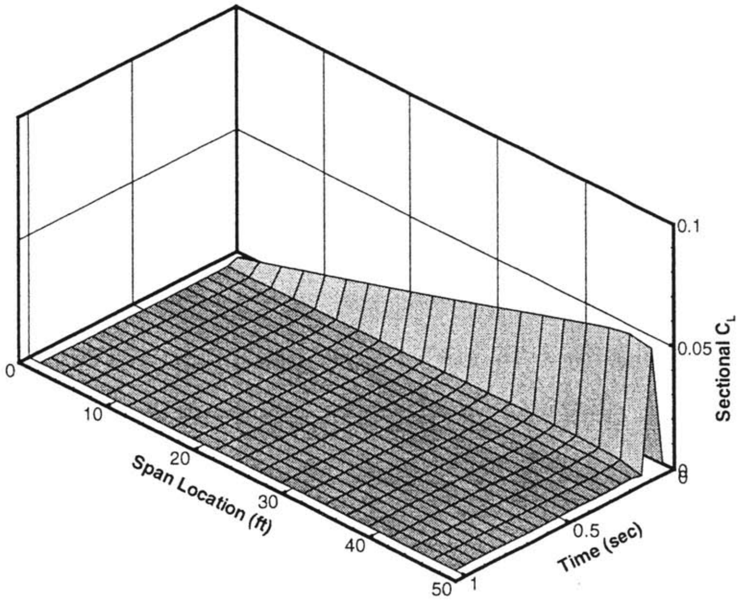


Figure 7-41: Response of the Pitch Channel to a Linear Impulse Distribution with $\Delta t_{impulse} = 0.05$ seconds, SHAR2 Wing

7.4.3 Linear Responses to Various Input Motions

Only for limited flexible-body cases can comparisons with PMARC be made because PMARC is, strictly speaking, a rigid-body code. The case presented here compares the impulsive start of a linearly twisted rigid wing with the the response of a flexible wing to a step input of the pitch channel. The linearly twisted rigid wing can be modeled easily with PMARC by, starting with the SHAR2 Wing input file, specifying a 10° rotation of the outboard wing section. The pitch channel response of this wing to an impulsive start was found using PMARC with no initial wake. For the flexible-body response, the linearly distributed pitch impulse response (Figure 7-41) was convolved with a step function of step amplitude 10°. These two responses are compared in Figures 7-42 through 7-44, where Figures 7-42 and 7-43 show the full three-dimensional responses from two different views and Figure 7-44 shows a two-dimensional slice at the 60% span location. The three-dimensional representations show that there are indeed noticeable differences between the two responses. The magnitude of the flexible-body response is slighly higher (as was also seen in the comparison of the rigid-body PMARC/AIR code with the original PMARC code, Figure 7-33), and at the tip, the flexible body response drops to zero while the rigid-body response drops but remains significant. Scrutiny of the two-dimesional slice in Figure 7-44 shows that, as hypothesized earlier, the convolved response reaches a constant value after the twentieth time step, the point at which the impulse response was truncated. While differences in the responses are present, the AIR method result is reasonably accurate.

The remaining flexible-body response is shown independently as no control data is available. However, based on the comparisons shown already, it is expected that the method does predict the response to arbitrary flexible-body motions with reasonable accuracy. Figure 7-45 shows the response to sinusoidally oscillating linear bending—basically, flapping—with an amplitude of 0.1 ft at the wing tip and a frequency of 1 Hz. Results appear reasonable. The figure shows, as is expected, a linear increase in lift along the span, and at the tip the lift drops to zero.

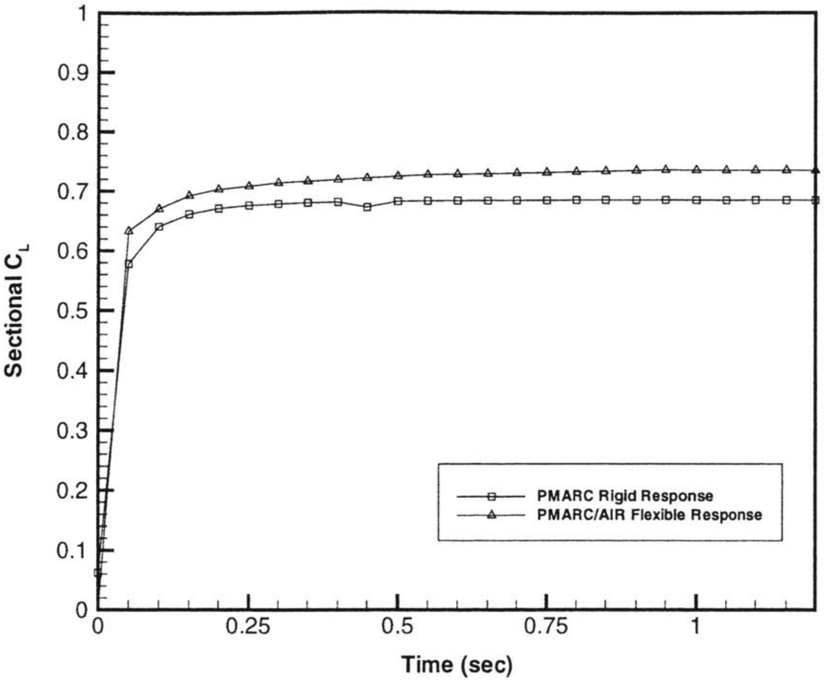


Figure 7-44: Response of the SHAR2 Wing to a Step Linear Twist (10° at Tip), y/b=0.6

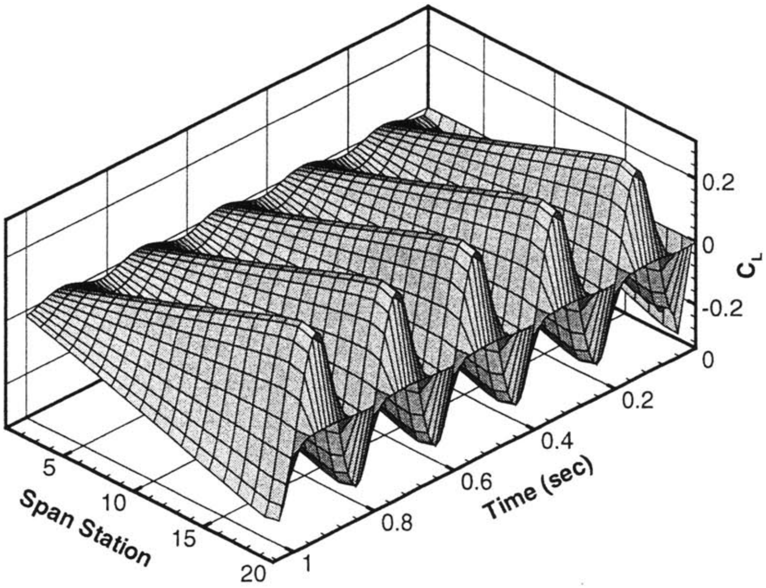


Figure 7-45: Response of the SHAR2 Wing to 1 Hz Sinusoidally Oscillating Linear Bending

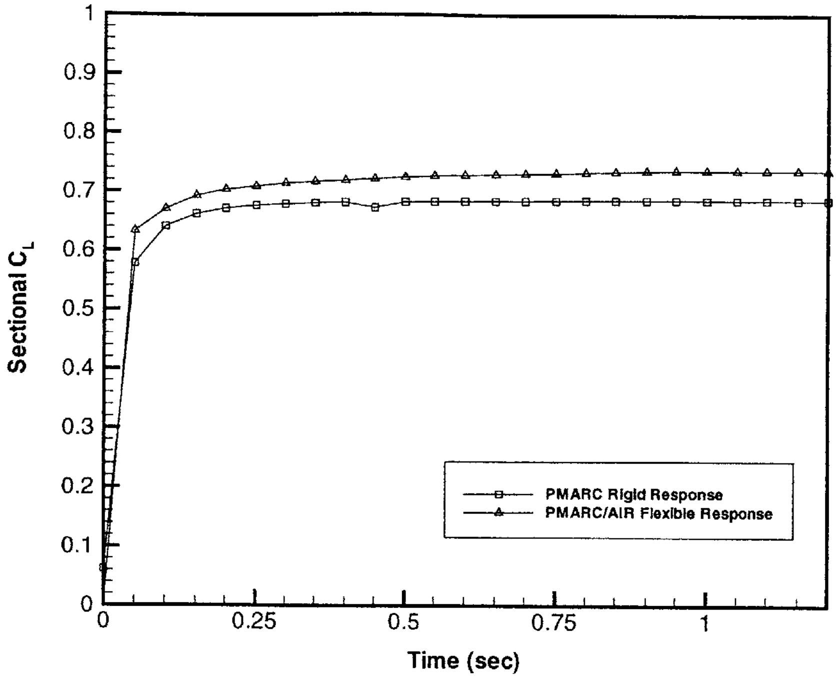


Figure 7-44: Response of the SHAR2 Wing to a Step Linear Twist (10° at Tip), y/b=0.6

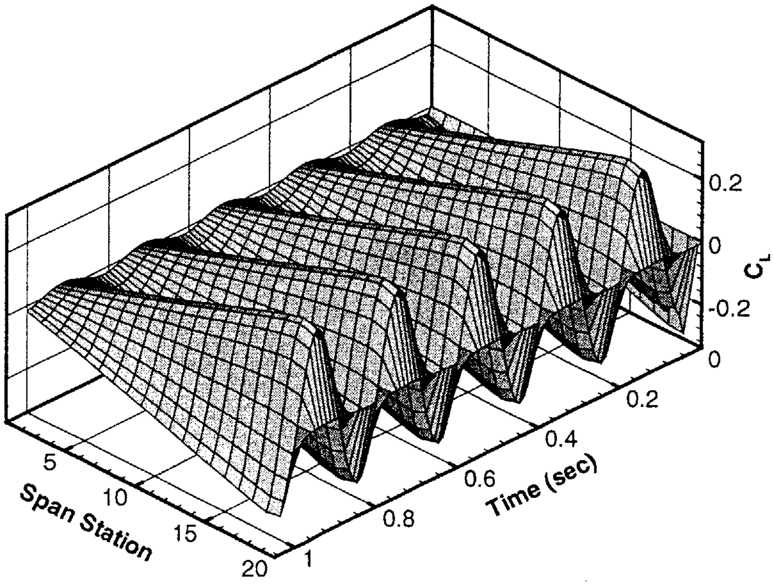


Figure 7-45: Response of the SHAR2 Wing to 1 Hz Sinusoidally Oscillating Linear Bending

7.4.4 Summary of Flexible-Body Results

For the aeroelastic analysis tool that is the ultimate end of this research, the flexible-body responses are those of most interest. The results shown here indicate that the flexible-body formulation of the Aerodynamic Impulse Response method is an effective one for predicting the unsteady aerodynamic forces on a deformable body. The code developed was able to duplicate earlier results for steady and unsteady rigid bodies. The accuracy of the flexible responses studied was difficult to ascertain due to the lack of control data. However, the responses appear reasonable. The frequency-time step factor was not studied for the flexible-body response, though it is assumed to behave similarly to the rigid-body implementation. Further study is required to resolve the issue. Finally, the small deformation assumption remains in this implementation. The code is valid for small deformations about the undeflected shape or about a highly deformed nonlinear shape; the code cannot, strictly speaking, handle large motions of the wing. Dynamic modification of the influence coefficient matrix in PMARC will be required to eliminate this limitation.

Chapter 8

Concluding Remarks

This thesis has presented the development and implementation of the Aerodynamic Impulse Response method with the aerodynamic panel code PMARC for rigid and flexible structures. The AIR method is a general reduced-order method that identifies the impulse response as the fundamental unsteady aerodynamic response, and, given this fundamental response, can determine the response to an arbitrary excitation quickly and efficiently. The method has linear and nonlinear forms employing, respectively, linear convolution and the Volterra series.

Prior work with the AIR method has been done employing CFD methods to obtain impulse responses. For the design-phase analysis, often sufficient accuracy can be obtained with a simplified aerodynamic model, such as a panel code, without the man and machine costs of preparing and running a full CFD model. Such models are much less computationally expensive than CFD methods, but are valid for limited flow regimes and are often limited to steady flow or simple unsteady motions. Coupling a simplified model to the impulse response method enables the prediction of arbitrary unsteady responses with a modest additional time savings. The panel code PMARC was selected as the aerodynamic platform for this research.

The PMARC source code was modified to obtain the impulse response of rigid wings; in conjunction with a simple convolution code, the response of these wings to any unsteady motion can be found. Obtaining the impulse response is the most computationally demanding part of the process, and this needs only to be done once.

Convolution of the impulse response with the arbitrary input takes seconds at most.

For the response of flexible wings, as is required for an aeroelastic analysis, a novel approach was taken. Instead of using a constant impulse amplitude for the entire wing, the impulse was allowed to vary. Using impulse "modes" corresponding to structural mode shapes, the response of a flexible wing can be determined. In essence this is similar to performing a rigid-body impulse response analysis at each span station; however, because the problem is approached globally rather than stripwise, interactions between span stations are accounted for.

8.1 Conclusions

The results of the various formulations of the impulse response method presented in Chapter 7 show clearly that the method is a capable and effective one for predicting arbitrary responses. Using this method, the capabilities of PMARC are extended to include arbitrary unsteady motions for both rigid and flexible structures. Besides extending PMARC's capabilities, because the same impulse response can be used to determine the response of various input motions, a substantial computational savings is realized as well.

Sinusoidal responses with the rigid-body implementation do reveal some limitations which will have to be studied further. Most significantly, for best accuracy the time step size of the impulse response is dictated by the frequency of the input function. Specifically, it was found that best results were obtained for large time steps. A frequency-time step factor, $f\Delta t_{impulse}$, was identified; best accuracy is obtained when this factor nears 0.05. This factor, however, is somewhat ambiguous for non-sinusoidal input functions. Experimentation with a three element sine-series input revealed that even for input with discrete but multiple frequencies, the frequency-time step factor remains ambiguous.

In addition to the frequency-time step dependence, some input amplitude restrictions were identified as well. Good accuracy was obtained for pitch through $\pm 10^\circ$, but a noticeable degradation of the solution was found for plunge amplitudes greater than about half-chord, $0.5c$. The nonlinear form of the impulse response method was employed to attempt to improve the results for high-amplitude inputs. First- and second-order kernels for a plunging wing were obtained and the response to a sinusoidal input was generated using the Volterra series. However, rather than an overall improvement, the response improved in some regions and worsened in others when compared to linear theory. This behavior remains unexplained.

Besides the frequency-time step dependence and some input amplitude limitations, the rigid-body formulation of the AIR method has been found effective for several wing configurations undergoing various motions. With that foundation, a flexible-body implementation was developed. Flexible-body analyses were approached with a modal-type formulation. Preliminary testing with a lumped vortex model in MATLAB indicated that arbitrary wing deformations can be described with the kinematic velocity. An impulse varying across the wing then provides the fundamental response of the wing undergoing deformations of that form. Validation tests showed that this new implementation can duplicate steady and rigid-body analyses. Limited experimentation with the code in flexible-body mode suggests that the formulation is valid and is an effective one for predicting the aerodynamic response of flexible bodies. With this flexible-body code, an aeroelastic analysis tool can be put together by coupling in a structural modal analysis module.

8.2 Recommendations for Future Study

While a great deal has been accomplished towards using the Aerodynamic Impulse Response method with PMARC, the ultimate goal of developing an aeroelastic analysis tool for highly-flexible structures demands considerably more work. The work presented in this thesis is merely the first of several stepping-stones towards that goal. Several refinements to PMARC/AIR remain necessary before the next steps can be taken towards the aeroelastic model. Steps yet to be taken include extension of the linear flexible-body PMARC/AIR implementation to nonlinear aerodynamic systems and coupling PMARC/AIR to a linear and then nonlinear structural analysis.

There remains work to be done with the Aerodynamic Impulse Response method as implemented here, as well. Most significantly, no satisfactory explanation for the loss of accuracy for $f\Delta t_{impulse} < 0.05$ has been found. Because of this time step requirement, the current implementation of the method is limited from highly accurate responses across a broad range of input excitations. Desirably, the time step size should be selected to resolve the *highest* frequency to be studied; currently, the time step needs to be selected for every frequency to obtain best accuracy. Further research will have to be dedicated towards studying the nature of the impulse response from the panel code to get a better understanding of this behavior.

Additional work is also necessary on the nonlinear formulation of the impulse response method. That all the components of the second-order kernel were positive except the second is puzzling and should be looked into futher. The nonlinear response shown in Section 7.3.6 is clear evidence that further work is required. Though overall a poor match to the PMARC control response, the lower half of the response matches, suggesting that further tuning of the technique could produce a better result. It is still not clear whether the mismatch is the result of a conceptual or an implementation error.

Though structurally sound, the code implementations could also benefit from some revisions to improve efficiency and robustness. As noted earlier, considerable savings could be realized by eliminating the spanwise loop in the flexible-body PMARC/AIR code for all time steps but that of the impulse. Though not a particularly expensive operation, the fact remains that for all but one time step the loop is unnecessary. A more efficient algorithm not involving significant code modifications would greatly benefit the final model.

The flexible-body PMARC/AIR unsteady aerodynamics model presented here remains limited in one key aspect that will have to be reconsidered in the final model. Namely, the code is limited to small perturbations. These perturbations may be about a static nonlinear deflected shape, provided that shape is specified in the PMARC input file, but the code, as is, does not handle large dynamic deformations. The reason for this is that large deformations result in nontrivial changes to the influence coefficient matrix. Currently, that matrix is calculated only once at the beginning of the code execution. To account for large deflections, that matrix will need to be recomputed at each time step.

Once these refinement issues have been settled, development of the aeroelastic model may continue. For a first-order aeroelastic model, linear aerodynamics will suffice; the flexible-body PMARC/AIR code need only be coupled with a structural modal analysis program. This will enable the determination of linear aeroelastic responses to arbitrary motions of a vehicle in the time-domain. For greater accuracy for highly flexible wings, the method will then have to be extended to include at least nonlinear structural dynamics. Some aerodynamic nonlinear effects such as dynamic stall should be incorporated, as well, for completeness and robustness of the model.

Bibliography

[1] Unmanned Aerial Vehicles Web Site. uav.wff.nasa.gov.

[2] K. C. Hall. Eigenanalysis of unsteady flows about airfoils, cascades, and wings. *AIAA Journal*, 32(12):2426–2432, December 1994.

[3] K. E. Willcox. *Reduced-Order Aerodynamic Models for Aeroelastic Control of Turbomachines.* PhD Thesis, Massachusetts Institute of Technology, 2000.

[4] D. A. Peters, S. Karunamoorthy, and W. Cao. Finite state induced flow models part I: Two-dimensional thin airfoil. *Journal of Aircraft*, 32(2):313–322, March-April 1995.

[5] M. Drela. Integrated simulation model for preliminary aerodynamic, structural, and control-law design of aircraft. In *Proceedings of the 40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 1999.

[6] M. J. Patil and D. H. Hodges. On the importance of aerodynamic and structural geometrical nonlinearities in aeroelastic behavior of high-aspect-ratio wings. In *Proceedings of the AIAA/ASME/AHS Adaptive Structures Forum*, 2000.

[7] J. Katz and B. Maskew. Unsteady low-speed aerodynamic model for complete aircraft configurations. *Journal of Aircraft*, 25(4):302–310, April 1988.

[8] W. A. Silva. *Discrete-Time Linear and Nonlinear Aerodynamic Impulse Responses for Efficient CFD Analyses.* PhD Thesis, The College of William and Mary, 1997.

[9] W. A. Silva. Reduced order models based on linear and nonlinear aerodynamic impulse responses. In *Proceedings of the 40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 1999.

[10] W. A. Silva. Identification of linear and nonlinear aerodynamic impulse responses using digital filter techniques. NASA TM 112872, 1997.

[11] P. Worch, J. Borky, R. Gabriel, W. Heiser, T. Swalm, and T. Wong. UAV technologies and combat operations, Vol. 1. Technical Report SAB-TR-96-01, USAF Scientific Advisory Board, 1996.

[12] G. H. McCall and J. A. Corder. *New World Vistas: Air and Space Power for the 21st Century.* DIANE Pulishing Co, 1996.

[13] R. L. Bisplinghoff, H. Ashley, and R. L. Halfman. *Aeroelasticity.* Dover, 1996.

[14] P. P. Friedmann. Renaissance of aeroelasticity and its future. *Journal of Aircraft*, 36(1):105–121, January-February 1999.

[15] R. A. Ibrahim and T. D. Woodall. Linear and nonlinear modal analysis of aeroelastic structural systems. *Computers and Structures*, 22(4):699–707, 1986.

[16] S. Setio, H. D. Setio, and L. Jezequel. Modal analysis of nonlinear multi-degree-of-freedom structures. *International Journal of Analytical and Experimental Modal Analysis*, 7(2):75–93, April 1992.

[17] N. Boivin, C. Pierre, and S. W. Shaw. Non-linear modal analysis of the forced response of structural systems. In *Proceedings of the AIAA Dynamics Specialists Conference*, pp. 385–405, 1996.

[18] C. T. Tran and D. Petot. Semi-empirical models for the dynamic stall of airfoils in view of the application to the calculation of responses of a helicopter blade in forward flight. *Vertica*, 5:35–53, May 1981.

[19] O. Bendiksen. A new approach to computational aeroelasticity. In *Proceedings of the 32nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 1991.

[20] G. Guruswamy. ENSAERO:A multi-disciplinary program for fluid/structural interaction studies of aerospace vehicles. *Computing Systems in Engineering*, 1(2–4):237–256, 1990.

[21] G. Guruswamy. User's guide for ENSAERO: A multi-disciplinary program for fluid/structural/control interaction studies of aircraft. NASA TM 108853, 1994.

[22] B. A. Robinson, J. T. Batina, and H. T. Y. Yang. Aeroelastic analysis of wings using the Euler equations with a deforming mesh. NASA TM 102733, 1990.

[23] M. J. Smith, D. M. Schuster, L. Huttsell, and B. Buxton. Development of an Euler/Navier-Stokes aeroelastic method for 3-D vehicles with multiple flexible surfaces. In *Proceedings of the 37th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 1996.

[24] M. J. Smith, C. E. S. Cesnik, and D. H. Hodges. Evaluation of computational algorithms suitable for fluid-structure interactions. *Journal of Aircraft*, 37(2):282–294, March–April 2000.

[25] M. J. Smith, C. E. S. Cesnik, and D. H. Hodges. An evaluation of computational algorithms to interface between CFD and CSD methodologies. U. S. Air Force Research Laboratory, TR-96-3055, November 1995.

[26] E. H. Dowell (Editor), E. F. Crawley, H. C. Curtiss, Jr., D. A. Peters, R. H. Scanlan, and F. Sisto. *A Modern Course in Aeroelasticity*. Kluwer Academic Publishers, 3rd edition, 1995.

[27] B. A. Schimel and W. J. Grantham. *The Engineering Handbook*. CRC Press, 1995.

[28] J. W. Edwards and J. B. Malone. Current status of computational methods for transonic unsteady aerodynamics and aeroelastic applications. *Computing Systems in Engineering*, 3(5):545–569, 1992.

[29] E. H. Dowell. Eigenmode analysis in unsteady aerodynamics: Reduced-order models. *AIAA Journal*, 34(8):1578–1583, August 1996.

[30] J. Heeg and E. H. Dowell. Aerodynamic and aeroelastic insights using eigenanalysis. In *Proceedings of the CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics*, pp. 403–413, 1999.

[31] R. Florea, K. C. Hall, and E. H. Dowell. Eigenmode analysis and reduced order modeling of unsteady transonic full potential flow around isolated airfoils. In *Proceedings of the CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics*, 1999.

[32] J. P. Thomas, K. C. Hall, and E. H. Dowell. Reduced-order aeroelastic modeling using proper-orthogonal decompositions. In *Proceedings of the CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics*, 1999.

[33] J. D. Anderson, Jr. *Fundamentals of Aerodynamics*. McGraw-Hill, 2nd edition, 1991.

[34] D. A. Peters and M. J. Johnson. Finite-state airloads for deformable airfoils on fixed and rotating wings. In *Proceedings of the Symposium on Aeroelasticity and Fluid/Structure Interaction*, 1994.

[35] D. A. Peters. Finite-state airloads modeling with compressibility and unsteady free-stream. In *Proceedings of the Sixth International Workshop on Dynamics and Aeroelastic Stabilty Modeling of Rotorcraft Systems*, 1995.

[36] M. J. Patil, D. H. Hodges, and C. E. S. Cesnik. Characterizing the effects of geometrical nonlinearities on aeroelastic behavior of high-aspect-ratio wings. In

Proceedings of the CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics, pp. 501–510, 1999.

[37] C. P. Mracek and D. T. Mook. Aerodynamic potential flow panel method coupled with dynamics and controls. AIAA Paper 91-2846, 1991.

[38] B. D. Hall, S. Preidikman, D. T. Mook, and A. H. Nayfey. A time-domain simulation for evaluating smart wing concepts for reducing gust loads. In *Proceedings of the SPIE Conference on Smart Structures and Integrated Systems*, pp. 105–116, 1999.

[39] B. Maskew. Program VSAERO, a computer program for calculating the nonlinear aerodynamic characteristics of arbitrary configurations. NASA CR-166476, November 1982.

[40] D. L. Ashby, M. R. Dudley, S. K. Iguchi, Lindsey Browne, and Joseph Katz. Potential flow theory and operation guide for the panel code PMARC. NASA TM 102851, 1991.

[41] M. van Schoor and A. H. von Flotow. Aeroelastic characteristics of a highly flexible aircraft. *Journal of Aircraft*, 27(10):901–908, October 1990.

[42] J. W. Edwards, H. Ashley, and J. V. Breakwell. Unsteady aerodynamic modeling for arbitrary motions. *AIAA Journal*, 17(4):365–374, April 1979.

[43] R. A. Ibrahim. Response analysis of nonlinear systems using functional-perturbational type approach. In *Proceedings of the 6th International Modal Analysis Conference*, pp. 1480–1488, 1988.

[44] F. Udwadia. The identification of building structural systems. In *Proceddings of the First Symposium on Testing and Identification of Nonlinear Systems*, pp. 38–55, 1975.

[45] W. Krabs. *Mathematic Foundations of Signal Theory.* Helderman Verlag, 1995.

[46] L. C. Ludeman. *Fundamentals of Digital Signal Processing.* Harper and Row, 1986.

[47] C. D. McGillem and G. R. Cooper. *Continuous and Discrete Signal and System Analysis.* Holt, Rinehart, and Winston, 1974.

[48] J. Katz and L. Plotkin. *Low-Speed Aerodynamics: From Wing Theory to Panel Methods.* McGraw-Hill, 1992.

[49] H. W. M. Hoeijmakers. Panel methods for aerodynamic analysis and design. In *Special Course on Engineering Methods in Aerodynamic Analysis and Design of Aircraft*, January 1992. AGARD Report 783.

[50] J. L. Hess. Calculation of potential flow about arbitrary three-dimensional lifting bodies. Technical Report MDC J5679-01, McDonnell Douglas, Long Beach, California, October 1972.

[51] F. A. Woodward. An improved method for the aerodynamic analysis of wing-body tail configurations in subsonic and supersonic flow. NASA CR-2228, May 1973.

[52] D. R. Bristow. Development of panel methods for subsonic analysis and design. NASA CR-3234, 1980.

[53] L. Morino. Steady, oscillatory, and unsteady subsonic and supersonic aerodynamics—Production Version (SOUSSA-P1,1) Vol. 1, Theoretical Manual. NASA CR-159130, 1980.

[54] R. L. Carmichael and L. I. Ericson. PAN-AIR—A higher order panel method for predicting subsonic or supersonic linear potential flows. AIAA Paper 81-1255, June 1981.

[55] A. E. Magnus and M. A. Epton. PAN-AIR—A computer program for predicting subsonic or supersonic linear potential flows about arbitrary configurations using

a higher order panel method, Vol. 1—Theory Document. NASA CR-3251, November 1983.

[56] J. L. Hess and D. M. Friedman. An improved high-order panel method for three-dimensional lifting flow. Technical Report NADC-79277-60, Douglas Aircraft Co., 1981.

[57] H. H. Youngren, E. E. Bouchard, R. M. Coppersmith, and L. R. Miranda. Comparison of panel method formulations and its influence on the development of QUADPAN, an advanced low-order method. AIAA Paper 83-1827, 1983.

[58] H. Lamb. *Hydrodynamics.* Cambridge University Press, 1993.

[59] D. M. Martin and I. Kroo. Comparison of PMARC and analytic results for two-dimensional unsteady airfoils. In *Proceedings of the 31st Aerospace Sciences Meeting and Exhibit*, 1993.

[60] M. Ortega-Morales and C. E. S. Cesnik. Modeling and control of the aeroelastic response of highly flexible active wings. AMSL Report #00-2, Active Materials and Structures Labortatory, Massachusetts Institute of Technology, February 2000.

[61] R. C. Scott, S. T. Hoadley, C. D. Wieseman, and M. H. Durham. The benchmark active controls technology model aerodynamic data. In *Proceedings of the 35th Aerospace Sciences Meeting & Exhibit*, 1997.

Appendix A

PMARC/AIR Code Listings

A.1 Linear Rigid-Body PMARC/AIR

The rigid-body code required modifications to only the PATH and WAKSTEP subroutines. The PATH subroutine is exerpted here; modifications to the WAKSTEP subroutine are similar.

```
      SUBROUTINE PATH(T)
      .
      .
      .
C
C  ALPHA AND YAW, AS INPUT, CONVERTED TO RADIANS
C
      ALPHA = (ALDEG) * PI/180.
      YAW = (YAWDEG) * PI/180.
C
C  SET UP INITIAL FREE-STREAM VELOCITY VECTOR
C
      SB = SIN(YAW)
      CB = COS(YAW)
      CA = CB * COS(ALPHA)
      SA = CB * SIN(ALPHA)
      VX1 = -CA * VINF
      VY1 = SB * VINF
      VZ1 = -SA * VINF
C
```

```
C  --ADDED BY REG 05 AUG 99
C  PLUNGE RATE CHANNEL; CHANGE VZ TO 1.0, SET Q AND THETA TO 0.0
C  FOR IMPULSE RESPONSE OF THIS CHANNEL
C
      IF (WTZ.LT.0.0) THEN
         IF (ITSTEP.GE.IABS(INT(WTZ/(2*PI)))) THEN
            VX=VX1
            VY=VY1
            VZ=VZ1
         ELSE
            VX=VX1
            VY=VY1
            VZ=VZ1
         ENDIF
C
C  DETERMINE THE CONTRIBUTION TO TRANSLATION VELOCITY
C  COMPONENTS FROM HEAVING OSCILLATIONS
C
      ELSE
 VX = VX1 + DXMAX * WTX * COS(WTX*T)
         VY = VY1 + DYMAX * WTY * COS(WTY*T)
         VZ = VZ1 + DZMAX * WTZ * COS(WTZ*T)
      ENDIF
C
C  DETERMINE CURRENT ANGULAR ROTATION RATES IN RADIANS (INCLUDING
C  OSCILLATORY COMPONENT)
C
      P = PHIDOT * PI/180. + PHIMAX * WRX * COS(WRX*T) * PI/180.
      Q = THEDOT * PI/180. + THEMAX * WRY * COS(WRY*T) * PI/180.
      R = PSIDOT * PI/180. + PSIMAX * WRZ * COS(WRZ*T) * PI/180.
C
C  ADDED BY REG JAN 00
C  PITCH RATE CHANNEL UNIT PULSE INPUT
C
      IF(WTZ.LT.0.0)THEN
 IF (ITSTEP.EQ.IABS(INT(WTZ/(2*PI)))) THEN
            Q=1.0
        ELSE
            Q=0.0
         END IF
      END IF
C
C  SET TO ZERO FOR PITCH IMPULSE RESPONSE
C
      Q=0.0
```

```
C
C  DETERMINE CURRENT ANGULAR POSITION IN RADIANS (INCLUDING
C  OSCILLATORY CONTRIBUTION)
C

      PHI = PHIDOT * PI/180. * T + PHIMAX * SIN(WRX*T) * PI/180.
      THETA = THEDOT * PI/180. * T + THEMAX * SIN(WRY*T) * PI/180.
      PSI = PSIDOT * PI/180. * T + PSIMAX * SIN(WRZ*T) * PI/180.
C
C  ADDED BY REG JAN 00
C  PITCH CHANNEL UNIT PULSE INPUT
C
      IF(WTZ.LT.0.0)THEN
 IF (ITSTEP.EQ.IABS(INT(WTZ/(2*PI)))) THEN
            THETA=0.01
         ELSE
            THETA=0.0
         END IF
      END IF
C
C  SET TO ZERO FOR PITCH RATE IMPULSE RESPONSE
C
C      THETA=0.0
C
C  WRITE INPUT TO FILE
C
      WRITE(14,602)ITSTEP,THETA
C
C  STORE VELOCITY VECTOR AND ROTATION RATES IN ARRAYS.
C
      SPSI = SIN(PSI)
      CPSI = COS(PSI)
      STHETA = SIN(THETA)
      CTHETA = COS(THETA)
      SPHI = SIN(PHI)
      CPHI = COS(PHI)
      DC(1,1) = CTHETA * CPSI
      DC(1,2) = CTHETA * SPSI
      DC(1,3) = -STHETA
      DC(2,1) = -CPHI*SPSI + SPHI*STHETA*CPSI
      DC(2,2) = CPHI*CPSI + SPHI*STHETA*SPSI
      DC(2,3) = SPHI*CTHETA
      DC(3,1) = SPHI*SPSI + CPHI*STHETA*CPSI
      DC(3,2) = -SPHI*CPSI + CPHI*STHETA*SPSI
      DC(3,3) = CPHI*CTHETA
```

```
IFLAG = 1
CALL TRANFRM(IFLAG,DC,VX,VY,VZ)
CALL TRANFRM(IFLAG,DC,VX1,VY1,VZ1)
VFR(1) = VX
VFR(2) = VY
VFR(3) = VZ
OMEGA(1) = P
OMEGA(2) = Q
OMEGA(3) = R
.
.
.
```

A.2 Linear Flexible-Body PMARC/AIR

The flexible-body code required more extensive modifications. The PATH subroutine is shown here. In addition, throughout the code, VFR and OMEGA were redefined as two-dimensional arrays. The VFR(0,x) element is used repeatedly through the code for the calculation of velocity potential off the wing surface.

```
      SUBROUTINE PATH(T)
      .
      .
      .
C
C  ALPHA AND YAW, AS INPUT, CONVERTED TO RADIANS
C
      ALPHA = (ALDEG) * PI/180.
      YAW = (YAWDEG) * PI/180.
C
C  SET UP INITIAL FREE-STREAM VELOCITY VECTOR
C
      SB = SIN(YAW)
      CB = COS(YAW)
      CA = CB * COS(ALPHA)
      SA = CB * SIN(ALPHA)
      VX1 = -CA * VINF
      VY1 = SB * VINF
      VZ1 = -SA * VINF
C
C  Added REG 21 Feb 00
C  Set up impulse vector
C
      DO I=1,NCOL(1)
C  No impulse
         IF(IMPTYP.EQ.0)THEN
            IMP_VZ(I)=0.0
            IMP_TH(I)=0.0
            IMP_Q(I)=0.0
C  Rigid-body impulse
         ELSEIF(IMPTYP.EQ.1)THEN
            IF(CHAN.EQ.1)THEN
               IMP_VZ(I)=1.0
               IMP_TH(I)=0.0
               IMP_Q(I)=0.0
            ELSEIF(CHAN.EQ.2)THEN
               IMP_VZ(I)=0.0
               IMP_TH(I)=0.01
               IMP_Q(I)=0.0
            ELSEIF(CHAN.EQ.3)THEN
               IMP_VZ(I)=0.0
               IMP_TH(I)=0.0
               IMP_Q(I)=1.0
            ENDIF
C  Flexible-body impulse
         ELSEIF(IMPTYP.EQ.2)THEN
            IF(ITSTEP.EQ.STEP)THEN
               READ(10,*)DUM,IMP_VZ(I),IMP_TH(I),IMP_Q(I)
            ELSE
               IMP_VZ(I)=0.0
               IMP_TH(I)=0.0
               IMP_Q(I)=0.0
            END IF
         END IF
      END DO
C
C  Added REG 21 Feb 00
C  Loop over columns to determine stripwise kinematic velocity
C  components.  Column zero is the rigid-body mode
C
      DO I=0,NCOL(1)
C
C  DETERMINE THE CONTRIBUTION TO TRANSLATION VELOCITY
C  COMPONENTS FROM HEAVING OSCILLATIONS
C
      VX = VX1 + DXMAX * WTX * COS(WTX*T)
      VY = VY1 + DYMAX * WTY * COS(WTY*T)
      VZ = VZ1 + DZMAX * WTZ * COS(WTZ*T)
C
C  DETERMINE CURRENT ANGULAR ROTATION RATES IN RADIANS
C  (INCLUDING OSCILLATORY COMPONENT)
C
      P = PHIDOT * PI/180. + PHIMAX * WRX * COS(WRX*T) * PI/180.
      Q = THEDOT * PI/180. + THEMAX * WRY * COS(WRY*T) * PI/180.
      R = PSIDOT * PI/180. + PSIMAX * WRZ * COS(WRZ*T) * PI/180.
C
C  DETERMINE CURRENT ANGULAR POSITION IN RADIANS (INCLUDING
C  OSCILLATORY CONTRIBUTION)
C
      PHI = PHIDOT * PI/180. * T + PHIMAX * SIN(WRX*T) * PI/180.
      THETA = THEDOT * PI/180. * T + THEMAX * SIN(WRY*T) * PI/180.
      PSI = PSIDOT * PI/180. * T + PSIMAX * SIN(WRZ*T) * PI/180.
C
C  ADDED BY REG JAN 00/ Modified 21 Feb 00
C  PLUNGE/PITCH/PITCH RATE CHANNEL UNIT PULSE INPUT
C
      IF((IMPTYP.NE.0).AND.(I.NE.0))THEN
 IF (ITSTEP.EQ.STEP) THEN
            VZ=IMP_VZ(I)
            Q=IMP_Q(I)
            THETA=IMP_TH(I)
         ELSE
            VZ=0.0
            Q=0.0
            THETA=0.0
         END IF
      END IF
C
C  STORE VELOCITY VECTOR AND ROTATION RATES IN ARRAYS.
C
      SPSI = SIN(PSI)
      CPSI = COS(PSI)
      STHETA = SIN(THETA)
      CTHETA = COS(THETA)
      SPHI = SIN(PHI)
      CPHI = COS(PHI)
      DC(1,1) = CTHETA * CPSI
      DC(1,2) = CTHETA * SPSI
      DC(1,3) = -STHETA
      DC(2,1) = -CPHI*SPSI + SPHI*STHETA*CPSI
      DC(2,2) = CPHI*CPSI + SPHI*STHETA*SPSI
      DC(2,3) = SPHI*CTHETA
      DC(3,1) = SPHI*SPSI + CPHI*STHETA*CPSI
      DC(3,2) = -SPHI*CPSI + CPHI*STHETA*SPSI
      DC(3,3) = CPHI*CTHETA
      IFLAG = 1
      CALL TRANFRM(IFLAG,DC,VX,VY,VZ)
C
C  Modified REG 22 Feb 00
C  Transform VX1,VY1, and VZ1 only for the rigid body mode
C
      IF(I.EQ.0)THEN
         VX1_RIG=VX1
         VY1_RIG=VY1
         VZ1_RIG=VZ1
         CALL TRANFRM(IFLAG,DC,VX1_RIG,VY1_RIG,VZ1_RIG)
      END IF
C
      VFR(I,1) = VX
      VFR(I,2) = VY
      VFR(I,3) = VZ
      OMEGA(I,1) = P
      OMEGA(I,2) = Q
      OMEGA(I,3) = R
C  End column loop
      END DO
C
C  SET THE CURRENT ALPHA AND YAW ANGLES
C
      IF(ABS(VX1_RIG).GT.EPS)THEN
        ALPHA = ATAN(VZ1_RIG/VX1_RIG)
        YAW = -ATAN(VY1_RIG/VX1_RIG)
      ELSE
        IF(VZ1_RIG.GT.0)THEN
          ALPHA = PI/2.0
        ELSE
          ALPHA = -PI/2.0
        ENDIF
        IF(VY1_RIG.GT.0)THEN
          YAW = -PI/2.0
        ELSE
          YAW = PI/2.0
        ENDIF
      ENDIF
      .
      .
      .
```

A.3 Convolution

```
C     convolution.f
C
C     Convolutes the "impulse response" in file IMPULSE with
C     an arbitary input in file INPUT to yield the response
C     to the arbitrary input.  Output is written to the file
C     RESPONSE.  Filenames are entered at run-time.
C
C     By Randal E. Guendel
C     24 November 1999
C

      PROGRAM CONVOLVE

      PARAMETER (NMAX=10000)

      REAL Y(0:NMAX),H(0:NMAX),X(0:NMAX)
      CHARACTER*25 INPUT,IMPULSE,RESPONSE
      REAL FACT

      WRITE(*,*)'Enter the name of the impulse response file:'
      READ(*,*)IMPULSE
      WRITE(*,*)'Enter a scaling factor for the impulse:'
      READ(*,*)FACT
      WRITE(*,*)'Enter the name of the input file:'
      READ(*,*)INPUT
      WRITE(*,*)'Enter the name of the output file:'
      READ(*,*)RESPONSE

      OPEN(UNIT=10,FILE=IMPULSE,FORM='FORMATTED',STATUS='OLD')
      OPEN(UNIT=11,FILE=INPUT,FORM='FORMATTED',STATUS='OLD')
      OPEN(UNIT=12,FILE=RESPONSE,FORM='FORMATTED',STATUS='NEW')
C---- Initialize arrays to zero
C
      DO I=0,NMAX
         Y(I)=0.0
         H(I)=0.0
         X(I)=0.0
      END DO
C---- Read impulse response from the file
C
      READ(10,101)TSTEP_P
      READ(10,102)NTSTEP_P
 101  FORMAT(F8.5)
 102  FORMAT(I5)
C---- Skip column heading lines and zeroth iteration
C     (because pulse is applied at iteration 1)
C
      READ(10,*)
      READ(10,*)
      DO I=0,NTSTEP_P-1
         READ(10,*)DUM,H(I)
      END DO
C---- Read arbitrary input data from file
C
      READ(11,101)TSTEP_I
      READ(11,102)NTSTEP_I
      IF(TSTEP_I.NE.TSTEP_P)THEN
         WRITE(*,*)'Timestep of impulse resp. and input must match!'
         STOP
      END IF
C---- Skip column heading line
C
      READ(11,*)
      DO I=0,NTSTEP_I
         READ(11,*)DUM,X(I)
      END DO
C---- Perform convolution
C
      DO N=0,NTSTEP_I
         DO K=0,NTSTEP_I
            IF(((N-K).GE.0).AND.((N-K).LT.NTSTEP_P))THEN
               Y(N)=Y(N)+FACT*H(N-K)*X(K)
            END IF
         END DO
      END DO

      WRITE(12,103)
 103  FORMAT('RESPONSE TO ARBITRARY INPUT GENERATED BY CONVOLUTION')
      DO I=0,NTSTEP_I
         WRITE(12,110)I*TSTEP_P,Y(I)
      END DO

 110  FORMAT(F8.5,6X,E12.4)

      STOP
      END
```

A.4 Volterra Series

```
C     volterra.f
C
C     Convolutes the "impulse response" in file IMPULSE with
C     an arbitary input in file INPUT to yield the response
C     to the arbitrary input.  Output is written to the file
C     RESPONSE.  Filenames are entered at run-time.
C
C     By Randal E. Guendel
C     12 April 2000
C

      PROGRAM VOLTERRA

      PARAMETER (NMAX=10000)
      PARAMETER (KMAX=50)

      REAL Y(0:NMAX),Y2(0:NMAX),H1(0:KMAX),H2(0:KMAX,0:KMAX)
      REAL X(0:NMAX)
      CHARACTER*25 INPUT,KERNEL1,KERNEL2,RESPONSE
      REAL FACT
      INTEGER K1,K2

      WRITE(*,*)'Enter the name of the 1st order kernel file:'
      READ(*,*)KERNEL1
      WRITE(*,*)'Enter the name of the 2nd order kernel file:'
      READ(*,*)KERNEL2
      WRITE(*,*)'Enter a scaling factor for the impulse:'
      READ(*,*)FACT
      WRITE(*,*)'Enter the name of the input file:'
      READ(*,*)INPUT
      WRITE(*,*)'Enter the name of the output file:'
      READ(*,*)RESPONSE

      OPEN(UNIT=9,FILE=KERNEL2,FORM='FORMATTED',STATUS='OLD')
      OPEN(UNIT=10,FILE=KERNEL1,FORM='FORMATTED',STATUS='OLD')
      OPEN(UNIT=11,FILE=INPUT,FORM='FORMATTED',STATUS='OLD')
      OPEN(UNIT=12,FILE=RESPONSE,FORM='FORMATTED',STATUS='NEW')
C---- Initialize arrays to zero
C
      DO I=0,NMAX
         Y(I)=0.0
         Y2(I)=0.0
         X(I)=0.0
      END DO

      DO I=0,KMAX
         H1(I)=0.0
         DO J=0,KMAX
            H2(I,J)=0.0
         END DO
      END DO
C---- Read impulse response from the file
C
      READ(10,101)TSTEP_P
      READ(10,102)NTSTEP_P
 101  FORMAT(F8.5)
 102  FORMAT(I5)
C---- Skip column heading lines and zeroth iteration
C     (because pulse is applied at iteration 1)
C
      READ(10,*)
      READ(10,*)
      DO I=0,NTSTEP_P-1
         READ(10,*)DUM,H1(I)
         H2(I,I)=H1(I)
      END DO
C---- Read 2nd order kernel; fill in matrix
C
      READ(9,101)TSTEP_K2
      READ(9,102)NTSTEP_K2
      READ(9,102)NCOMP
      IF(TSTEP_K2.NE.TSTEP_P)THEN
         WRITE(*,*)'Time step of kernels must match!'
         STOP
      END IF
      READ(9,*)
      DO I=1,NCOMP
         DO J=0,NTSTEP_K2
           READ(9,*)DUM,TEMP
           IF(J.GE.I)THEN
              H2(J-I,J-1)=TEMP
              H2(J-1,J-I)=TEMP
           END IF
        END DO
      END DO
C---- Read arbitrary input data from file
```

```
C
      READ(11,101)TSTEP_I
      READ(11,102)NTSTEP_I
      IF(TSTEP_I.NE.TSTEP_P)THEN
         WRITE(*,*)'Timestep of impulse resp. and input must match!'
         STOP
      END IF
C---- Skip column heading line
C
      READ(11,*)
      DO I=0,NTSTEP_I
         READ(11,*)DUM,X(I)
      END DO
C---- Perform convolution
C
      DO N=0,NTSTEP_I

         DO K=0,NTSTEP_I
            IF(((N-K).GE.0).AND.((N-K).LT.NTSTEP_P))THEN
               Y(N)=Y(N)+FACT*H1(N-K)*X(K)
            END IF
         END DO

         DO K1=0,NTSTEP_I
            DO K2=0,NTSTEP_I
               IF(((N-K1).GE.0).AND.((N-K2).GE.0))THEN
               IF(((N-K1).LT.NTSTEP_P).AND.((N-K2).LT.NTSTEP_P))THEN
                  Y2(N)=Y2(N)+FACT*H2(N-K1,N-K2)*X(K1)*X(K2)
               END IF
               END IF
            END DO
         END DO

         Y(N)=Y(N)+Y2(N)

      END DO

      WRITE(12,103)
 103  FORMAT('RESPONSE TO ARBITRARY INPUT GENERATED BY CONVOLUTION')
      DO I=0,NTSTEP_I
         WRITE(12,110)I*TSTEP_P,Y(I)
      END DO

 110  FORMAT(F8.5,6X,E12.4)
```

```
STOP
END
```

A.5 Input Generation Program

```
C     Generates an input file for convolution with an impulse
C     response file.  User specifies the frequency, amplitude,
C     and duration of the input, and can select a pitch, pitch
C     rate, or plunge type function.
C
C     by Randal E. Guendel
C     14 Feb 00

      PROGRAM GENINPUT

      INTEGER INPTYP

      PI=3.14159

      WRITE(*,*)'Select:'
      WRITE(*,*)'  1. Pitch Input (sine)'
      WRITE(*,*)'  2. Pitch Rate Input d/dt(sine)'
      WRITE(*,*)'  3. Plunge Input (cosine)'
      READ(*,*)INPTYP

      WRITE(*,*)'Enter frequency (in Hz):'
      READ(*,*)FREQ

      WRITE(*,*)'Enter time step size:'
      READ(*,*)DT

      WRITE(*,*)'Enter the duration of the oscillation (in sec):'
      READ(*,*)TIME

      WRITE(*,*)'Enter the amplitude (in deg/length units):'
      READ(*,*)AMP

      OPEN(UNIT=10,FILE='input.txt',FORM='FORMATTED',STATUS='NEW')

C     Convert to radians if either pitch; convert to radians
C     and multiply by omega=2*pi*f if pitch rate
      IF(INPTYP.EQ.1)THEN
         AMP=AMP*PI/180
      ELSEIF(INPTYP.EQ.2)THEN
         AMP=AMP*PI/180*FREQ*2*PI
      ELSEIF(INPTYP.EQ.3)THEN
```

```
   AMP=AMP*2*FREQ*PI
ENDIF

NTSTEPS=NINT(TIME/DT)

WRITE(10,*)DT
WRITE(10,*)NTSTEPS
WRITE(10,*)'STEP   AMPLITUDE'

DO I=0,NTSTEPS
   T=I*DT
   IF(INPTYP.EQ.1)THEN
      WRITE(10,*)I,AMP*SIN(2*PI*FREQ*T)
   ELSE
      WRITE(10,*)I,AMP*COS(2*PI*FREQ*T)
   ENDIF
END DO

WRITE(*,*)'Input data written to file input.txt'

STOP
END
```

Appendix B

Other Code Listings

B.1 MATLAB Lumped Vortex Model

```
% LumpVortexWing.m
%
% by Randal E. Guendel
%
% 18 November 1999
%
% Uses a lumped vortex formulation and strip theory to calculate
% the circulation and lift of a rectangular wing with span b and
% chord c for a unit perturbation of the pitch, pitch rate, and
% plunge rate.
%
% Then computes the circulation response for a sinusoidal input by
% convolution with the impulse responses.  This is compared to the
% sinusoidal responses as computed by direct time integration.

clear all;

% Compute impulse responses

T=10.0;
dt=0.01;

N=10;
M=T/dt;
Uinf=1.0;
```

```
beta=2/pi;
b=10;
c=1;
eps=1e-5;
rho=1.0;

time(1)=0;
for i=1:M
   time(i+1)=i*dt;
end;

time0=clock;

a=zeros(M+1,N);
adot=zeros(M,N);
hdot=zeros(M,N);
dy=b/(N*2);
for j=1:N
   a(1,j)=2*dy*j/b;
   adot(1,j)=2*dy*j/b;
   hdot(1,j)=2*dy*j/b;
   y(j)=j*dy;
end;

% Circ1 -> response to the alpha (pitch) channel
% Circ2 -> response to the alpha_dot (pitch rate) channel
% Circ3 -> response to the h_dot (plunge rate) channel
for j=1:N
   Circ1(1,j)=0.0;
   Circ2(1,j)=0.0;
   Circ3(1,j)=0.0;
   for i=2:M+1
      Circ1(i,j)=Circ1(i-1,j)+(Uinf*dt/beta)*(Uinf*a(i-1,j)-
                 (1/(pi*c))*Circ1(i-1,j));
      Circ2(i,j)=Circ2(i-1,j)+(Uinf*dt/beta)*((c/4)*adot(i-1,j)-
                 (1/(pi*c))*Circ2(i-1,j));
      Circ3(i,j)=Circ3(i-1,j)+(Uinf*dt/beta)*(-hdot(i-1,j)-
                 (1/(pi*c))*Circ3(i-1,j));
   end;
end;

etime(clock,time0)
time0=clock;

figure(1);
```

```
subplot(3,1,1), plot(time,Circ1(:,N))
title('Response at tip of alpha input channel
       (linear distribution)');
subplot(3,1,2), plot(time,Circ2(:,N))
title('Response at tip of d(alpha)/dt input channel
       (linear distribution)');
ylabel('circulation');
subplot(3,1,3), plot(time,Circ3(:,N))
title('Response at tip of d(h)/dt input channel
       (linear distribution)');
xlabel('time (sec)');

T1=10;
A=1;
M2=T1/dt;
for i=1:M2+5
      Inp_a(i)=sin(4*pi*dt*i/T1);
      Inp_a_dot(i)=(4*pi/T1)*cos(4*pi*dt*i/T1);
      Inp_h_dot(i)=0;
end;

time2(1)=0;
for i=1:M2-1
   time2(i+1)=i*dt;
end;

% Convolution of the impulse responses with the sinusoidal input

Resp_a=zeros(M2,N);
Resp_a_dot=zeros(M2,N);
Resp_h_dot=zeros(M2,N);
for j=1:N
   Temp1=conv(Circ1(:,j),Inp_a);
   Temp2=conv(Circ2(:,j),Inp_a_dot);
   Temp3=conv(Circ3(:,j),Inp_h_dot);

   for k=1:length(Circ1)-1
      Resp_a(k,j)=Temp1(k);
      Resp_a_dot(k,j)=Temp2(k);
      Resp_h_dot(k,j)=Temp3(k);
   end;
end;

etime(clock,time0)
time0=clock;
```

```
Resp=Resp_a+Resp_a_dot+Resp_h_dot;

% Time-marching, fully discretized lumped vortex model

for j=1:N
   for i=1:M2+5
      Inp_a2(i,j)=a(1,j)*sin(4*pi*dt*i/T1);
      Inp_a_dot2(i,j)=adot(1,j)*(4*pi/T1)*cos(4*pi*dt*i/T1);
      Inp_h_dot2(i,j)=0;
   end;
end;

for j=1:N
   CompareCirc(1,j)=0.0;
   CompCirc2(1,j)=0.0;
   for i=2:M+1
      CompareCirc(i,j)=(Uinf/beta)*((Uinf*dt-c/4)*Inp_a2(i-1,j)+
                       (c/4)*Inp_a2(i,j)-(dt/(pi*c))*
                       CompareCirc(i-1,j))+CompareCirc(i-1,j);
   end;
end;

figure(2);
plot(time,CompareCirc(:,N));
title('Response at tip of lumped vortex model to sinusoidal pitching');
xlabel('Time (sec)');
ylabel('Circulation');
hold on
plot(time2,Resp(:,N),':');

etime(clock,time0)
```

B.2 Deflected Wing Generation Program

```
C    FOILGEN
C
C Written by Randal E. Guendel
C Technology Laboratory for Advanced Composites (TELAC)
C Massachusetts Institute of Technology
C
C 5 October 1999
C
C Program to generate an arbitrarily deflected and twisted
C rectangular wing comprised of NACA 00xx airfoil sections.
C Output is two files: 'wing.txt' is the PMARC-formatted
C geometry and 'curve.txt' gives the displacement and rotation
C for each section.
C
C Notes:
C
C 1. Currently only wing deflection has been implemented.
C 2. Currently the wing root leading edge is at (0.0, 0.0, 0.0)
C 3. The wing tip deflection ratio (TIP) actually represents
C    the deflection if the wing were extended to y=b.  When
C    the wing is deflected, the actual wing tip deflection
C    will be somewhat smaller than TIP.
C 4. Any consitent set of units may be used.
C
C
      PROGRAM FOILGEN
C
C VARIABLE DECLARATIONS------------------
C
      REAL X0,Y0,Z0,X,Y,Z,SPN,CHD,THK,TIP,TWIST,A(5),CHD_X,
     +     CHD_Y,CHD_Z,SURF_Z,THETA,L,TY,TRAT
      INTEGER NPANX,NPANY,XSPACE,YSPACE,I,J
      CHARACTER TAPERED
C
C DATA INPUT
C
      WRITE(*,*)'AIRFOIL AND WING DATA'
      WRITE(*,*)'------- --- ---- ----'
C      WRITE(*,*)'ENTER THE STARTING COORDINATE (X,Y,Z)'
C      READ(*,*)X0,Y0,Z0
      WRITE(*,*)'ENTER THE WING SPAN'
      READ(*,*)SPN
      WRITE(*,*)'ENTER THE CHORD LENGTH'
      READ(*,*)CHD
      WRITE(*,*)'ENTER THE THICKNESS RATIO (DECIMAL)'
      READ(*,*)THK
      WRITE(*,*)'TAPERED WING? (Y or N)'
      READ(*,*)TAPERED
      IF (TAPERED.EQ.'Y')THEN
         WRITE(*,*)'ENTER NORMALIZED LOCATION OF START OF TAPER'
         READ(*,*)TY
         WRITE(*,*)'ENTER TAPER RATIO'
         READ(*,*)TRAT
      END IF
      WRITE(*,*)'DISCRETIZATION DATA'
      WRITE(*,*)'-------------- ----'
      WRITE(*,*)'ENTER NUMBER OF SPANWISE PANELS'
      READ(*,*)NPANY
      WRITE(*,*)'ENTER NUMBER OF CHORDWISE PANELS'
      READ(*,*)NPANX
      WRITE(*,*)'ENTER SPANWISE SPACING'
      WRITE(*,*)'  FULL COSINE=      0     INC. HALF COSINE= 1,'
      WRITE(*,*)'  DEC. HALF COSINE= 2     EQUAL=            3'
      READ(*,*)YSPACE
      WRITE(*,*)'ENTER CHORDWISE SPACING'
      READ(*,*)XSPACE
      WRITE(*,*)'DEFLECTION DATA'
      WRITE(*,*)'---------- ----'
      WRITE(*,*)'ENTER TIP DEFLECTION RATIO (0-1)'
      READ(*,*)TIP
      WRITE(*,*)'ENTER TIP TWIST (DEGREES)'
      READ(*,*)TWIST
C
C AIRFOIL POLYNOMIAL COEFFICIENTS
C
      A(1)=0.2969
      A(2)=-0.1260
      A(3)=-0.3516
      A(4)=0.2843
      A(5)=-0.1015
C
C ADJUST FOR THICKNESS OTHER THAN 0.20
C
      DO I=1,5
         A(I)=THK*A(I)/0.20
      END DO
```

```
C
C OPEN OUTPUT FILES
C
       OPEN(UNIT=9,FILE='wing.txt',STATUS='NEW')
       OPEN(UNIT=10,FILE='curve.txt',STATUS='NEW')
C
C WRITE PATCH HEADER
C
       WRITE(9,501)
       WRITE(9,506)
C
C GENERATE DEFLECTED AIRFOIL SECTIONS
C
       DO J=0,NPANY
          L=J*(SPN/NPANY)
          CHD_Y=YPRIME(L,ABS(TIP),SPN)
          CHD_Z=SPN*TIP*(CHD_Y/SPN)**2
          IF ((TIP.NE.0.0).AND.(CHD_Y.NE.0.0)) THEN
             THETA=ATAN(2*(TIP/SPN)*CHD_Y)
          ELSE
             THETA=0.0
          END IF
C
C WRITE SECTION HEADER
          IF(J.EQ.NPANY)THEN
             WRITE(9,505)NPANY,YSPACE
          ELSE
             WRITE(9,502)
          END IF
C
C WRITE DISPLACEMENT AND ROTATION DATA
          WRITE(10,520)J,CHD_Y,THETA
C
C LOWER SURFACE, FROM TE TO LE
          DO I=0,NPANX
             CHD_X=CHD-I*(CHD/NPANX)
             SURF_Z=-POLY(CHD_X,CHD,A)
             X=CHD_X
             Y=CHD_Y-SURF_Z*SIN(THETA)*COS(THETA)
             Z=CHD_Z+SURF_Z-SURF_Z*SIN(THETA)*SIN(THETA)
             WRITE(9,510)X,Y,Z
          END DO
          WRITE(9,503)NPANX,XSPACE
C
C UPPER SURFACE, FROM LE TO TE
```

```
          DO I=0,NPANX
             CHD_X=I*(CHD/NPANX)
             SURF_Z=POLY(CHD_X,CHD,A)
             X=CHD_X
             Y=CHD_Y-SURF_Z*SIN(THETA)*COS(THETA)
             Z=CHD_Z+SURF_Z-SURF_Z*SIN(THETA)*SIN(THETA)
             WRITE(9,510)X,Y,Z
          END DO
          WRITE(9,504)NPANX,XSPACE

       END DO

C
C FORMAT STATEMENTS
C
 501   FORMAT(' &PATCH1  IREV=0  IDPAT=1  MAKE=0  KCOMP=1'/
     +        '          KASS=1  IPATSYM=0  IPATCOP=0           /')
 502   FORMAT(' &SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00'/
     +        '         ALF=0.0  THETA=0.0  INMODE=4'/
     +        '         TNODS=0  TNPS=0  TINTS=0                /')
 503   FORMAT(' &BPNODE  TNODE=2  TNPC=',I2,'  TINTC=',I2,'     /')
 504   FORMAT(' &BPNODE  TNODE=3  TNPC=',I2,'  TINTC=',I2,'     /')
 505   FORMAT(' &SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00'/
     +        '         ALF=0.0  THETA=0.0  INMODE=4'/
     +        '         TNODS=3  TNPS=',I2,'  TINTS=',I3,'      /')
 506   FORMAT('   WING')
 510   FORMAT(5X,F10.5,5X,F10.5,5X,F10.5)
 520   FORMAT(I4,5X,F10.5,5X,F10.5)
       STOP
       END


C FUNCTION POLY
C
C AIRFOIL POLYNOMIAL (FROM NASA TM 4741, 'COMPUTER PROGRAM TO OBTAIN
C ORDINATES FOR NACA AIRFOILS', DEC 96)
C
       REAL FUNCTION POLY(X,CHD,A)

       REAL X,CHD
       REAL A(5)

       POLY=CHD*(A(1)*(X/CHD)**0.5+A(2)*(X/CHD)+A(3)*(X/CHD)**2+
     +      A(4)*(X/CHD)**3+A(5)*(X/CHD)**4)
       RETURN
       END

C
C FUNCTION YPRIME
C
C COMPUTES THE DEFLECTED Y-POSITION OF THE CURRENT AIRFOIL SECTION
C CHORDLINE
C
      REAL FUNCTION YPRIME(B,A,SPN)

      REAL A,B,SPN
      INTEGER I
      REAL LEN,W
      I=0
      LEN=0.0
      W=0.001
      IF (B.EQ.0.0) THEN
         YPRIME=0.0
      ELSE
         DO WHILE(LEN.LT.B)
            LEN=LEN+TRAPAREA(W,I,A,SPN)
            I=I+1
         END DO
         IF (LEN.EQ.B) THEN
            YPRIME=W*I
         ELSE
            YPRIME=(W*I+W*(I+1))/2
         END IF
      END IF
      RETURN
      END


C
C FUNCTION TRAPAREA
C
C COMPUTES THE AREA OF THE CURRENT TRAPEZOID FOR THE COMPUTATION
C OF THE CURVE LENGTH BY THE TRAPEZOIDAL RULE
C
      REAL FUNCTION TRAPAREA(W,I,A,SPN)

      REAL W,A
      INTEGER I
      REAL X1,X2,F1,F2
      X1=W*I
```

```
X2=W*(I+1)
F1=SQRT(1+(2*A/SPN)**2*X1**2)
F2=SQRT(1+(2*A/SPN)**2*X2**2)
TRAPAREA=W/2*(F1+F2)
RETURN
END
```

Appendix C

Input Files

C.1 SHAR1 PMARC Input File for 29% Bending

```
  BASELINE WING IN 1st BENDING MODE
&BINP2   LSTINP=2,    LSTOUT=1,    LSTFRQ=0,    LENRUN=0,
         LPLTYP=1,     /
&BINP3   LSTGEO=1,    LSTNAB=0,    LSTWAK=2,    LSTCPV=0,    /
&BINP4   MAXIT=500,    SOLRES=0.0005     /
&BINP5   NTSTPS=125     DTSTEP=0.01      /
&BINP6   RSYM=0.0     RGPR=0.0     RFF=5.0   RCORES=0.0080
         RCOREW=0.0080  /
&BINP7   VINF=10.00    VSOUND=340.0   /
&BINP8   ALDEG=0.0    YAWDEG=0.0   PHIDOT=0.0   THEDOT=0.0
         PSIDOT=0.0    /
&BINP8A  PHIMAX= 0.0  THEMAX= 3.0  PSIMAX=0.0
         WRX=0.0      WRY=2.0      WRZ=0.0    /
&BINP8B  DXMAX=0.0    DYMAX=0.0    DZMAX=0.0
         WTX=0.0      WTY=0.0      WTZ=0.0   /
&BINP9   CBAR=0.2    SREF=0.80  SSPAN= 2.00
         RMPX=0.00   RMPY=0.00    RMPZ=0.00  /
&BINP10  NORSET=1     NBCHGE=1     NCZONE=0
         NCZPCH=0     CZDUB=0.0    VREF=0.0   /
&BINP11  NORPCH=0   NORF=0     NORL=0
         NOCF=0     NOCL=0    VNORM=0.0    /
&BINP12  KPAN=0    KSIDE=0   NEWNAB=0    NEWSID=0      /
&BINP13  NBLIT=0 /

&ASEM1   ASEMX=0.00       ASEMY=0.00       ASEMZ=0.00
```

```
         ASCAL=1.00        ATHET=0.00        NODEA=5  /
&ASEM2   APXX=0.00         APYY=0.00         APZZ=0.00
         AHXX=0.00         AHYY=1.00         AHZZ=0.00 /
&COMP1   COMPX= 0.0000      COMPY= 0.0000      COMPZ= 0.0000
         CSCAL=  1.000      CTHET=   0.0       NODEC= 5   /
&COMP2   CPXX= 0.0000       CPYY= 0.0000       CPZZ= 0.0000
         CHXX= 0.0000       CHYY=  1.000       CHZZ= 0.0000  /

&PATCH1  IREV=0  IDPAT=1  MAKE=0  KCOMP=1
         KASS=1  IPATSYM=0  IPATCOP=0          /
  WING
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                /
       0.20000        0.00000       -0.00029
       0.18667        0.00000       -0.00239
       0.17333        0.00000       -0.00433
       0.16000        0.00000       -0.00612
       0.14667        0.00000       -0.00777
       0.13333        0.00000       -0.00929
       0.12000        0.00000       -0.01065
       0.10667        0.00000       -0.01183
       0.09333        0.00000       -0.01281
       0.08000        0.00000       -0.01354
       0.06667        0.00000       -0.01395
       0.05333        0.00000       -0.01394
       0.04000        0.00000       -0.01339
       0.02667        0.00000       -0.01204
       0.01333        0.00000       -0.00935
       0.00000        0.00000        0.00000
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        0.00000        0.00000
       0.01333        0.00000        0.00935
       0.02667        0.00000        0.01204
       0.04000        0.00000        0.01339
       0.05333        0.00000        0.01394
       0.06667        0.00000        0.01395
       0.08000        0.00000        0.01354
       0.09333        0.00000        0.01281
       0.10667        0.00000        0.01183
       0.12000        0.00000        0.01065
       0.13333        0.00000        0.00929
       0.14667        0.00000        0.00777
       0.16000        0.00000        0.00612
       0.17333        0.00000        0.00433
       0.18667       0.00000       0.00239
       0.20000       0.00000       0.00029
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                /
       0.20000       0.08052       0.00078
       0.18667       0.08063      -0.00131
       0.17333       0.08073      -0.00325
       0.16000       0.08082      -0.00503
       0.14667       0.08091      -0.00668
       0.13333       0.08099      -0.00819
       0.12000       0.08106      -0.00955
       0.10667       0.08113      -0.01073
       0.09333       0.08118      -0.01171
       0.08000       0.08122      -0.01243
       0.06667       0.08124      -0.01284
       0.05333       0.08124      -0.01283
       0.04000       0.08121      -0.01228
       0.02667       0.08114      -0.01094
       0.01333       0.08100      -0.00825
       0.00000       0.08050       0.00107
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000       0.08050       0.00107
       0.01333       0.08000       0.01039
       0.02667       0.07986       0.01308
       0.04000       0.07979       0.01442
       0.05333       0.07976       0.01497
       0.06667       0.07976       0.01498
       0.08000       0.07978       0.01457
       0.09333       0.07982       0.01385
       0.10667       0.07987       0.01287
       0.12000       0.07994       0.01169
       0.13333       0.08001       0.01033
       0.14667       0.08009       0.00882
       0.16000       0.08018       0.00717
       0.17333       0.08027       0.00539
       0.18667       0.08037       0.00345
       0.20000       0.08048       0.00136
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                /
       0.20000       0.16053       0.00396
       0.18667       0.16075       0.00189
```

```
         0.17333        0.16095       -0.00003
         0.16000        0.16114       -0.00180
         0.14667        0.16131       -0.00344
         0.13333        0.16147       -0.00493
         0.12000        0.16162       -0.00628
         0.10667        0.16174       -0.00745
         0.09333        0.16184       -0.00842
         0.08000        0.16192       -0.00914
         0.06667        0.16196       -0.00954
         0.05333        0.16196       -0.00954
         0.04000        0.16190       -0.00899
         0.02667        0.16176       -0.00766
         0.01333        0.16148       -0.00499
         0.00000        0.16050        0.00425
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
         0.00000        0.16050        0.00425
         0.01333        0.15952        0.01350
         0.02667        0.15924        0.01616
         0.04000        0.15910        0.01749
         0.05333        0.15904        0.01804
         0.06667        0.15904        0.01804
         0.08000        0.15908        0.01764
         0.09333        0.15916        0.01692
         0.10667        0.15926        0.01595
         0.12000        0.15938        0.01478
         0.13333        0.15953        0.01343
         0.14667        0.15969        0.01194
         0.16000        0.15986        0.01030
         0.17333        0.16005        0.00853
         0.18667        0.16025        0.00661
         0.20000        0.16047        0.00454
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
         ALF=0.0  THETA=0.0  INMODE=4
         TNODS=0  TNPS=0  TINTS=0                /
         0.20000        0.24055        0.00926
         0.18667        0.24087        0.00721
         0.17333        0.24117        0.00532
         0.16000        0.24145        0.00357
         0.14667        0.24170        0.00196
         0.13333        0.24194        0.00048
         0.12000        0.24215       -0.00084
         0.10667        0.24233       -0.00200
         0.09333        0.24248       -0.00296
         0.08000        0.24260       -0.00366
```

```
       0.06667        0.24266       -0.00406
       0.05333        0.24266       -0.00406
       0.04000        0.24257       -0.00351
       0.02667        0.24236       -0.00220
       0.01333        0.24195        0.00042
       0.00000        0.24050        0.00954
&BPNODE  TNODE=2  TNPC=15  TINTC= 0      /
       0.00000        0.24050        0.00954
       0.01333        0.23905        0.01866
       0.02667        0.23864        0.02129
       0.04000        0.23843        0.02260
       0.05333        0.23834        0.02314
       0.06667        0.23834        0.02315
       0.08000        0.23840        0.02275
       0.09333        0.23852        0.02204
       0.10667        0.23867        0.02109
       0.12000        0.23885        0.01993
       0.13333        0.23906        0.01860
       0.14667        0.23930        0.01713
       0.16000        0.23955        0.01551
       0.17333        0.23983        0.01377
       0.18667        0.24013        0.01188
       0.20000        0.24045        0.00983
&BPNODE  TNODE=3  TNPC=15  TINTC= 0      /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                 /
       0.20000        0.32056        0.01667
       0.18667        0.32098        0.01466
       0.17333        0.32138        0.01281
       0.16000        0.32174        0.01109
       0.14667        0.32207        0.00951
       0.13333        0.32238        0.00806
       0.12000        0.32266        0.00676
       0.10667        0.32290        0.00562
       0.09333        0.32309        0.00468
       0.08000        0.32324        0.00399
       0.06667        0.32332        0.00360
       0.05333        0.32332        0.00360
       0.04000        0.32321        0.00413
       0.02667        0.32294        0.00542
       0.01333        0.32239        0.00800
       0.00000        0.32050        0.01695
&BPNODE  TNODE=2  TNPC=15  TINTC= 0      /
       0.00000        0.32050        0.01695
       0.01333        0.31861        0.02590
       0.02667        0.31806        0.02847
       0.04000        0.31779        0.02976
       0.05333        0.31768        0.03029
       0.06667        0.31768        0.03030
       0.08000        0.31776        0.02991
       0.09333        0.31791        0.02921
       0.10667        0.31810        0.02828
       0.12000        0.31834        0.02714
       0.13333        0.31862        0.02584
       0.14667        0.31893        0.02439
       0.16000        0.31926        0.02281
       0.17333        0.31962        0.02109
       0.18667        0.32002        0.01924
       0.20000        0.32044        0.01723
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                /
       0.20000        0.39957        0.02606
       0.18667        0.40009        0.02410
       0.17333        0.40057        0.02229
       0.16000        0.40101        0.02061
       0.14667        0.40142        0.01906
       0.13333        0.40179        0.01765
       0.12000        0.40213        0.01638
       0.10667        0.40242        0.01527
       0.09333        0.40266        0.01435
       0.08000        0.40284        0.01367
       0.06667        0.40294        0.01329
       0.05333        0.40294        0.01330
       0.04000        0.40280        0.01382
       0.02667        0.40247        0.01508
       0.01333        0.40180        0.01759
       0.00000        0.39950        0.02633
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        0.39950        0.02633
       0.01333        0.39720        0.03508
       0.02667        0.39653        0.03759
       0.04000        0.39620        0.03885
       0.05333        0.39606        0.03937
       0.06667        0.39606        0.03937
       0.08000        0.39616        0.03899
       0.09333        0.39634        0.03832
       0.10667        0.39658        0.03740
       0.12000        0.39687        0.03629
       0.13333        0.39721        0.03502
       0.14667        0.39758        0.03360
       0.16000        0.39799        0.03206
       0.17333        0.39843        0.03038
       0.18667        0.39891        0.02857
       0.20000        0.39943        0.02661
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                /
       0.20000        0.47958        0.03767
       0.18667        0.48019        0.03576
       0.17333        0.48075        0.03400
       0.16000        0.48126        0.03237
       0.14667        0.48174        0.03087
       0.13333        0.48217        0.02949
       0.12000        0.48256        0.02826
       0.10667        0.48290        0.02718
       0.09333        0.48319        0.02629
       0.08000        0.48340        0.02563
       0.06667        0.48351        0.02526
       0.05333        0.48351        0.02526
       0.04000        0.48335        0.02577
       0.02667        0.48296        0.02699
       0.01333        0.48219        0.02944
       0.00000        0.47950        0.03794
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        0.47950        0.03794
       0.01333        0.47681        0.04643
       0.02667        0.47604        0.04888
       0.04000        0.47565        0.05011
       0.05333        0.47549        0.05061
       0.06667        0.47549        0.05061
       0.08000        0.47561        0.05024
       0.09333        0.47581        0.04958
       0.10667        0.47610        0.04869
       0.12000        0.47644        0.04762
       0.13333        0.47683        0.04638
       0.14667        0.47726        0.04500
       0.16000        0.47774        0.04350
       0.17333        0.47826        0.04187
       0.18667        0.47881        0.04011
       0.20000        0.47942        0.03820
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
```

```
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                /
       0.20000        0.55760        0.05102
       0.18667        0.55827        0.04918
       0.17333        0.55890        0.04747
       0.16000        0.55948        0.04589
       0.14667        0.56002        0.04444
       0.13333        0.56051        0.04310
       0.12000        0.56095        0.04190
       0.10667        0.56134        0.04086
       0.09333        0.56165        0.04000
       0.08000        0.56189        0.03936
       0.06667        0.56202        0.03900
       0.05333        0.56202        0.03900
       0.04000        0.56184        0.03949
       0.02667        0.56140        0.04068
       0.01333        0.56053        0.04305
       0.00000        0.55750        0.05128
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        0.55750        0.05128
       0.01333        0.55447        0.05952
       0.02667        0.55360        0.06189
       0.04000        0.55316        0.06307
       0.05333        0.55298        0.06356
       0.06667        0.55298        0.06357
       0.08000        0.55311        0.06321
       0.09333        0.55335        0.06257
       0.10667        0.55366        0.06171
       0.12000        0.55405        0.06066
       0.13333        0.55449        0.05946
       0.14667        0.55498        0.05813
       0.16000        0.55552        0.05667
       0.17333        0.55610        0.05509
       0.18667        0.55673        0.05339
       0.20000        0.55740        0.05154
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                /
       0.20000        0.63660        0.06660
       0.18667        0.63735        0.06481
       0.17333        0.63805        0.06317
       0.16000        0.63869        0.06164
       0.14667        0.63928        0.06024
       0.13333       0.63982       0.05895
       0.12000       0.64030       0.05780
       0.10667       0.64073       0.05679
       0.09333       0.64108       0.05595
       0.08000       0.64133       0.05534
       0.06667       0.64148       0.05499
       0.05333       0.64148       0.05500
       0.04000       0.64128       0.05547
       0.02667       0.64080       0.05661
       0.01333       0.63984       0.05890
       0.00000       0.63650       0.06685
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000       0.63650       0.06685
       0.01333       0.63316       0.07479
       0.02667       0.63220       0.07708
       0.04000       0.63172       0.07823
       0.05333       0.63152       0.07870
       0.06667       0.63152       0.07870
       0.08000       0.63167       0.07836
       0.09333       0.63192       0.07774
       0.10667       0.63227       0.07691
       0.12000       0.63270       0.07590
       0.13333       0.63318       0.07474
       0.14667       0.63372       0.07346
       0.16000       0.63431       0.07205
       0.17333       0.63495       0.07053
       0.18667       0.63565       0.06888
       0.20000       0.63640       0.06710
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1   STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                  /
       0.20000       0.71461       0.08399
       0.18667       0.71542       0.08228
       0.17333       0.71617       0.08069
       0.16000       0.71686       0.07923
       0.14667       0.71750       0.07787
       0.13333       0.71808       0.07664
       0.12000       0.71861       0.07552
       0.10667       0.71907       0.07455
       0.09333       0.71944       0.07375
       0.08000       0.71972       0.07316
       0.06667       0.71988       0.07282
       0.05333       0.71988       0.07283
       0.04000       0.71966       0.07328
       0.02667        0.71914        0.07438
       0.01333        0.71811        0.07659
       0.00000        0.71450        0.08423
&BPNODE  TNODE=2  TNPC=15  TINTC= 0      /
       0.00000        0.71450        0.08423
       0.01333        0.71089        0.09188
       0.02667        0.70986        0.09408
       0.04000        0.70934        0.09519
       0.05333        0.70912        0.09564
       0.06667        0.70912        0.09564
       0.08000        0.70928        0.09531
       0.09333        0.70956        0.09472
       0.10667        0.70993        0.09392
       0.12000        0.71039        0.09294
       0.13333        0.71092        0.09183
       0.14667        0.71150        0.09059
       0.16000        0.71214        0.08924
       0.17333        0.71283        0.08777
       0.18667        0.71358        0.08619
       0.20000        0.71439        0.08447
&BPNODE  TNODE=3  TNPC=15  TINTC= 0      /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                 /
       0.20000        0.79262        0.10340
       0.18667        0.79348        0.10175
       0.17333        0.79428        0.10023
       0.16000        0.79501        0.09882
       0.14667        0.79569        0.09752
       0.13333        0.79631        0.09634
       0.12000        0.79687        0.09527
       0.10667        0.79736        0.09434
       0.09333        0.79776        0.09357
       0.08000        0.79806        0.09300
       0.06667        0.79823        0.09268
       0.05333        0.79823        0.09268
       0.04000        0.79800        0.09312
       0.02667        0.79744        0.09418
       0.01333        0.79634        0.09629
       0.00000        0.79250        0.10363
&BPNODE  TNODE=2  TNPC=15  TINTC= 0      /
       0.00000        0.79250        0.10363
       0.01333        0.78866        0.11097
       0.02667        0.78756        0.11308
       0.04000        0.78700        0.11414
       0.05333        0.78677        0.11458
       0.06667        0.78677        0.11458
       0.08000        0.78694        0.11426
       0.09333        0.78724        0.11369
       0.10667        0.78764        0.11292
       0.12000        0.78813        0.11199
       0.13333        0.78869        0.11092
       0.14667        0.78931        0.10973
       0.16000        0.78999        0.10844
       0.17333        0.79072        0.10703
       0.18667        0.79152        0.10551
       0.20000        0.79238        0.10386
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                        /
       0.20000        0.86963        0.12452
       0.18667        0.87053        0.12295
       0.17333        0.87137        0.12149
       0.16000        0.87214        0.12014
       0.14667        0.87286        0.11890
       0.13333        0.87351        0.11776
       0.12000        0.87410        0.11674
       0.10667        0.87461        0.11584
       0.09333        0.87503        0.11511
       0.08000        0.87535        0.11456
       0.06667        0.87552        0.11425
       0.05333        0.87552        0.11426
       0.04000        0.87528        0.11467
       0.02667        0.87470        0.11569
       0.01333        0.87354        0.11771
       0.00000        0.86950        0.12475
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        0.86950        0.12475
       0.01333        0.86546        0.13178
       0.02667        0.86430        0.13380
       0.04000        0.86372        0.13482
       0.05333        0.86348        0.13523
       0.06667        0.86348        0.13524
       0.08000        0.86365        0.13493
       0.09333        0.86397        0.13438
       0.10667        0.86439        0.13365
       0.12000        0.86490        0.13275
       0.13333        0.86549        0.13173
       0.14667        0.86614        0.13059
       0.16000        0.86686        0.12935
       0.17333        0.86763        0.12800
       0.18667        0.86847        0.12654
       0.20000        0.86937        0.12497
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                  /
       0.20000        0.94563        0.14729
       0.18667        0.94657        0.14578
       0.17333        0.94744        0.14439
       0.16000        0.94825        0.14310
       0.14667        0.94899        0.14191
       0.13333        0.94967        0.14082
       0.12000        0.95028        0.13984
       0.10667        0.95082        0.13899
       0.09333        0.95126        0.13828
       0.08000        0.95158        0.13776
       0.06667        0.95176        0.13747
       0.05333        0.95176        0.13747
       0.04000        0.95151        0.13787
       0.02667        0.95091        0.13884
       0.01333        0.94970        0.14078
       0.00000        0.94550        0.14751
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        0.94550        0.14751
       0.01333        0.94130        0.15423
       0.02667        0.94009        0.15617
       0.04000        0.93949        0.15714
       0.05333        0.93924        0.15754
       0.06667        0.93924        0.15754
       0.08000        0.93942        0.15725
       0.09333        0.93974        0.15673
       0.10667        0.94018        0.15602
       0.12000        0.94072        0.15517
       0.13333        0.94133        0.15419
       0.14667        0.94201        0.15310
       0.16000        0.94275        0.15191
       0.17333        0.94356        0.15062
       0.18667        0.94443        0.14923
       0.20000        0.94537        0.14772
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                  /
       0.20000        1.02164        0.17197
       0.18667        1.02261        0.17053
       0.17333        1.02351        0.16920
       0.16000        1.02434        0.16796
       0.14667        1.02510        0.16683
       0.13333        1.02580        0.16579
       0.12000        1.02644        0.16485
       0.10667        1.02699        0.16404
       0.09333        1.02744        0.16336
       0.08000        1.02778        0.16286
       0.06667        1.02796        0.16258
       0.05333        1.02796        0.16259
       0.04000        1.02771        0.16297
       0.02667        1.02708        0.16389
       0.01333        1.02583        0.16574
       0.00000        1.02150        0.17217
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        1.02150        0.17217
       0.01333        1.01717        0.17860
       0.02667        1.01592        0.18045
       0.04000        1.01529        0.18138
       0.05333        1.01504        0.18176
       0.06667        1.01504        0.18176
       0.08000        1.01522        0.18148
       0.09333        1.01556        0.18098
       0.10667        1.01601        0.18031
       0.12000        1.01656        0.17949
       0.13333        1.01720        0.17856
       0.14667        1.01790        0.17752
       0.16000        1.01866        0.17638
       0.17333        1.01949        0.17515
       0.18667        1.02039        0.17381
       0.20000        1.02136        0.17237
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                /
       0.20000        1.09764        0.19855
       0.18667        1.09864        0.19718
       0.17333        1.09956        0.19590
       0.16000        1.10041        0.19473
       0.14667        1.10119        0.19364
       0.13333        1.10191        0.19265
       0.12000        1.10256        0.19176
       0.10667        1.10312        0.19098
       0.09333       1.10359       0.19034
       0.08000       1.10393       0.18986
       0.06667       1.10413       0.18960
       0.05333       1.10412       0.18960
       0.04000       1.10386       0.18996
       0.02667       1.10322       0.19085
       0.01333       1.10194       0.19261
       0.00000       1.09750       0.19874
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000       1.09750       0.19874
       0.01333       1.09306       0.20488
       0.02667       1.09178       0.20664
       0.04000       1.09114       0.20752
       0.05333       1.09088       0.20789
       0.06667       1.09087       0.20789
       0.08000       1.09107       0.20762
       0.09333       1.09141       0.20715
       0.10667       1.09188       0.20651
       0.12000       1.09244       0.20573
       0.13333       1.09309       0.20483
       0.14667       1.09381       0.20384
       0.16000       1.09459       0.20276
       0.17333       1.09544       0.20158
       0.18667       1.09636       0.20031
       0.20000       1.09736       0.19894
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                /
       0.20000       1.17264       0.22665
       0.18667       1.17366       0.22534
       0.17333       1.17459       0.22413
       0.16000       1.17546       0.22301
       0.14667       1.17626       0.22197
       0.13333       1.17700       0.22103
       0.12000       1.17765       0.22018
       0.10667       1.17823       0.21943
       0.09333       1.17870       0.21882
       0.08000       1.17905       0.21837
       0.06667       1.17925       0.21811
       0.05333       1.17925       0.21811
       0.04000       1.17898       0.21846
       0.02667       1.17833       0.21930
       0.01333       1.17702       0.22099
       0.00000       1.17250       0.22683
```

```
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        1.17250        0.22683
       0.01333        1.16798        0.23268
       0.02667        1.16667        0.23437
       0.04000        1.16602        0.23521
       0.05333        1.16575        0.23556
       0.06667        1.16575        0.23556
       0.08000        1.16595        0.23530
       0.09333        1.16630        0.23485
       0.10667        1.16677        0.23424
       0.12000        1.16735        0.23349
       0.13333        1.16800        0.23264
       0.14667        1.16874        0.23170
       0.16000        1.16954        0.23066
       0.17333        1.17041        0.22954
       0.18667        1.17134        0.22833
       0.20000        1.17236        0.22702
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                /
       0.20000        1.24664        0.25620
       0.18667        1.24767        0.25495
       0.17333        1.24862        0.25379
       0.16000        1.24950        0.25272
       0.14667        1.25031        0.25173
       0.13333        1.25106        0.25083
       0.12000        1.25172        0.25002
       0.10667        1.25231        0.24931
       0.09333        1.25279        0.24873
       0.08000        1.25314        0.24830
       0.06667        1.25334        0.24805
       0.05333        1.25334        0.24806
       0.04000        1.25307        0.24839
       0.02667        1.25241        0.24919
       0.01333        1.25109        0.25080
       0.00000        1.24650        0.25637
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        1.24650        0.25637
       0.01333        1.24191        0.26195
       0.02667        1.24059        0.26355
       0.04000        1.23993        0.26435
       0.05333        1.23966        0.26469
       0.06667        1.23966        0.26469
       0.08000        1.23986        0.26445
       0.09333        1.24021        0.26401
       0.10667        1.24069        0.26343
       0.12000        1.24128        0.26272
       0.13333        1.24194        0.26191
       0.14667        1.24269        0.26101
       0.16000        1.24350        0.26002
       0.17333        1.24438        0.25895
       0.18667        1.24533        0.25780
       0.20000        1.24636        0.25655
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                       /
       0.20000        1.32065        0.28755
       0.18667        1.32168        0.28636
       0.17333        1.32264        0.28525
       0.16000        1.32353        0.28424
       0.14667        1.32435        0.28330
       0.13333        1.32510        0.28244
       0.12000        1.32577        0.28166
       0.10667        1.32636        0.28099
       0.09333        1.32685        0.28043
       0.08000        1.32721        0.28002
       0.06667        1.32741        0.27979
       0.05333        1.32741        0.27979
       0.04000        1.32713        0.28011
       0.02667        1.32646        0.28087
       0.01333        1.32513        0.28240
       0.00000        1.32050        0.28771
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        1.32050        0.28771
       0.01333        1.31587        0.29303
       0.02667        1.31454        0.29456
       0.04000        1.31387        0.29532
       0.05333        1.31359        0.29564
       0.06667        1.31359        0.29564
       0.08000        1.31379        0.29541
       0.09333        1.31415        0.29500
       0.10667        1.31464        0.29444
       0.12000        1.31523        0.29377
       0.13333        1.31590        0.29299
       0.14667        1.31665        0.29213
       0.16000        1.31747        0.29119
       0.17333        1.31836        0.29017
       0.18667        1.31932        0.28907
       0.20000        1.32035        0.28788
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0               /
       0.20000        1.39365        0.32024
       0.18667        1.39469        0.31911
       0.17333        1.39566        0.31806
       0.16000        1.39655        0.31709
       0.14667        1.39737        0.31619
       0.13333        1.39813        0.31537
       0.12000        1.39881        0.31464
       0.10667        1.39940        0.31399
       0.09333        1.39988        0.31346
       0.08000        1.40025        0.31307
       0.06667        1.40045        0.31285
       0.05333        1.40045        0.31285
       0.04000        1.40017        0.31315
       0.02667        1.39950        0.31388
       0.01333        1.39816        0.31534
       0.00000        1.39350        0.32040
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        1.39350        0.32040
       0.01333        1.38884        0.32547
       0.02667        1.38750        0.32693
       0.04000        1.38683        0.32766
       0.05333        1.38655        0.32796
       0.06667        1.38655        0.32796
       0.08000        1.38675        0.32774
       0.09333        1.38712        0.32735
       0.10667        1.38760        0.32682
       0.12000        1.38819        0.32617
       0.13333        1.38887        0.32544
       0.14667        1.38963        0.32462
       0.16000        1.39045        0.32372
       0.17333        1.39134        0.32275
       0.18667        1.39231        0.32170
       0.20000        1.39335        0.32056
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0               /
       0.20000        1.46565        0.35422
       0.18667        1.46669        0.35313
       0.17333        1.46766        0.35213
       0.16000        1.46856        0.35121
       0.14667        1.46939        0.35035
       0.13333        1.47014        0.34957
       0.12000        1.47082        0.34887
       0.10667        1.47141        0.34825
       0.09333        1.47190        0.34775
       0.08000        1.47227        0.34737
       0.06667        1.47247        0.34716
       0.05333        1.47247        0.34717
       0.04000        1.47219        0.34745
       0.02667        1.47152        0.34815
       0.01333        1.47017        0.34954
       0.00000        1.46550        0.35437
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        1.46550        0.35437
       0.01333        1.46083        0.35920
       0.02667        1.45948        0.36059
       0.04000        1.45881        0.36129
       0.05333        1.45853        0.36157
       0.06667        1.45853        0.36157
       0.08000        1.45873        0.36136
       0.09333        1.45910        0.36099
       0.10667        1.45959        0.36048
       0.12000        1.46018        0.35987
       0.13333        1.46086        0.35917
       0.14667        1.46161        0.35839
       0.16000        1.46244        0.35753
       0.17333        1.46334        0.35660
       0.18667        1.46431        0.35560
       0.20000        1.46535        0.35452
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                      /
       0.20000        1.53765        0.38990
       0.18667        1.53870        0.38887
       0.17333        1.53966        0.38791
       0.16000        1.54056        0.38703
       0.14667        1.54139        0.38621
       0.13333        1.54214        0.38547
       0.12000        1.54282        0.38480
       0.10667        1.54342        0.38421
       0.09333        1.54391        0.38373
       0.08000        1.54427        0.38337
       0.06667        1.54447        0.38317
       0.05333        1.54447        0.38318
       0.04000        1.54419        0.38345
       0.02667        1.54352        0.38411
       0.01333        1.54217        0.38544
       0.00000        1.53750        0.39004
&BPNODE  TNODE=2  TNPC=15  TINTC= 0      /
       0.00000        1.53750        0.39004
       0.01333        1.53283        0.39465
       0.02667        1.53148        0.39598
       0.04000        1.53081        0.39664
       0.05333        1.53053        0.39691
       0.06667        1.53053        0.39692
       0.08000        1.53073        0.39672
       0.09333        1.53109        0.39636
       0.10667        1.53158        0.39588
       0.12000        1.53218        0.39529
       0.13333        1.53286        0.39462
       0.14667        1.53361        0.39387
       0.16000        1.53444        0.39306
       0.17333        1.53534        0.39218
       0.18667        1.53630        0.39122
       0.20000        1.53735        0.39019
&BPNODE  TNODE=3  TNPC=15  TINTC= 0      /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
         ALF=0.0  THETA=0.0  INMODE=4
         TNODS=0  TNPS=0  TINTS=0                /
       0.20000        1.60865        0.42676
       0.18667        1.60969        0.42578
       0.17333        1.61066        0.42487
       0.16000        1.61155        0.42402
       0.14667        1.61238        0.42324
       0.13333        1.61314        0.42253
       0.12000        1.61381        0.42189
       0.10667        1.61441        0.42134
       0.09333        1.61490        0.42088
       0.08000        1.61526        0.42053
       0.06667        1.61546        0.42034
       0.05333        1.61546        0.42034
       0.04000        1.61518        0.42061
       0.02667        1.61451        0.42124
       0.01333        1.61317        0.42250
       0.00000        1.60850        0.42690
&BPNODE  TNODE=2  TNPC=15  TINTC= 0      /
       0.00000        1.60850        0.42690
       0.01333        1.60383        0.43130
```

```
          0.02667        1.60249        0.43256
          0.04000        1.60182        0.43319
          0.05333        1.60154        0.43346
          0.06667        1.60154        0.43346
          0.08000        1.60174        0.43327
          0.09333        1.60210        0.43292
          0.10667        1.60259        0.43246
          0.12000        1.60319        0.43191
          0.13333        1.60386        0.43127
          0.14667        1.60462        0.43056
          0.16000        1.60545        0.42978
          0.17333        1.60634        0.42893
          0.18667        1.60731        0.42802
          0.20000        1.60835        0.42704
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                /
          0.20000        1.67865        0.46473
          0.18667        1.67969        0.46379
          0.17333        1.68065        0.46292
          0.16000        1.68154        0.46212
          0.14667        1.68237        0.46137
          0.13333        1.68312        0.46069
          0.12000        1.68380        0.46008
          0.10667        1.68439        0.45955
          0.09333        1.68487        0.45911
          0.08000        1.68523        0.45879
          0.06667        1.68544        0.45860
          0.05333        1.68544        0.45860
          0.04000        1.68516        0.45885
          0.02667        1.68449        0.45946
          0.01333        1.68315        0.46067
          0.00000        1.67850        0.46486
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
          0.00000        1.67850        0.46486
          0.01333        1.67385        0.46906
          0.02667        1.67251        0.47027
          0.04000        1.67184        0.47088
          0.05333        1.67156        0.47113
          0.06667        1.67156        0.47113
          0.08000        1.67177        0.47094
          0.09333        1.67213        0.47062
          0.10667        1.67261        0.47018
          0.12000        1.67320        0.46965
```

```
       0.13333        1.67388        0.46903
       0.14667        1.67463        0.46836
       0.16000        1.67546        0.46761
       0.17333        1.67635        0.46681
       0.18667        1.67731        0.46594
       0.20000        1.67835        0.46500
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
        ALF=0.0  THETA=0.0  INMODE=4
        TNODS=0  TNPS=0  TINTS=0                /
       0.20000        1.74865        0.50432
       0.18667        1.74968        0.50342
       0.17333        1.75064        0.50259
       0.16000        1.75153        0.50182
       0.14667        1.75235        0.50111
       0.13333        1.75310        0.50046
       0.12000        1.75377        0.49988
       0.10667        1.75436        0.49937
       0.09333        1.75484        0.49895
       0.08000        1.75520        0.49864
       0.06667        1.75540        0.49847
       0.05333        1.75540        0.49847
       0.04000        1.75513        0.49871
       0.02667        1.75446        0.49928
       0.01333        1.75313        0.50044
       0.00000        1.74850        0.50445
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        1.74850        0.50445
       0.01333        1.74387        0.50846
       0.02667        1.74254        0.50961
       0.04000        1.74187        0.51019
       0.05333        1.74160        0.51043
       0.06667        1.74160        0.51043
       0.08000        1.74180        0.51025
       0.09333        1.74216        0.50994
       0.10667        1.74264        0.50952
       0.12000        1.74323        0.50901
       0.13333        1.74390        0.50843
       0.14667        1.74465        0.50778
       0.16000        1.74547        0.50707
       0.17333        1.74636        0.50630
       0.18667        1.74732        0.50547
       0.20000        1.74835        0.50457
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /
&SECT1  STX=0.00  STY=0.00  STZ=0.00  SCALE=1.00
```

```
          ALF=0.0   THETA=0.0   INMODE=4
          TNODS=0   TNPS=0   TINTS=0                   /
         0.20000         1.81764         0.54493
         0.18667         1.81868         0.54407
         0.17333         1.81963         0.54327
         0.16000         1.82051         0.54254
         0.14667         1.82132         0.54186
         0.13333         1.82207         0.54124
         0.12000         1.82274         0.54068
         0.10667         1.82332         0.54019
         0.09333         1.82380         0.53979
         0.08000         1.82416         0.53949
         0.06667         1.82436         0.53933
         0.05333         1.82436         0.53933
         0.04000         1.82408         0.53956
         0.02667         1.82342         0.54011
         0.01333         1.82210         0.54121
         0.00000         1.81750         0.54505
&BPNODE   TNODE=2   TNPC=15   TINTC= 0       /
         0.00000         1.81750         0.54505
         0.01333         1.81290         0.54888
         0.02667         1.81158         0.54998
         0.04000         1.81092         0.55053
         0.05333         1.81064         0.55076
         0.06667         1.81064         0.55076
         0.08000         1.81084         0.55060
         0.09333         1.81120         0.55030
         0.10667         1.81168         0.54990
         0.12000         1.81226         0.54941
         0.13333         1.81293         0.54885
         0.14667         1.81368         0.54823
         0.16000         1.81449         0.54756
         0.17333         1.81537         0.54682
         0.18667         1.81632         0.54603
         0.20000         1.81736         0.54517
&BPNODE   TNODE=3   TNPC=15   TINTC= 0       /
&SECT1   STX=0.00   STY=0.00   STZ=0.00   SCALE=1.00
          ALF=0.0   THETA=0.0   INMODE=4
          TNODS=3   TNPS=25   TINTS=  3       /
         0.20000         1.88564         0.58648
         0.18667         1.88667         0.58566
         0.17333         1.88761         0.58490
         0.16000         1.88849         0.58419
         0.14667         1.88930         0.58354
         0.13333         1.89003         0.58295
```

```
       0.12000        1.89070        0.58242
       0.10667        1.89128        0.58195
       0.09333        1.89176        0.58157
       0.08000        1.89211        0.58128
       0.06667        1.89231        0.58112
       0.05333        1.89231        0.58112
       0.04000        1.89204        0.58134
       0.02667        1.89138        0.58187
       0.01333        1.89006        0.58292
       0.00000        1.88550        0.58659
&BPNODE  TNODE=2  TNPC=15  TINTC= 0     /
       0.00000        1.88550        0.58659
       0.01333        1.88094        0.59026
       0.02667        1.87962        0.59132
       0.04000        1.87896        0.59185
       0.05333        1.87869        0.59206
       0.06667        1.87869        0.59207
       0.08000        1.87889        0.59191
       0.09333        1.87924        0.59162
       0.10667        1.87972        0.59124
       0.12000        1.88030        0.59077
       0.13333        1.88097        0.59024
       0.14667        1.88170        0.58964
       0.16000        1.88251        0.58899
       0.17333        1.88339        0.58829
       0.18667        1.88433        0.58753
       0.20000        1.88536        0.58671
&BPNODE  TNODE=3  TNPC=15  TINTC= 0     /

&PATCH1  IREV=  0  IDPAT= 1  MAKE=   1  KCOMP= 1  KASS= 1
         IPATSYM=0    IPATCOP=0   /
          WING_TIP
&PATCH2  ITYP=  1    TNODS= 5    TNPS=   2    TINTS= 3   /


&WAKE1   IDWAK=1         IFLXW=0   /
    WING_WAKE
&WAKE2   KWPACH=1    KWSIDE=2    KWLINE=0     KWPAN1=0
         KWPAN2=0    NODEW=5     INITIAL=0    /


&ONSTRM  NONSL = 0  KPSL = 0    /

&VS1     NVOLR= 0        NVOLC= 0   /
&VS2     XO=      5.0000  YO=      3.0000  ZO=     -2.0000   /
```

```
&VS3     X1=    10.0000  Y1=      3.0000  Z1=     -2.0000  NPT1=  3/
&VS4     X2=     5.0000  Y2=      0.0000  Z2=     -2.0000  NPT2=  5/
&VS5     X3=     5.0000  Y3=      3.0000  Z3=      2.0000  NPT3= 10/
&VS6     XR0=    2.0000  YR0=     2.0000  ZR0=     0.0000 /
&VS7     XR1=    4.0000  YR1=     2.0000  ZR1=     0.0000
         XR2=    2.0000  YR2=     2.0000  ZR2=     1.0000 /
&VS8     R1=     0.1000  R2=      1.0000  PHI1=  0.0     PHI2=360.0/
&VS9     NRAD=   5       NPHI=   12       NLEN=  3  /


&SLIN1   NSTLIN=0  /
&SLIN2   SX0=   -2.0000  SY0=     1.0000  SZ0=    -0.5000
         SU=     0.0000  SD=     10.0000  DS=      0.2500  /
&SLIN2   SX0=   -2.0000  SY0=     1.0000  SZ0=    -0.4000
         SU=     0.0000  SD=     10.0000  DS=      0.2500  /
&SLIN2   SX0=   -2.0000  SY0=     1.0000  SZ0=    -0.3000
         SU=     0.0000  SD=     10.0000  DS=      0.2500  /
&SLIN2   SX0=   -2.0000  SY0=     1.0000  SZ0=    -0.2000
         SU=     0.0000  SD=     10.0000  DS=      0.2500  /
&SLIN2   SX0=   -2.0000  SY0=     1.0000  SZ0=    -0.1000
         SU=     0.0000  SD=     10.0000  DS=      0.2500  /
&SLIN2   SX0=   -2.0000  SY0=     1.0000  SZ0=     0.0000
         SU=     0.0000  SD=     10.0000  DS=      0.2500  /
```

C.2 SHAR2 PMARC Input File

```
SIMPLE HIGH ASPECT RATIO WING - NACA 0014
 &BINP2   LSTINP=2,    LSTOUT=1,    LSTFRQ=0,    LENRUN=0,
          LPLTYP=1,   /
 &BINP3   LSTGEO=1,    LSTNAB=0,    LSTWAK=2,    LSTCPV=0,    /
 &BINP4   MAXIT=250,    SOLRES=0.0005   /
 &BINP5   NTSTPS=20     DTSTEP=0.005         /
 &BINP6   RSYM=0.0     RGPR=0.0     RFF=5.0   RCORES=0.0080
          RCOREW=0.0080  /
 &BINP7   VINF=100.00    VSOUND=1116.0   /
 &BINP8   ALDEG=0.0    YAWDEG=0.0   PHIDOT=0.0   THEDOT=0.0
          PSIDOT=0.0    /
 &BINP8A  PHIMAX= 0.0  THEMAX=0.0  PSIMAX=0.0
          WRX=0.0      WRY=0.0      WRZ=0.0       /
 &BINP8B  DXMAX=0.0    DYMAX=0.0    DZMAX=0.0
          WTX=0.0      WTY=0.0      WTZ=-1.0   /
 &BINP9   CBAR=1.0  SREF=100.00  SSPAN= 50.00
          RMPX=0.00  RMPY=0.00    RMPZ=0.00       /
 &BINP10  NORSET=1     NBCHGE=1     NCZONE=0
          NCZPCH=0     CZDUB=0.0    VREF=0.0     /
 &BINP11  NORPCH=0   NORF=0     NORL=0
          NOCF=0     NOCL=0    VNORM=0.0       /
 &BINP12  KPAN=0    KSIDE=0   NEWNAB=0    NEWSID=0      /
 &BINP13  NBLIT=0 /

 &ASEM1   ASEMX=0.00       ASEMY=0.00       ASEMZ=0.00
          ASCAL=1.00       ATHET=0.00       NODEA=5      /
 &ASEM2   APXX=0.00        APYY=0.00        APZZ=0.00
          AHXX=0.00        AHYY=1.00        AHZZ=0.00      /
 &COMP1   COMPX= 0.0000     COMPY= 0.0000     COMPZ= 0.0000
          CSCAL=  1.000     CTHET=   0.0      NODEC= 5    /
 &COMP2   CPXX= 0.0000      CPYY= 0.0000      CPZZ= 0.0000
          CHXX= 0.0000      CHYY=  1.000      CHZZ= 0.0000  /

 &PATCH1  IREV=  0  IDPAT= 1  MAKE=   0  KCOMP= 1  KASS= 1
          IPATSYM=0   IPATCOP=0   /
           WING
 &SECT1   STX= 0.0000  STY= 0.0     STZ= 0.0000   SCALE= 1.000
          ALF=   0.0   THETA=   0.0
          INMODE=  5   TNODS= 0     TNPS=  0     TINTS= 0     /
 &SECT2   RTC=    0.1400  RMC= 0.0000     RPC= 0.0000
          IPLANE=  2   TNPC= 15     TINTC= 0          /
```

```
&SECT1   STX= 0.0000  STY= 50.00   STZ= 0.0000  SCALE= 1.000
         ALF=    0.0  THETA=   0.0
         INMODE=  0   TNODS= 3     TNPS= 20     TINTS= 2      /

&PATCH1  IREV=  0  IDPAT= 1  MAKE=   1  KCOMP= 1  KASS= 1
         IPATSYM=0    IPATCOP=0       /
          WING_TIP
&PATCH2  ITYP=  1     TNODS= 5     TNPS=   3    TINTS= 3          /

&WAKE1   IDWAK=1         IFLXW=0     /
 WING_WAKE
&WAKE2   KWPACH=1        KWSIDE=2        KWLINE=0        KWPAN1=0
         KWPAN2=0        NODEW=5         INITIAL=0     /

&ONSTRM  NONSL = 0  KPSL = 0  /

&VS1     NVOLR= 0        NVOLC= 0      /
&VS2     X0=     5.0000  Y0=     3.0000  Z0=    -2.0000   /
&VS3     X1=    10.0000  Y1=     3.0000  Z1=    -2.0000  NPT1=  3  /
&VS4     X2=     5.0000  Y2=     0.0000  Z2=    -2.0000  NPT2=  5  /
&VS5     X3=     5.0000  Y3=     3.0000  Z3=     2.0000  NPT3= 10  /
&VS6     XR0=    2.0000  YR0=    2.0000  ZR0=    0.0000   /
&VS7     XR1=    4.0000  YR1=    2.0000  ZR1=    0.0000
         XR2=    2.0000  YR2=    2.0000  ZR2=    1.0000   /
&VS8     R1=     0.1000  R2=     1.0000  PHI1=  0.0   PHI2=360.0  /
&VS9     NRAD=   5       NPHI=  12       NLEN=  3    /

&SLIN1   NSTLIN=0      /
&SLIN2   SX0=   -2.0000  SY0=    1.0000  SZ0=   -0.5000
         SU=     0.0000  SD=    10.0000  DS=     0.2500  /
&SLIN2   SX0=   -2.0000  SY0=    1.0000  SZ0=   -0.4000
         SU=     0.0000  SD=    10.0000  DS=     0.2500  /
&SLIN2   SX0=   -2.0000  SY0=    1.0000  SZ0=   -0.3000
         SU=     0.0000  SD=    10.0000  DS=     0.2500  /
&SLIN2   SX0=   -2.0000  SY0=    1.0000  SZ0=   -0.2000
         SU=     0.0000  SD=    10.0000  DS=     0.2500  /
&SLIN2   SX0=   -2.0000  SY0=    1.0000  SZ0=   -0.1000
         SU=     0.0000  SD=    10.0000  DS=     0.2500  /
&SLIN2   SX0=   -2.0000  SY0=    1.0000  SZ0=    0.0000
         SU=     0.0000  SD=    10.0000  DS=     0.2500  /
```

C.3 BACT PMARC Input File

```
BACT TEST WING NACA 0012
 &BINP2   LSTINP=2,    LSTOUT=1,    LSTFRQ=0,    LENRUN=3,
          LPLTYP=1,     /
 &BINP3   LSTGEO=1,    LSTNAB=0,    LSTWAK=2,    LSTCPV=0,    /
 &BINP4   MAXIT=500,    SOLRES=0.0005    /
 &BINP5   NTSTPS=10     DTSTEP=0.1       /
 &BINP6   RSYM=1.0     RGPR=0.0     RFF=5.0   RCORES=0.0080
          RCOREW=0.0080  /
 &BINP7   VINF=100.00    VSOUND=1116.0    /
 &BINP8   ALDEG=0.0    YAWDEG=0.0   PHIDOT=0.0   THEDOT=0.0
          PSIDOT=0.0    /
 &BINP8A  PHIMAX= 0.0  THEMAX= 0.0  PSIMAX=0.0
          WRX=0.0      WRY=0.0      WRZ=0.0    /
 &BINP8B  DXMAX=0.0    DYMAX=0.0    DZMAX=0.0
          WTX=0.0      WTY=0.0      WTZ=0.0    /
 &BINP9   CBAR=1.3333  SREF=3.5555  SSPAN= 2.6667
          RMPX=0.00  RMPY=0.00    RMPZ=0.00      /
 &BINP10  NORSET=1     NBCHGE=1     NCZONE=0
          NCZPCH=0     CZDUB=0.0    VREF=0.0    /
 &BINP11  NORPCH=0   NORF=0     NORL=0
          NOCF=0     NOCL=0    VNORM=0.0                     /
 &BINP12  KPAN=0    KSIDE=0   NEWNAB=0    NEWSID=0      /
 &BINP13  NBLIT=0 /

 &ASEM1   ASEMX=0.00       ASEMY=0.00       ASEMZ=0.00
          ASCAL=1.00       ATHET=0.00       NODEA=5     /
 &ASEM2   APXX=0.00        APYY=0.00        APZZ=0.00
          AHXX=0.00        AHYY=1.00        AHZZ=0.00  /


 &COMP1   COMPX= 0.0000     COMPY= 0.0000     COMPZ= 0.0000
          CSCAL=  1.000     CTHET=   0.0      NODEC= 5   /
 &COMP2   CPXX= 0.0000      CPYY= 0.0000      CPZZ= 0.0000
          CHXX= 0.0000      CHYY=  1.000      CHZZ= 0.0000  /



 &PATCH1  IREV=  0  IDPAT= 1  MAKE=   0  KCOMP= 1  KASS= 1
          IPATSYM=0    IPATCOP=0  /
           WING
 &SECT1   STX= 0.0000  STY= 0.0   STZ= 0.0000  SCALE=  1.3333
```

```
         ALF=   0.0   THETA=   0.0
         INMODE=  5   TNODS= 0     TNPS=  0     TINTS= 0   /
&SECT2   RTC=    0.1200  RMC= 0.0000     RPC= 0.0000
         IPLANE=  2   TNPC= 15     TINTC= 0          /
&SECT1   STX= 0.0000  STY= 2.6667   STZ= 0.0000   SCALE= 1.3333
         ALF=   0.0   THETA=   0.0
         INMODE=  0   TNODS= 3     TNPS= 20     TINTS= 2   /

&PATCH1  IREV=  0  IDPAT= 1  MAKE=   1  KCOMP= 1  KASS= 1
         IPATSYM=0    IPATCOP=0        /
          WING_TIP
&PATCH2  ITYP=  2     TNODS= 5     TNPS=   4    TINTS= 3   /

&WAKE1   IDWAK=1         IFLXW=0      /
 WING_WAKE
&WAKE2   KWPACH=1   KWSIDE=2    KWLINE=0    KWPAN1=0
         KWPAN2=0   NODEW=5     INITIAL=1    /
&SECT1   STX= 0.0000  STY=    0.0000  STZ= 0.0000  SCALE= 1.0000
         ALF=   0.0   THETA=   0.0
         INMODE=-1    TNODS= 3     TNPS=  0     TINTS= 1    /

&ONSTRM  NONSL = 0  KPSL = 0  /

&VS1     NVOLR= 0        NVOLC= 0       /
&VS2     X0=      5.0000  Y0=      3.0000  Z0=     -2.0000   /
&VS3     X1=     10.0000  Y1=      3.0000  Z1=     -2.0000  NPT1=  3/
&VS4     X2=      5.0000  Y2=      0.0000  Z2=     -2.0000  NPT2=  5/
&VS5     X3=      5.0000  Y3=      3.0000  Z3=      2.0000  NPT3= 10/
&VS6     XR0=     2.0000  YR0=     2.0000  ZR0=     0.0000   /
&VS7     XR1=     4.0000  YR1=     2.0000  ZR1=     0.0000
         XR2=     2.0000  YR2=     2.0000  ZR2=     1.0000   /
&VS8     R1=      0.1000  R2=      1.0000  PHI1=  0.0    PHI2=360.0 /
&VS9     NRAD=    5       NPHI=  12        NLEN=  3     /

&SLIN1   NSTLIN=0      /
&SLIN2   SXO=    -2.0000  SYO=     1.0000  SZO=    -0.5000
         SU=      0.0000  SD=     10.0000  DS=      0.2500  /
&SLIN2   SXO=    -2.0000  SYO=     1.0000  SZO=    -0.4000
         SU=      0.0000  SD=     10.0000  DS=      0.2500  /
&SLIN2   SXO=    -2.0000  SYO=     1.0000  SZO=    -0.3000
         SU=      0.0000  SD=     10.0000  DS=      0.2500  /
&SLIN2   SXO=    -2.0000  SYO=     1.0000  SZO=    -0.2000
         SU=      0.0000  SD=     10.0000  DS=      0.2500  /
&SLIN2   SXO=    -2.0000  SYO=     1.0000  SZO=    -0.1000
         SU=      0.0000  SD=     10.0000  DS=      0.2500  /
```

```
&SLIN2    SXO=    -2.0000  SYO=     1.0000  SZO=     0.0000
          SU=      0.0000  SD=     10.0000  DS=      0.2500  /
```

C.4 Flexible-Body Impulse Distribution File (IMPULSE)

```
Linear Twist, 20 span stations
Y VZ THETA Q
1.25 0 0.00025 0
3.75 0 0.00075 0
6.25 0 0.00125 0
8.75 0 0.00175 0
11.25 0 0.00225 0
13.75 0 0.00275 0
16.25 0 0.00325 0
18.75 0 0.00375 0
21.25 0 0.00425 0
23.75 0 0.00475 0
26.25 0 0.00525 0
28.75 0 0.00575 0
31.25 0 0.00625 0
33.75 0 0.00675 0
36.25 0 0.00725 0
38.75 0 0.00775 0
41.25 0 0.00825 0
43.75 0 0.00875 0
46.25 0 0.00925 0
48.75 0 0.00975 0
```