ABSTRACT

Title of dissertation: A Time Parallel Approach to Numerical Simulation of Asymptotically Stable Periodic Dynamical Systems with Application to CFD Models of Helicopter Rotors

Benjamin S. Silbaugh, Doctor of Philosophy, 2023

Dissertation directed by: Professor James D. Baeder
Department of Aerospace Engineering

Modern High Performance Computing (HPC) machines are distributed memory clusters, consisting of multi-core compute nodes. Engineering simulation and analysis tools must employ efficient parallel algorithms in order to fully utilize the compute capability of modern HPC machines. The trend in Computational Fluid Dynamcis (CFD) has been to construct parallel solution algorithms based on some form of spatial domain decomposition. This approach has been shown to be a success for many practical applications. However, as one attempts to utlize more compute cores, limitations in strong scalability are inevitably reached due to a diminishing compute workload per compute core and either fixed or increasing communication cost. Furthermore, spatial domain decomposition approaches cannot be easily applied to mid-fidelity structural dynamics or rigid body dynamics models. A significant majority of industrial fluid and structural dynamic models utilize some form of time marching. Thus, if the domain decomposition strategy may be extended to include the temporal dimension, additional opportunity for increased parallelism

may be realized.

A new form of periodic multiple shooting is proposed that is matrix-free and may be applied to high-fidelity multiphysics models or other high dimensional systems. The proposed methodology is formulated entirely in the time domain. Therefore, existing time-domain simulation tools may utilize the proposed approach to achieve a high degree of distributed memory parallelism without requiring any reformulation. Furthermore, the proposed methodology may be combined with conventional space domain decomposition techniques and other forms of data parallelism to achieve maximal performance on modern HPC architectures. The proposed algorithm retains the iterative shoot-correct approach of conventational periodic shooting methods. However, the correction stage is formulated using a hierarchical evaluation strategy combined with an Arnoldi subspace approximation to eliminate the need for explicit formulation of Jacobian matricies. The local convergence of the proposed method is formally proven for the case of an asyptotically stable dynamical system. The proposed method is numerically tested for a 2D limit cycle problem, a rigid blade helicoper rotor model with quasi-steady aerodynamics and autopilot trim, and an OVERSET CFD model of a helicopter rotor with prescribed elastic blade motions. The method is observed to be convergent in all test cases and found to exhibit good scalability.

The proposed periodic multiple shooting method is a practical means of reducing time-to-solution for numerical simulations of asymptotically stable periodic systems on distributed memory parallel computers. Furthermore, the proposed method may be used to enhance the parallel scalability of OVERSET CFD models of heli-

copter rotors in steady periodic flight.

A Time Parallel Approach to Numerical Simulation of Asymptotically Stable Periodic Dynamical Systems with Application to CFD Models of Helicopter Rotors

by

Benjamin Silbaugh

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:
Professor James D. Baeder, Chair/Advisor
Professor Inderjit Chopra
Professor Anubhav Data
Professor Roberto Celi
Professor Amr Baz

© Copyright by
Benjamin S. Silbaugh
2023

Table of Contents

Chapter 1: Introduction

1.1 Time Parallelism

[TODO]

1.2 Research Objectives

The objective of this work is to develop a new method for time-parallel numerical simulation of asymptotically stable dynamical systems. The new method ought to be applicable to high dimensional systems, including multi-physics simulations. It is desirable to formulate the methodology in the time-domain, as many state-of-the-art multi-physics simulation tools utilize time marching techniques. Finally, it ought to be possible for the methodology to be combined with established space domain decomposition techniques and other forms of data parallelism to maximize performance on modern HPC architectures.

1.3 Organization of Dissertation

[TODO]

Chapter 2: A Periodic Multiple Shooting Method for Asymptotically Stable Periodic Systems

{ "periodicity error" needs to be replaced with "continuity error". The intermediate solutions are already periodic by construction }

2.1 Preliminaries

The theoretical framework of the method of Periodic Shooting for asymptotically stable periodic systems is built upon the theory of Dynamical Systems. In the interest in being self contained, the essential concepts, definitions, and theorems from dynamical systems are reviewed here. For a more systematic development of these ideas the reader may wish to read Hirsch [36] and Wiggins [37]. Given that the ultimate goal is to develop a solution method for numerical models, it will suffice to consider dynamical systems on open subsets of Euclidean spaces ($\mathbb{R}^n$ with the Euclidean metric). For an introduction to the more general case of dynamical systems on manifolds, see Arnold [38].

Informally, a dynamical system may be thought of as a system that changes or evolves with time. A dynamical system embodies the concept of identity in the sense that it may change from one moment to the next and yet be considered the same

object. The state of the system is a quantity (a vector of numbers) that encodes all of the necessary information required to determine the future and past history of the system, together with a law that governs the advancement from one state to the next. The phase space of a dynamical system is the set of all possible states that the dynamical system can rationally posses. Mathematically, a dynamical system is defined as follows [37].

Definition 1. *Let $\mathcal{S}$ be an open subset of a Euclidean space. A "dynamical system" is a C^1 map $\phi : \mathbb{R} \times \mathcal{S} \to \mathcal{S}$ such that the maps of the form $\phi_t : \mathcal{S} \to \mathcal{S}$, each defined by the equation*

$$\phi_t(x) = \phi(t, x), \quad \forall x \in \mathcal{S},$$

satisfy the following:

1. *$\phi_0 : \mathcal{S} \to \mathcal{S}$, is the identity map,*

2. *The composition $\phi_t \circ \phi_s = \phi_{t+s}$ for all $t, s \in \mathbb{R}$.*

The set $\mathcal{S}$ is called the "phase space."

The map ϕ may be viewed as the law that governs the evolution of a particular dynamical system from one state to another over a time of t.

The following proposition from Hirsch [36] will prove useful.

Proposition 1. *The map $\phi_t : \mathcal{S} \to \mathcal{S}$ is C^1 for each t and has a C^1 inverse.*

The concept of a vector field provides a useful way to geometrically visualize and qualitatively reason about the evolution of a dynamical system. It also provides an important link to the classical notion of a differential equation.

Definition 2. *A "vector field" on $\mathcal{S}$ is a map $f : \mathcal{S} \to E$, where E is some vector space and $\mathcal{S}$ is an open subset of E.*

As shown by Hirsch [36], every dynamical system corresponds to a differential equation of the form

$$\dot{x} = f(x) \tag{2.1}$$

where $f \in C^k(U)$, $k \geq 1$, for some open subset U of $\mathbb{R}^n$. Furthermore, every differential equation of the form (2.1) corresponds to a dynamical system. Differential equations of the form (2.1) define a vector field on the phase space. The map ϕ_t of a dynamical system is a solution of (2.1). Wiggins [37] considers dynamical systems and differential equations of form (2.1) as one in the same. Hirsch [36], on the other hand, considers dynamical systems and differential equations to be distinct mathematical objects related by way of the map ϕ_t. If one only considers dynamical systems on open subsets of Euclidean spaces, then the two viewpoints are interchangeable.

Three geometric concepts are essential to any discussion on the behavior of a dynamical system ϕ: trajectory, integral curve, and orbit.

Definition 3. *The "trajectory of ϕ through point x_0 at time t_0" is the function defined by $(t, t_0, x_0) \mapsto \phi(t - t_0, x_0)$.*

Definition 4. *The "integral curve of ϕ passing through point x_0 at time t_0", is the set $\{(x, t) \in \mathcal{S} \times \mathbb{R} : x = \phi(t - t_0, x_0), t \in I\}$, where I is the largest interval, containing t_0, where $\phi(t - t_0, x_0)$ is defined.*

Definition 5. *The "orbit of ϕ through point x_0" is the set $\{x \in \mathcal{S} : x = \phi(t, x_0), t \in I\}$, where I is the largest interval, containing zero, where $\phi(t, x_0)$ is defined.*

Observe that the trajectory of a dynamical system is effectively the solution of the corresponding differential equation with initial condition x_0 at time t_0. The integral curve may be thought of as the graph of system states x as a function of time t. An orbit of ϕ is the set of points in the phase space that belong to the same history or evolution of the system represented by ϕ. Sometimes the orbit of ϕ through point x_0 is written $O(x_0)$. By definition $x_0 \in O(x_0)$.

The notion of stability needs to be defined. The most general notion of stability is defined first.

Definition 6. *A trajectory $\bar{x}$ is said to be "Liapunov stable" if, for any $\epsilon > 0$, there exists a $\delta > 0$ such that, for any other trajectory y satisfying $\|\bar{x}(t_0) - y(t_0)\| < \delta$, then $\|x(t) - y(t)\| < \epsilon$ for all $t > t_0$, $t_0 \in \mathbb{R}$.*

Note that δ appearing in the above definition may depend on ϵ. Observe that Liapunov stability does not require neighboring trajectories to converge towards a common trajectory. Rather, it suffices that a neighboring trajectory remains within a neighborhood of radius ϵ. As an example, consider the equilibrium position of a spring-mass system. The phase portrait of such a system is a continuous family of concentric ellipses centered around the equilibrium point $\bar{x}$. For any $\epsilon > 0$, one can identify a displacement δ from equilibrium that will result in a motion whose trajectory in the phase space does not exceed a distance of ϵ from $\bar{x}$.

Definition 7. *A trajectory $\bar{x}$ is said to be "asymptotically stable" if it is Liapunov*

stable, and for any other trajectory y, there exists a constant $b > 0$ such that, if $\|\bar{x}(t_0) - y(t_0)\| < b$, then $\lim_{t\to\infty} \|\bar{x}(t) - y(t)\| = 0$.

Asymptotic stability requires that neighboring trajectories converge towards a common trajectory. An example of an asymptotically stable system is a spring-mass-damper system.

Definition 8. *A trajectory $\bar{x}$ through point x_0 is said to be "periodic of period T" if there exists a $T > 0$ such that $\bar{x}(t, t_0) = \bar{x}(t + T, t_0)$ for all $t \in \mathbb{R}$.*

Definition 9. *An orbit of a point $x_0 \in \mathcal{S}$ is said to be "periodic of period $k > 0$" if there exists an integer $k > 0$ and a map $g : \mathcal{S} \to \mathcal{S}$ such that $g^k(x_0) = g(x_0)$.*

Note that, given a periodic trajectory $\bar{x}$ of period T, the corresponding orbit in $\mathcal{S}$ traced by $\bar{x}$ is a periodic orbit. This follows by simply taking g in the foregoing definition to be ϕ_T and $k = 1$.

Definition 10. *Suppose ϕ is a dynamical system on $\mathcal{S}$, $f \in C^r(\mathcal{S})$ is the vector field, and $O(x_0)$ is a periodic orbit. Let $\mathcal{P}$ be a $n-1$ dimensional surface, where n is the dimension of $\mathcal{S}$. If $u \in \mathbb{R}^n$ is a vector normal to $\mathcal{P}$ at x_0, and $f(x_0) \cdot u \neq 0$, then $\mathcal{P}$ is called a "cross section" to the vector field f.*

The terms "Poincare section", "Poincare surface", or "Poincare plane" are alternative names for cross section.

By continuity of ϕ, one can be assured of the existence of an open set $U \subset \mathcal{P}$ containing x_0, and an open interval $\mathcal{T} \subset \mathbb{R}$ containing T, such that $\phi_t(x) \in \mathcal{P}$ for any $x \in U$ and $t \in \mathcal{T}$. This justifies the following definition for a Poincare map.

Definition 11. *A "Poincare map" is a map $P : U \to \mathcal{P}$, that satisfies $P(x) = \phi_t(x)$ for any $x \in U$ and $t \in \mathcal{T}$.*

In other words, a Poincare map is a map that takes a point near a periodic orbit and on a cross section of the flow, to the point in the same cross section that corresponds to the first return along the trajectory passing through the original point. In the special case where the original point coincides with the periodic orbit, the Poincare map returns the same point, $P(x_0) = x_0$; i.e. x_0 is a fixed point of P.

Proposition 2. *Suppose ϕ is a dynamical system and $x_0 \in \mathcal{S}$ is a point on a periodic orbit of ϕ. Let $\mathcal{P} \subset \mathcal{S}$ be a cross section of $O(x_0)$ containing x_0, and $U \subset \mathcal{P}$ an open neighborhood of x_0 such that the Poincare map $P : U \to \mathcal{P}$ is defined. The orbit $O(x_0)$ is asymptotically stable if and only if the Eigenvalues of $DP(x_0)$ all have negative real components.*

Proof. See Wiggens [37]. □

2.2 Theory

{Show method convergence by: 1. recall asymptotic stability def and corresponding eigenvalues of poincare map, 2. show that continuity of f and subsequently Df implies the existence of a region containing the attractor (fixed point in transverse section), where the Eigenvalues are all of same sign as that on the attractor, 3. use properties of spectral radius to prove convergence.}

Theorem 1. *Suppose ϕ is a dynamical system on $\mathcal{S}$, and $x_0 \in \mathcal{S}$ is such that $O(x_0)$*

is an asymptotically stable periodic orbit. Let $\mathcal{P}_0, \cdots, \mathcal{P}_{p-1}$ *be a sequence of* p *cross sections of* $O(x_0)$, $x_i \in O(x_0) \cap \mathcal{P}_i$, $i = 0, \cdots, p-1$, *and* $P_i : U_i \to \mathcal{P}_{(i+1) \bmod p}$.

Theorem 2. *Suppose the state space* $\mathcal{S} \subset \mathbb{R}^n$ *of (2.1) is divided into* $p \in \mathbb{N}$ *partitions at Poincare section. Furthermore, suppose each of the* p *partitions is labeled with an integer, beginning at* 0, *incremented by* 1 *in the direction of the flow on* A, *up to* $p-1$.

Let $l : [-1, 0, \cdots, p] \to [0, \cdots, p-1]$ *be defined by the equation*

$$l(i) = \begin{cases} 0, & i = p, \\ p-1, & i = -1, \\ i, & \text{otherwise} \end{cases} \tag{2.2}$$

Furthermore, let $\sigma_i^j : M \to M$ *be the map that takes a point* x *from the* i^{th} *Poincare section to the* j^{th} *Poincare section along the orbit passing through* x, *and let* S_i^j *be the linearization of* σ_i^j.

For each $i \in \{1, \cdots, p-1\}$, *and any choice of* $x_{i,0}$ *in the* i^{th} *Poincare section, the sequence* $\{x_{i,j}\}_{j=0}^{\infty}$ *defined by*

$$x_{i,j+1} = x_{i,j} + \sum_{k=0}^{p-1} S^i_{l(i-k)} \left(\sigma^{l(i-k)}_{l(i-k-1)} \left(x_{l(i-k-1),j} \right) - x_{l(i-k),j} \right) \tag{2.3}$$

is contained in the i^{th} *Poincare section and converges to a unique point* $x_m^\star \in A$.

Proof. {TODO} □

Proposition 3. *The sequence of approximations generated by Theorem 1 are* C^0 *periodic functions.* {*REFINE ME*}

Proof. {TODO} □

Theorem 1 lays the foundation for a numerical method for approximating periodic solutions to (2.1). Choose the number of partitions p that will be used to divide the state space $M \subset \mathbb{R}^n$ of (2.1) at Poincare section. Label each of the p partitions with an integer, begining at 0, incrementing by 1 in the direction of the flow on A, up to $p-1$. For each partition i, $i = 1, \cdots, p-1$, estimate where the flow on A enters the partition i, and label this point $x_{i,0}$. Then, starting with $j = 0$, proceed as follows:

1. For each $x_{i,j}$, integrate equation (2.1), using $x_{i,j}$ as an initial condition, until the Poincare boundary containing $x_{l(i+1),j}$ is reached. Denote $\tilde{x}_{l(i+1),j}$ as the point that results by integrating (2.1) with $x_{i,j}$ as an initial condition.

2. For each i Poincare boundary, $i = 0, \cdots, p-1$, compute the periodicity error $e_{i,j}$, where "periodicity error" is gven by the equation

$$e_{i,j} = \tilde{x}_{i,j} - x_{i,j}.$$

3. Compute

$$x_{i,j} = x_{i,j} + \sum_{k=0}^{p-1} S^i_{j(i-k)} e_{m,j(i-k)}. \tag{2.4}$$

{Do I need to label S as changing with each iteration? I think yes.}

TODO:

- Basic properties of asymptotically stable periodic systems: eigensystem, poincare planes, basin of attraction, reparameterization (choose any translation of t), etc.

- State transition operators and matrices.
- In the case that there are more attractors, it is enough that the initial guesses are in the basin of attraction of a stable periodic attractor?
- How does one determine if such a system has a periodic attractor, and that its stable? How does one determine the flow on X? Assumption. Experience/physical argument. Analysis.
- Concept of "periodicity error" as error between x_0 and $x(T)$ values.
- Basic convergence analysis.
- Parallel performance: fixed k, $k = N_p - 1$.
- Use of 1 time partition is equivolent to direct time marching.
- Show that intermediate solutions are periodic (if end points are dropped), but not necessarily continuous. The iteration process serves to remove the discontinuity, to map the approximation from C^0 to C^1. Remark on the implications this may have for rotor modeling, if any.

2.3 Algorithms and Data Structures

The numerical method proposed in Section 2.2 consists of formulation steps and computatational steps. The formulation steps are expected to be performed by a human with a view towards solving a specific class of problems (or equations). The computational steps are expected to be performed by a computer. In this

section, serial and parallel algorithms for performing the computational steps of the proposed numerical method are presented.

For all of the algorithms presented, the only data structures required are arrays of vectors (column vectors) and arrays of matrices. In a language such as Fortran, it suffices to model these data structures using the intrinsic multidimensional arrays. In C++, one may want to consider combining an STL container that supports efficient random access with elements defined as vector or matrix classes provided by a dedicated C++ linear algebra package.

The algorithms appearing in the algorithm listings that follow are expressed in the form of scripts, with procedures and functions representing key logical units (or steps) of the algorithm defined first, followed by a series of statements that consitute the main driver program of the algorithm. Unless stateted otherwise, all arguments passed to a procedure are passed by reference. This means that any modification of these variables by the procedure will be reflected in the corresponding variables of the calling procedure or program. All arguments passed to functions, on the other hand, are passed by value (or equivolently, passed as an unmutable reference). Each function returns its results by explictly invoking the **return** statement. Some algorithm listings may depend on externally defined data, procedures, or functions. All such dependencies are listed at the top of each algorithm listing, and denoted by the keyword **Require**. These external dependencies are introduced either to simplify the exposition or to abstract out algorithm elements whose implementation is problem dependent. The intent of the algorithm listings is to communicate the essential elements of each algorithm as a concept, and how those elements come

together to form a whole. In practice, one may need pass additional data to the procedures and functions shown in each algorithm listing, add statements to write or return intermediate solution data, add mechanisms for exception handling and error reporting, or repackage the algorithm in a way that adheres to modern Procedural, Functional, or Object Oriented design practices.

{Algorithm listings do not show how to compute S. There are many strategies for doing this, all of which depend on the size of the problem (are we using dense matrix operations, sparse matrix operations, or Krylov approximations) and the approximations being used (exact linearization or some other approximation).}

2.3.1 A Serial Algorithm

A serial algorithm implementing the computational steps of the numerical method proposed in Section 2.2 is given in Listing 1.

The listed algorithm requires two parameters, one procedure, and two functions to be externally defined. The parameters may be thought of as global parameters whose values are set based on experience or some other heuristic. The implementation of these external procedures and functions are problem dependent and therefore will implicitly defined by their input-output behavior. The first parameter ϵ is the convergence tolerance. The algorithm terminates when the periodicity error falls below this threshold. The second parameter is $kmax$. This is the maximum number of partitions over which periodicity error is propagated.

The first of these externally defined procedures is INITIALIZE. This procedure

takes X as an argument, and initializes its values to the estimate location of the intersection between the periodic attractor A and the p Poincare sections.

The externally defined function TimeStep(,) advances each state vector from one time step to the next. The implementation of this function may be realized as the application of a conventional time marching algorithm to the governing equations of the system to be analyzed. The listing shows TimeStep(a)s a function of a single argument x; however, if a multi-step time marching algorithm is desired, then additional state vectors would be passed as additional arguments.

The externally defined function AtPoincareSection(,) is a predicate function that returns true when a state vector is contained in one of the p Poincare sections, and false otherwise.

The first procedure in the listing is Shoot. This procedure computes the periodicity error, and takes two arguments X and E. X is an array containing the current estimate for the intersection of the periodic attractor at each Poincare section. X is unmodified by the procedure. E is an array that accumulates the periodicity error computed by the Shoot procedure. The Shoot procedure computes the periodicity error by using a time marching algorithm to follow the orbit that passes through each point described in X from the one Poincare section to the next. Then the periodicity error is computed as the difference between the integrated value and the current approximation.{This should be worded better.} (Since Shoot is free of any side-effects, one could also define Shoot as a function that takes X as input and returns E.)

The second procedure listed is Correct. This procedure uses the periodicity

error E computed by SHOOT to compute a correction to X that brings X closer to the periodic attractor. The procedure begins by initializing a temporary array Δx to 0, which will subsequently be used to accumulate the corrections to be applied to X. This is followed by three nested loops. The first loop increments the "order" of the approximation used to correct X. The second loop increments the partition index from which the periodic error is propagated, and the third loop effectively propagates this periodic error through each partition by application of the S operator associated with each partition. The results of the second and third loop are accumulated in Δx. Finally, X is corrected by performing array addition (elemenwise sum) of X and Δx. Note that algorithm as shown presumes that each $S_j^{l(j+1)}$ operator is known analytically. For "real-world" problems, this may need to be based on a finite difference approximation that is computed and during the time marching process in SHOOT, or some other approximation. {Clarify wording: the idea of error propgation has not really be formally defined, although may be it should be.}

The main driver program of the algorithm begins at line 23. First X is initialized, then the main iteration loop is started. First SHOOT is called, which returns the periodicity error E. Then the periodicity error is checked against the convergence tolerance ϵ. If the error is sufficiently small, the program exits. Otherwise, X is corrected by calling CORRECT, and then another iteration cycle begins.

2.3.2 Parallel Algorithms

A parallel algorithm implementing the computational steps of the numerical method proposed in Section 2.2 is given in Listing 2. This algorithm exploits the natural parallelism that exists in the previously discussed algorithm (Listing 1). The parallel algorithm may be characterised as a data-parallel algorithm based on domain decomposition of the state space, with Poincare sections defining the domain boundaries. The algorithm in Listing 2 implicitly associates each partition of the state space to an computational process. In principle, one could associate multiple partitions to each process. However, there does not appear to be any utility in doing this. It is also assumed that this algorithm is executed on a machine that supports mechanisms for inter-process communication and synchronization. The parallel programming model assumed to be similar to that of the Message Passing Interface {cite me}.

The algorithm in Listing 2 depends on the same two external parameters as previously discussed for the serial algorithm (Listing 1). Likewise, it depends on external procedures Initialize, and functions TimeStep, AtPoincareSection. These remain essentially unchanced, with the only difference being that Initialize need only take X as a single vector, rather than an array of vectors. The procedure Initialize sets X to the estimated location of the periodic attractor on the "left" or "upwind" Poincare boundary of the partition associated with the process executing the algorithm.

A new external procedure is introduced, ShiftRight. This procedure takes

a single argument, a vector, and utilizes the inter-process communication and synchronization mechanisms to send the value of its argument to the process on the right, overwriting it with the value received from the left. This procedure may be viewed as an analog of the MPI procedure MPI_SENDRECV combined with a ring process topology. {this could be clearer}

The SHOOT procedure computes the periodic error associated X. As with the serial algorithm (Listing 1), it does this by integrating the governing equations, with X as an initial condition, until the Poincare section forming the "downwind" boundary of the domain is reached. Then, the result of the integration process, initially stored in x, is sent to the process on the right and overwritten by the value received by the process on the left by calling SHIFTRIGHT. The periodic error is computed by evaluating $E = x - X$ as vectors. Note that the procedure arguments X and E are now vectors, whereas these arguments were arrays of vectors for the serial algorithm.

Procedure CORRECT uses the periodicity error E to correct X such that it is closer to the periodic attractor. Unlike the serial version of this procedure, no array of vectors or array of matrices are needed. All variables are either scalars, vectors (or 1D-arrays), or matrices (or 2D-arrays). The procedure begins by initializing a work vector Δx to zero. This vector is used to accumuate the correction to be applied to X. The integer k is a counter representing the current order of approximation used to compute the correction, and is initialized to zero. Next, the vector r, is a temporary variable that accumulates the action of S on each processor, as it is translated across each domain with each incremental increase in the order of

approximation. The matrix S, is the linearized state advancement operator that takes each point on the left Poincare boundary to the Poincare boundary on the right {This could be explained better.}

The main program, begining on line 31, is identical to the main program of the serial algorithm.

{Discuss mapping of processes to computational hardware units.} {Characterise communication pattern. Ring topology.} {Clarify MPI-like programming model?}

{logp parallel algorithm} {Provide parallel performance analysis.} {Is the parallel scalability of this method dependent on the number of states?}

2.4 The Connection to Multiple Shooting with Incomplete Cyclic Reduction

The method proposed in Section 2.2 may be viewed as a close cousin of a multiple shooting method combined with incomplete cyclic reduction. Futhermore, the two approaches yeild equivolent algorithms under certain circumstances.

Multiple shooting is a domain decomposition approach to boundary value problems. It may be used as a foundation for realizing parallel marching algorithms, as well as a way to improve upon the numerical stability limits of classical shooting methods {cite me}. In the context of periodic problems, multiple shooting may be viewed as a generalization of periodic shooting. Given a periodic system of known period T, one chooses a number of partitions p, and partitions the time

interval $[0, T]$ into p partitions,

$$I_i = [t_i, t_{i+1}], \quad i = 0, \cdots, p-1 \tag{2.5}$$

where $t_0 = 0$ and $t_p = T$. Let $x \in C^1(M)$ denote a periodic solution to (2.1). For each i partition, define $a_i, b_i \in M$ by the equations

$$a_i = x(t_i) \tag{2.6}$$

$$b_i = x(t_{i+1}) \tag{2.7}$$

By definition x is a continuous periodic function. Therefore

$$b_i = a_j, \quad j = i + 1 \pmod{p}. \tag{2.8}$$

Conversely, if $x_i \in C^1 M : I_i \to M$ satisfy $x_i(t_i) = a_i$ and $x_i(t_{i+1}) = b_i$, then $x(t) = x_i(t)$. That is, each x_i is equal to the restriction of x to I_i. The multiple shooting method attempts to construct x by trying to guess the value of each a_i such that solving (2.1) with a_i as an initial condition yeilds an x_i whose end point b_i satisfies the continuity condition (2.8). In mathematical terms, let $\phi_{\Delta t_i} : M \to M$ be the map that takes each a_i to b_i. Then multiple shooting seeks a_i that satisfy

$$\begin{bmatrix} \phi_{\Delta t}(a_{p-1}) - a_0 \\ \phi_{\Delta t}(a_0) - a_1 \\ \vdots \\ \phi_{\Delta t}(a_{p-2}) - a_{p-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2.9}$$

A common approach to approximating a solution to (2.9) is to apply Newton's Method:

1. Guess values for $a_i^0, i = 0, \cdots, p-1$.

2. Solve the linear system

$$J\Delta a^k = -r^k \tag{2.10}$$

where

$$J = \begin{bmatrix} -I & & & & & D\phi_{\Delta t}(a_{p-1}^k) \\ D\phi_{\Delta t}(a_0^k) & -I & & & & \\ & D\phi_{\Delta t}(a_1^k) & -I & & & \\ & & \ddots & \ddots & & \\ & & & D\phi_{\Delta t}(a_{p-2}^k) & -I & \end{bmatrix}, \tag{2.11}$$

$$\Delta a^k = \begin{bmatrix} \Delta a_0^k \\ \Delta a_1^k \\ \Delta a_2^k \\ \vdots \\ \Delta a_{p-1}^k \end{bmatrix} \tag{2.12}$$

and

$$r^k = \begin{bmatrix} r_0^k \\ r_1^k \\ r_2^k \\ \vdots \\ r_{p-1}^k \end{bmatrix} = \begin{bmatrix} \phi_{\Delta t}(a_{p-1}^k) - a_0^k \\ \phi_{\Delta t}(a_0^k) - a_1^k \\ \phi_{\Delta t}(a_0^k) - a_2^k \\ \vdots \\ \phi_{\Delta t}(a_{p-2}^k) - a_{p-1}^k \end{bmatrix} \tag{2.13}$$

3. Then compute

$$a^{k+1} = a^k + \Delta a^k \tag{2.14}$$

4. Repeat steps 2 and 3 until satisfactory accuracy has been achieved.

Inspection of equation (2.11) shows that the Jacobian matrix is a banded sparse matrix. With the exception of the non-zero element in the upper right hand corner, it has the structure of a lower block diagonal matrix. There are many strategies one could use to solve equation (2.10). Iterative methods have become a popular choice for solving sparse linear systems, especially when exact solutions are not required. Many iterative methods, however, require that the system be either diagonally dominant or Symmetric Positive Definant (SPD). Futhermore, preconditioning may be required to ensure efficient convergence. Direct methods are an alternative approach to iterative methods. As a general rule, direct methods tend to be too expensive (or too inefficient) for application to sparse systems. A noteworthy exception to this rule, however, is cyclic reduction. A general introduction to cyclic reduction methods is provided by Gander and Golub [39]. Cyclic reduction has also found utility as a conceptual tool for designing parallel algorithms for sparse linear systems {cite me}.

Consider an arbitrary matrix M of the same form as J in equation (2.11),

$$M = \begin{bmatrix} -I & & & & & A_1 \\ A_2 & -I & & & & \\ & A_3 & -I & & & \\ & & A_4 & -I & & \\ & & & \ddots & \ddots & \\ & & & & A_n & -I \end{bmatrix}, \tag{2.15}$$

where each $A_i, i = 1, \cdots, n$ are submatrices. Furthermore, assume that n is an

even number. Applying the principle strategy of cyclic reduction, let P be the permutation matrix collects even and odd numbered elements,

$$P \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{bmatrix} = \left[\begin{array}{c} x_1 \\ x_3 \\ \vdots \\ x_{n-1} \\ \hline x_2 \\ x_4 \\ \vdots \\ x_n \end{array}\right] \tag{2.16}$$

Then

$$PMP^{-1} = \left[\begin{array}{ccccc|ccccc} -I & & & & & 0 & & & A_1 & \\ & -I & & & & A_3 & 0 & & & \\ & & -I & & & & A_5 & 0 & & \\ & & & \ddots & & & & \ddots & \ddots & \\ & & & & -I & & & & A_{n-1} & 0 \\ \hline A_2 & & & & & -I & & & & \\ & A_4 & & & & & -I & & & \\ & & A_6 & & & & & -I & & \\ & & & \ddots & & & & & \ddots & \\ & & & & A_n & & & & & -I \end{array}\right] \tag{2.17}$$

Thus, given a linear system

$$Mx = r \tag{2.18}$$

one can use P to transform this into

$$PMP^{-1}x = Pr \tag{2.19}$$

$$\Rightarrow \begin{bmatrix} -I & F \\ D & -I \end{bmatrix} \begin{bmatrix} x_o \\ x_e \end{bmatrix} = \begin{bmatrix} r_o \\ r_e \end{bmatrix} \tag{2.20}$$

where F is the submatrix corresponding to the partition in the upper right of (2.17), and D is the diagonal submatrix that correspond to the partition on the lower left of (2.17). By forming the Schur compliments of x_o and x_e, equation (2.20) may be convertex into a block diagonal system

$$\begin{bmatrix} FD - I & 0 \\ 0 & DF - I \end{bmatrix} \begin{bmatrix} x_o \\ x_e \end{bmatrix} = \begin{bmatrix} r_o + Fr_e \\ r_e + Dr_o \end{bmatrix} \tag{2.21}$$

Consider the matrix expression $FD - I$,

$$\begin{aligned} FD - I &= \begin{bmatrix} 0 & & & & A_1 \\ A_3 & 0 & & & \\ & A_5 & 0 & & \\ & & \ddots & \ddots & \\ & & & A_{n-1} & 0 \end{bmatrix} \begin{bmatrix} A_2 & & & & \\ & A_4 & & & \\ & & A_6 & & \\ & & & \ddots & \\ & & & & A_n \end{bmatrix} - I \\ &= \begin{bmatrix} -I & & & & A_1A_n \\ A_3A_2 & -I & & & \\ & A_5A_4 & -I & & \\ & & \ddots & \ddots & \\ & & & A_{n-1}A_{n-2} & -I \end{bmatrix} \end{aligned} \tag{2.22}$$

Likewise, consider $DF - I$,

$$DF - I = \begin{bmatrix} A_2 & & & & \\ & A_4 & & & \\ & & A_6 & & \\ & & & \ddots & \\ & & & & A_n \end{bmatrix} \begin{bmatrix} 0 & & & & A_1 \\ A_3 & 0 & & & \\ & A_5 & 0 & & \\ & & \ddots & \ddots & \\ & & & A_{n-1} & 0 \end{bmatrix} - I$$

$$= \begin{bmatrix} -I & & & & A_2A_1 \\ A_4A_3 & -I & & & \\ & A_6A_5 & -I & & \\ & & \ddots & \ddots & \\ & & & A_nA_{n-1} & -I \end{bmatrix} \tag{2.23}$$

Observe that both $FD - I$ and $DF - I$ have the same form as the original matrix M. Thus, one may solve equation (2.21) by recursively applying the permutation strategy just outlined to each subsystem and forming schur compliments. Let $A_{-j} = A_{n-j}$ for $j = 0, 1, \cdots, n-1$. Then, repeating this process m times, $0 \leq m \leq \log_2 n - 1$, and solving for x_i gives

$$x_i = \left(\prod_{j=1}^{2m} A_{i-j} \right) x_{i-2m} - \left[r_i + \sum_{j=l}^{i-1} \left(\prod_{l=j}^{i-1} A_l \right) r_j \right], \tag{2.24}$$

and for $m = \log_2 n$

$$x_i = - \left(I - \prod_{j=1}^{n-1} A_{i+n-j} \right)^{-1} \left[r_i + \sum_{j=l}^{i-1} \left(\prod_{l=j}^{i-1} A_l \right) r_j \right], \tag{2.25}$$

Observe that the diagonal elements in (2.24) involve products of A_i. Likewise, in the case of full application of the cyclic reduction process, represented by equation (2.25), the inverse matrix is the inverse of identity minus products of A_i. It

was identified by Hockney {cite me; see ref in Gander} that there are conditions under which the diagonal elements diminish at a quadratic rate, and that under such conditions one may be able to terminate the reduction process early (i.e. use $m < \log_2 n$) and ignore off-diagonal elements. This is called "incomplete cyclic reduction". It was shown by Heller {cite me}, incomplete cyclic reduction may be applied to diagonally dominate systems. Application of incomplete cyclic reduction to (2.18) yeilds the solution to be

$$x_i = -\left[r_i + \sum_{j=i-m}^{i-1} \left(\prod_{l=j}^{i-1} A_l \right) r_j \right]. \tag{2.26}$$

Recall the multiple shooting problem. Application of incomplete cyclic reduction to equation (2.14), gives

$$\begin{aligned} a_i^{k+1} &= a_i^k + \Delta a_i^k \\ &= a_i^k + r_i + \sum_{j=i-m}^{i-1} \left(\prod_{l=j}^{i-1} D\phi_{\Delta t}(a_l^k) \right) r_j \\ &= \phi_{\Delta t}(a_{i-1}^k) + \sum_{j=i-m}^{i-1} \left(\prod_{l=j}^{i-1} D\phi_{\Delta t}(a_l^k) \right) r_j. \end{aligned} \tag{2.27}$$

Observe that this has the same form as equation (2.3) in Theorem 1. This principle difference is that equation (2.27) is based on partitioning of the time domain, whereas (2.3) is based on a partitioning of the state space at Poincare sections. This is a subtle but important difference. Partitioning at the Poincare boundaries ensures that the continuity error is always orthogonal to the flow, and does not require one to know a-prior the fundamental period of the system.

2.5 The Connection to Parallel Time Integration Methods

{Summarize similarities and differences with parallel time integration methods}

{Unlike established time parallel integration methods, this new approach is a fully parallel algorithm; there are no serial steps}

2.6 Homotopy Relaxation

{FINISH}

2.7 Numerical Tests

2.7.1 2D Limit Cycle

2.7.1.1 Governing Equations

Consider a one-dimensional system governed by the nonlinear second order differential equation

$$\ddot{x} + 2\zeta(\omega x, \dot{x})\omega\dot{x} + \omega^2 x = 0, \tag{2.28}$$

where the function $\zeta : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is given by the equation of the form

$$\zeta(a, b) = c(a^2 + b^2 - 1), \tag{2.29}$$

for some constant $c \in \mathbb{R}$.

The foregoing second order differential equation (2.28) may be reduced to a

first order system. Let $y = \dot{x}$. Then $\dot{y} = \ddot{x}$, and equation (2.28) may be written as

$$\begin{bmatrix} \dot{x} \\ \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \\ -\omega^2 & -2\zeta(\omega x, y)\omega \end{bmatrix} \begin{bmatrix} x \\ \\ y \end{bmatrix}. \tag{2.30}$$

At any point $(x_0, y_0) \in \mathbb{R}^2$, the linearized form of (2.30) is given by

$$\begin{bmatrix} \dot{x} \\ \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \\ -\omega^2 - 4c\omega^2 x_0 y_0 & -4\omega c y_0^2 - 2\omega\zeta(x_0, y_0) \end{bmatrix} \begin{bmatrix} x \\ \\ y \end{bmatrix}. \tag{2.31}$$

It can be shown that for any $c > 0$, the the long term dynamics of the foregoing system are governed by a single periodic attractor centered on the origin of $\mathbb{R}^2$. A plot of the flow of the dynamical system represented by (2.30) is shown in Figure 2.1.

2.7.1.2 Numerical Model

There are a number of time marching methods one could use to effect a solution to equation (2.28). Implicit time marching methods are a practical necessity for URANS CFD as a means of overcomming the otherwise strict time step size limits required for numerical stability. Implicit integrators have also gained popularity in structural mechanics solvers. Thus, the implicit Euler method will be used for this study. (Higher order implicit methods are also of interest, and will be investigated in subsequent work.)

Let $x_n \in \mathbb{R}^n$ denote a state vector of an autonomous differential system given by equation (2.1) at time t_n. Given a time step h, the state x_{n+1} at time $t_{n+1} = t_n + h$ is implicitly defined by the equation

$$0 = hf(x_{n+1}) - x_{n+1} + x_n. \tag{2.32}$$

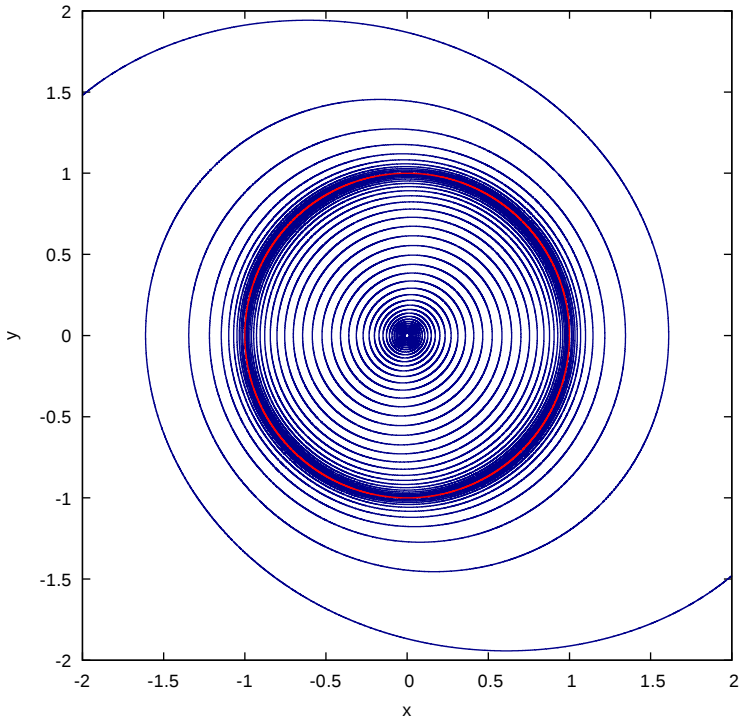


Figure 2.1: Flow of the dynamical system represented by equation (2.30). Transient behavior corresponds to blue curves. Periodic attractor is represented by the red curve.

Thus, given a solution x_0, the states along the trajectory $x : \mathbb{R} \to \mathbb{R}^n$ satisfying equation (2.1) and $x(0) = x_0$ may be approximated by repeated application of (2.32).

The time step h was chosen such that the periodic solution of the numerical model was visibly indistinguishable from that of the analytical solution when plotted. During the exploration process of finding a suitable value for h, it was observed that larger values of h tended to produce a numerical solution whose periodic attractor was of smaller radius than the true solution. However, even for large values of h, the center of the attractor correctly remains centered at $(0, 0)$, and all states are all

equidistant from the origin and equally distributed around the numerical attractor (the attractor of the discrete system remains circular).

Newton's method was used to solve (2.32) at each time level. Since the test problem is rather simple, the analyical Jacobian was used. Also, given the low dimensionality of this test problem, dense matrix operations together with direct inversion of the Jacobian was used. For larger "real-world" problems, one would typically use an approximate Jacobian, and seek to use an iterative linear solution method that exploits the sparse matrix structure that often results from discretization of PDEs. A fixed convergence tolerance of 0.5×10^{-6} was used to terminate the Newton iteration sequence at each time level, with the number of Newton iterations being allowed to vary as needed to meet this criteria.

The state advancement operator was formulated to be consistent with the implicit Euler time integration scheme. Consider a single time step, from t_n to t_{n+1}. Applying implicit differentiation with respect to x_n to equation (2.32) yields,

$$0 = h\frac{\partial f}{\partial x}\frac{\partial x_{n+1}}{\partial x_n} - \frac{\partial x_{n+1}}{\partial x_n} + I. \tag{2.33}$$

Let $A_{n+1} = \frac{\partial f}{\partial x}(x_{n+1})$. Furthermore, recognize that $\frac{\partial x_{n+1}}{\partial x_n}$ is the linear state advancement operator ϕ_{n+1} that takes x_n at time t_n to its linear approximation at t_{n+1}. Rewriting (2.33) using this notation and factoring gives

$$I = [I - hA_{n+1}]\,\phi_{n+1}. \tag{2.34}$$

Thus,

$$\phi_{n+1} = [I - hA_{n+1}]^{-1}. \tag{2.35}$$

It follows from the chain rule of Calculus that

$$\phi_{n+m,n} = \phi_{n+m} \cdots \phi_{n+1}\phi_n. \tag{2.36}$$

2.7.1.3 Results

First, the behavior of the multiple algorithm was explored for a range of time partition counts. All cases were executed using the serial algorithm. (The parallel algorithm is discussed later.)

The solution history for direct time marching and multiple shooting with $1, 2, 4, 8$ and 16 time partitions are shown in Figure 2.2. Comparing Figures 2.2(a) and 2.2(b), it can be seen that the two solutions are identical. This behavior is to be expected, as the correction to x_0 at each iteration is to simply assume the value of $x(T)$ from the previous iteration. Comparing Figures 2.2(b) through 2.2(f), it can be seen that increasing the number of time parititions reduces the initial periodicity error, while initial error between the numerical solution and analytical solution remains essentially the same. Thus, caution is needed when using the periodicity error as a measure of the numerical error. Likewise, if periodicity error is used as a stopping criterion for the multiple shooting method, one needs to be sure that a sufficiently small tolerance is used to avoid prematurely terminating the algorithm.

A comparision of convergence histories is shown in Figure 2.3. The "residual" on the y-axis of the plots is the l_2-norm of the periodicity error. As previously observed, the initial periodicity error of the solution is inversly proportional to the number of time partitions. It is also observed that the number of iterations required

to reach the target convergence level is of the same order of magnitude; in this case, 10-15 iterations are required.

DOUBLE CHECK: Inspection of the solution files (not shown) revealed that the total number of time steps executed by the 1-partition case was the same as that required by the direct time marching algorithm.

The foregoing results have used $k = N_p - 1$. This means that the correction of the initial conditions for each time partition is based on periodicity error information gathered at every partition boundary. It is natural to wonder, however, how sensitive is the convergence rate to this choice? Can one still realize an acceptable rate of convergence by using a smaller value of k? The convergence histories for values of k ranging from 1 to 15 with the number of time partitions fixed at 16 is shown in Figure 2.3. It can be observed that the effect of decreasing k from the maximum possible value of 15 to the minimum value of 1 causes corresponding increase in the number of iterations required to reach convergence. Thus, the benefits of using a reduced value of k, namely a reduction in the cost of computing the correction of initial conditions, must be weighted against the cost of additional iteration cycles. The optimal trade will depend on the particular techniques used to compute the state transition operators.

Given the foregoing observations, it can be concluded that the proposed multiple shooting method does provide a reliable means of computing periodic solutions, however, it is of little practical value when used as a basis for a serial algorithm: the computational work is not reduced by adding more time partitions, nor is the stability of the method improved in a meaningful way by increasing the number

of time partitions. The practical value of the multiple shooting method becomes much more apparent when considered as a parallel algorithm. This is illustrated in Figure 2.5, which shows the speed up realized by executing the proposed multiple shooting algorithm in parallel. The number of time partitions was taken to be equal to the number of MPI processes, and each MPI process was assigned to one time partition. The maximum allowable value of k was used in each test case. It can be seen that the parallel speedup of the proposed multiple shooting algorithm is nearly linear over the range of processors available on the test machine. Furthermore, the parallel efficiency of this algorithm for this test case remains above 90%. This is quite good, considering the low dimensionality of the problem being solved.

Algorithm 1 A serial algorithm for implementing the computational steps of the numerical method identified in Section 2.2.

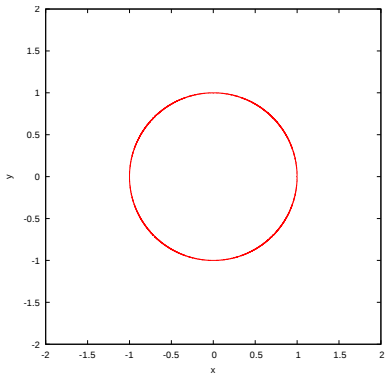
Require: ϵ, $kmax$, INITIALIZE, TIMESTEP, ATPOINCARESECTION

```
1: procedure SHOOT(X, E)
2:     for i = 0 to p − 1 do
3:         x ← X[i]
4:         repeat
5:             x ← TIMESTEP(x)
6:         until ATPOINCARESECTION(x)
7:         E[i] ← x − X[MOD(i + 1, p)]
8:     end for
9: end procedure

10: procedure CORRECT(X, E)
11:     Δx ← 0
12:     for k = 0 to kmax do
13:         for i = 0 to p − 1 do
14:             r ← E[i]
15:             for j = i to i + k − 1 do
16:                 r ← S_j^{l(j+1)} r
17:             end for
18:             Δx[MOD(i + k, p)] ← Δx[MOD(i + k, p)] + r
19:         end for
20:     end for
21:     X ← X + Δx
22: end procedure

23: INITIALIZE(X)
24: loop
25:     SHOOT(X, E)
26:     if max_{0≤i<p} ‖E[i]‖ < ε then
27:         EXIT
28:     end if
29:     CORRECT(X, E)
30: end loop
```

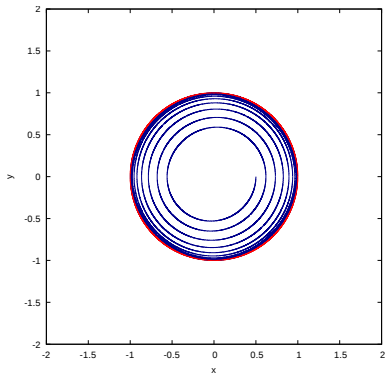
Algorithm 2 A parallel algorithm for implementing the computational steps of the numerical method identified in Section 2.2.

Require: ϵ, $kmax$, INITIALIZE, TIMESTEP, ATPOINCARESECTION, SHIFTRIGHT

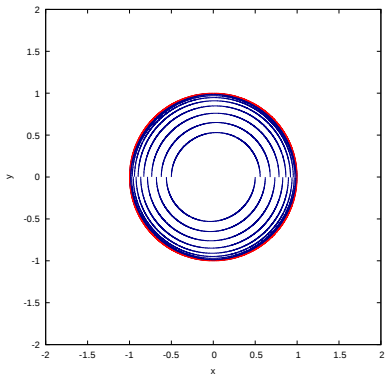
```
1:  procedure SHOOT(X, E)
2:      x ← X
3:      repeat
4:          x ← TIMESTEP(x)
5:      until ATPOINCARESECTION(x)
6:      SHIFTRIGHT(x)
7:      E ← x − X
8:  end procedure

9:  procedure CORRECT(X, E)
10:     Δx ← 0
11:     k ← 0
12:     r ← E
13:     loop
14:         SHIFTRIGHT(r)
15:         Δx = Δx + r
16:         k = k + 1
17:         if k > kmax then
18:             BREAK
19:         end if
20:         r = Sr
21:     end loop
22:     X ← X + Δx
23: end procedure

24: INITIALIZE(X)
25: loop
26:     SHOOT(X, E)
27:     if max_{0≤i<p} ‖E[i]‖ < ε then
28:         EXIT
29:     end if
30:     CORRECT(X, E)
31: end loop
```



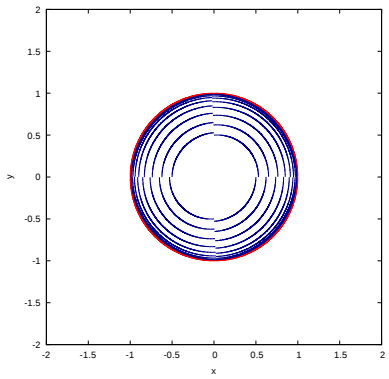
(a) Direct time marching.



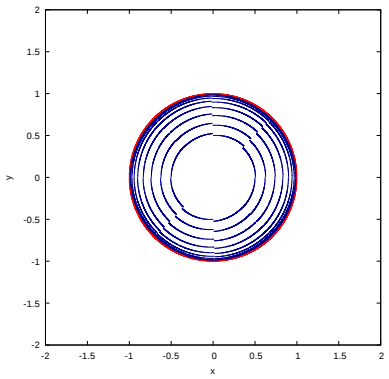
(b) 1 time partition.



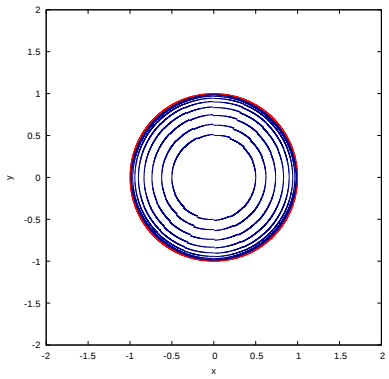
(c) 2 time partitions.



(d) 4 time partitions.



(e) 8 time partitions.



(f) 16 time partitions.

Figure 2.2: Solution histories for direct time marching and multiple shooting with $1, 2, 4, 8$ and 16 time partitions. The exact solution is denoted by the red curve. The numerical solution history is shown by the blue curves.

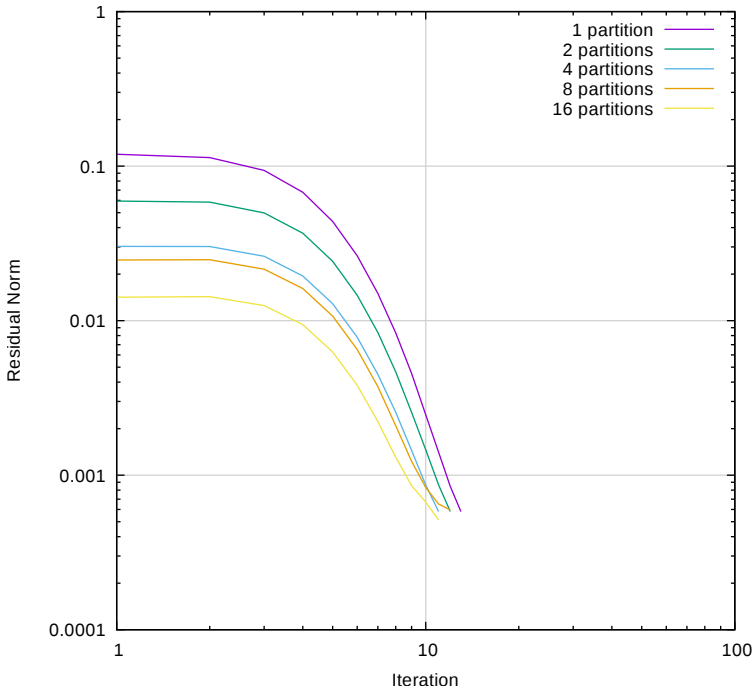


Figure 2.3: Convergence history for 1, 2, 4, 8, and 16 time partition test cases.

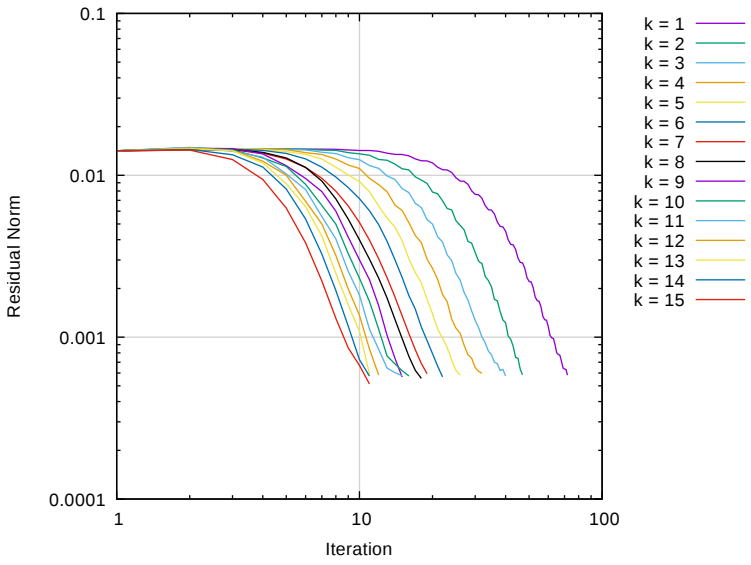


Figure 2.4: Convergence history as a function of k with number of time partitions fixed at 16.

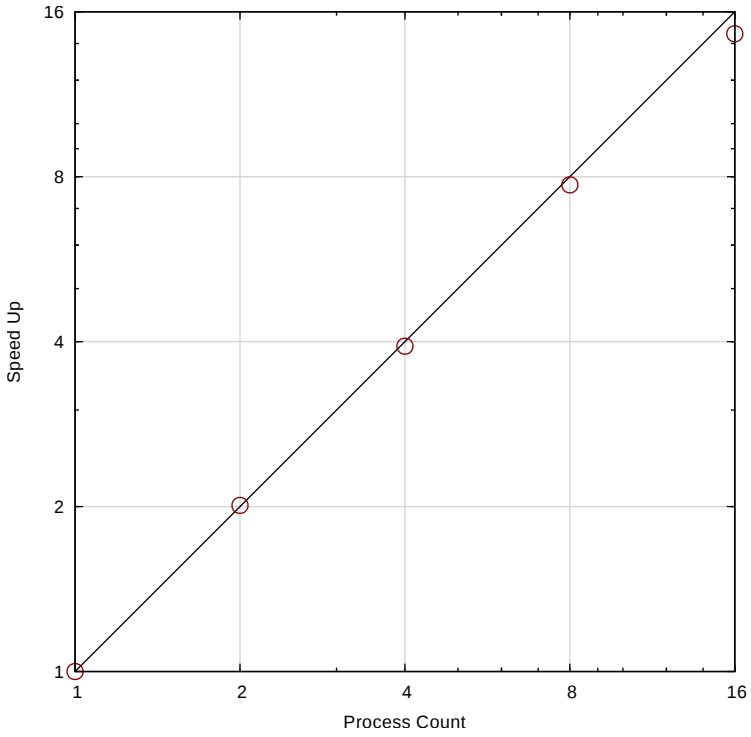


Figure 2.5: Parallel speedup.

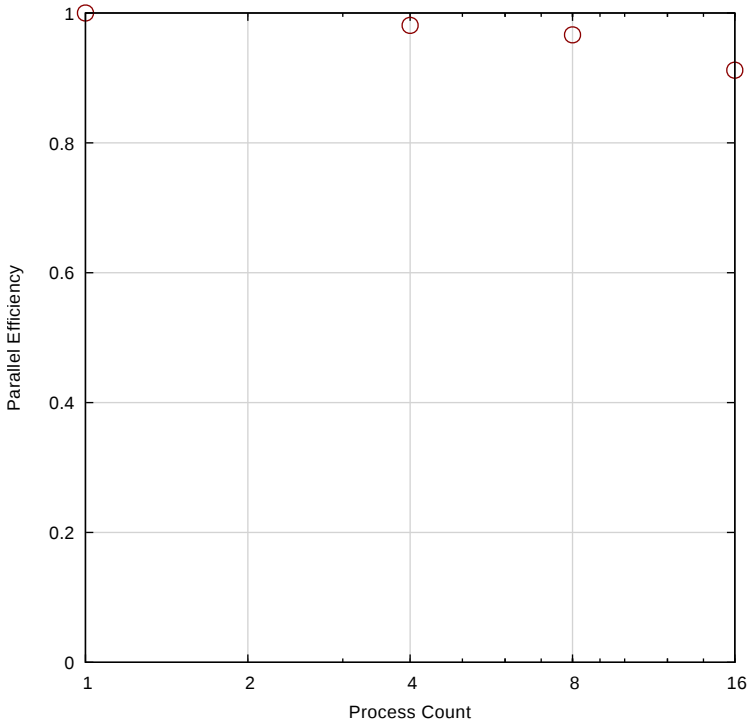


Figure 2.6: Parallel efficiency.

Chapter 3: Periodic Multiple Shooting for Rotorcraft Trim

3.1 The Trim Problem

A distinguishing characteristic of helicopter aeromechanics analysis from fixed-wing aeroelastic analysis, is that even the most basic of aerodynamic or structural dynamic analysis requires one to determine the fixed-stick control inputs that brings the rotor system into equilibrium for the flight condition under consideration. That is, one must "trim" the rotor. The task of finding the control parameters and rotor states that correspond to a given operating state is refereed to as a "trim problem". One must also solve a trim problem as a prerequisite to performing a stability analysis of a rotor. Thus the theory and practice of solving the rotor trim problem stands as a corner stone of rotorcraft aeromechanics analysis.

3.2 State-of-the-Art Helicopter Trim Theory and Practice

For over the past 30 years, researchers have investigated the trim problem and various solution methods. Despite this, there are still many open questions regarding the mathematical theory of helicopter trim. Furthermore, the demand for new, more effective, solution methods are continually driven by the growing

mathematical complexity of rotor models and evolving computer architectures.

In regards to the theory of trim, there does not exist a universal theory or method [31]. Each type of trim problem posses unique mathematical characteristics which demand different solution approaches.

A survey of the current theory and solution methods is as follows.

3.2.1 Mathematical Formulations of Helicopter Trim

The mathematical formulations of the trim problem as they have appeared in the literature are summarized. The key observation is that the trim problem corresponds to unique kind of differential algebraic system of equations–one that is unlike the differential algebraic equations found commonly found in mechanics. In particular, the constraints that express the desired trim conditions are functions of temporal averages rather than instantaneous values.

3.2.1.1 System of ODE's with Integral Constraints

In it's simplest form, the trim problem may be mathematically expressed as follows: Given a known functions $f : \mathbb{R}^n \mapsto \mathbb{R}^n$, $y : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^m$, known vector $\bar{y} \in \mathbb{R}^m$, and known period $T \in \mathbb{R}$, find $x : \mathbb{R} \mapsto \mathbb{R}^n$ and $c \in \mathbb{R}^m$ such that

$$\dot{x}(t) = f(x(t), t, c), \quad \forall t \in \mathbb{R}, \tag{3.1}$$

$$x(0) = x(T), \tag{3.2}$$

and

$$\bar{y} = \frac{1}{T}\int_0^T y(x(\tau), c)d\tau. \tag{3.3}$$

Equation (3.1) represents the system of differential equations that govern the dynamics of the rotor (vehicle), with x corresponding to the states of the rotor (vehicle) and c the vector of control parameters. Typically,f is periodic with respect to the time parameter t. If the vector c were known, one could hypothetically determine x using only equations (3.1) and (3.2). Equation (3.3) has been referred to as the "trim constraint" [31]. Physically, it usually either represents some kind of force and moment balance or a set of constraints on vehicle accelerations, with y corresponding to the instantaneous observation of forces and moments or vehicle accelerations, and $\bar{y}$ the corresponding mean value. The period T is typically taken to be period of the main rotor, or average period in the case of a system with multiple rotors operating at non-harmonic periods [31].

The dynamics of the rotor and vehicle typically involve second order differential equations. However, all such systems can be converted for first order. In fact, systems with higher order time derivatives can be converted to first order [36]. Thus, the above form subsumes all autonomous formulations with explicit states (assuming fixed controls).

Since the control parameters are taken to be constants, the foregoing system of equations could be interpreted to represent a family of autonomous systems that are parameterized by c, and the equation of constraint (3.3) representing a desired set of system properties.[THIS IS NEW; FACTOR OUT INTO SEPARATE THESIS

CHAPTER?]

[NEED TO CITE RELAVENT PAPERS WHERE THIS FORM WAS USED; SUMMARIZE RANGE OF MODEL FIDELITIES CONSIDERED]

3.2.1.2 Explicit System of DAE's with Integral Constraints

Detailed models of rotor and vehicle systems may result in the appearance of additional constraints. In recent years, multi-body dynamics has gained increasing popularity as a means of modeling complex rotor systems. Depending on the level of generality, a multi-body dynamics formulation may employ the use of Lagrange multipliers. This leads to a modification of the trim problem: Given a known functions $f : \mathbb{R}^n \mapsto \mathbb{R}^n$, $g : \mathbb{R}^n \mapsto \mathbb{R}^{n'}$, $y : \mathbb{R}^n \times \mathbb{R}^{n'} \times \mathbb{R}^m \mapsto \mathbb{R}^m$, known vector $\bar{y} \in \mathbb{R}^m$, and known period $T \in \mathbb{R}$, find $x : \mathbb{R} \mapsto \mathbb{R}^n$, $\lambda : \mathbb{R} \mapsto \mathbb{R}^{n'}$ and $c \in \mathbb{R}^m$ such that

$$\dot{x}(t) = f(x(t), \lambda(t), t, c), \quad \forall t \in \mathbb{R}, \tag{3.4}$$

$$0 = g(x), \tag{3.5}$$

$$x(0) = x(T), \tag{3.6}$$

and

$$\bar{y} = \frac{1}{T} \int_0^T y(x(\tau), \lambda(\tau), c) d\tau. \tag{3.7}$$

3.2.1.3 Hamiltonian Formulations with Integral Constraints

Hamiltonian formulations have appeared in the literature as a basis for formulating Finite Element models of helicopter rotors[CITE ME]. In the context of trim, one augments the Hamiltonian with a "trim constraint" as is done for ODE's. Thus, the Hamiltonian formalism is only applied to the dynamics of the rotor system– without regard to the concept of trim. [GIVE EQUATIONS and EXPLAIN.]

3.2.1.4 Semi-Discrete Formulations with Integral Constraints

With the exception of Hamiltonian formulations of rotor systems with elastic blades, little work has been done with respect to trim solutions of systems governed by hyberbolic PDE's. In particular, aerodynamic modeling based numerical solutions to the Euler or Navier-Stokes Equations. In literature, the prevailing assumption is that one employs a semi-discrete (i.e. method of lines) approach[CITE ME]to reduce the system of PDE's to a system of ODE's. Then one theorizes about the existence the solutions and/or selects a solution methodology as though one were solving a system of ODE's[CITE PETER]. This is not necessarily wrong. However, it remains as an open question if one can exploit certain characteristics of the underlying PDE's system to realize a more computationally efficient and robust method for effecting trim solutions. Furthermore, modern approaches to PDE may involve adaptive meshing, which implies a variable number of state variables. Thus, the assumption that a system governed by a PDE can be equated to a ODE has its limits.

3.2.1.5 Systems with Hidden States

As identified by Peters[CITE ME], some aerodynamic models may have "hidden states". A common example of this is a dynamic stall model that contains time delays.

The significance of hidden states is that one cannot directly construct a linear approximation to the model by way of computing derivative of model with respect to the state vector. This has implications both towards (asymptotic) stability analysis as well the application of iterative methods based on successive linear approximations (e.g. Newton's Method, Periodic Shooting).

3.2.1.6 Under-determined systems

Advanced rotorcraft design concepts may allow for more control parameters than there are trim targets. This results in an under-determined system of equations. An example of this is a helicopter with variable RPM[CITE ME]. [cite/summarize recent AHS forum paper by RPI guy. He also explored optimal trim of a compound helicopter.]

One typically effects a solution to this class of problems by treating imposing the usual trim constraints (e.g. force moment balance) together with some kind of optimization criteria (e.g. power minimization).

3.2.2 Existence, Uniqueness, and Solvability

Very little has been done with respect to determining the conditions under which the trim problem has a solution, if the solution is unique, and if it is possible to determine a solution through a sequence of finite steps (or at least approximate it to arbitrary accuracy). In practice, one chooses a method that seems to have a probability of working, and tries it. If they fail to find a solution, the initial guess is adjusted or another method is tried. If they continue to fail to find a solution, one simply gives up and concludes that either there is not a solution or the problem cannot be solved using known methods.

Peters [31] has proposed conditions under which a DAE model of a rotor system is solvable. The solvability conditions proposed by Peters are based on the sufficiency conditions for Newton-like and Auto-Pilot algorithms to be converge. Unfortunately, none of these conditions can be easily checked in practice.

3.2.3 Trim Methods and Algorithms

3.2.3.1 Closed Form Control Relations

One strategy for determining the trim state of a helicopter (or rotor) is to seek a closed form expression of the control parameters as a function of the trim targets.[NOTE: Maybe outline the general process here as a sequence of steps expressed as mathematical operations. Then go on to explain its limitations.] Once such an expression is found, then one can eliminate the unknown control parame-

ters from the governing equations. Unfortunately, such expressions have only been found for very simple models. Thus, this approach may only be used to approximate the trim solution of more sophisticated models. When the control parameters are approximated in this way, the resulting accuracy has been reported to be generally inadequate for loads, vibration, and stability analysis[NOTE: NEED CITATION]. Furthermore, one still must decide how to solve the differential equations that govern the vehicle/rotor response. That said, one could consider using such relations to determine the "initial guess" of a more accurate iterative method.

An example of such an approach is that used by Wei for predicting control settings of a trimmed rotor operating in the "stalled regime" (as cited in [23]).

$$\theta_0 = \left(6\bar{C}_T + \frac{9}{8}\phi\right) + \left(15\bar{C}_T + \frac{19}{16}\phi\right)\mu^2 \tag{3.8}$$

$$\theta_s = -\left(16\bar{C}_T + \frac{3}{2}\phi\right)\mu - \left(16\bar{C}_T - \frac{3}{4}\phi\right)\mu^3 \tag{3.9}$$

$$\theta_c = \frac{\gamma}{p^2}\left[\left(\bar{C}_T + \frac{1}{48}\phi\right)\mu - \left(\frac{5}{9}\bar{C}_T\frac{1}{16}\phi\right)\mu^3\right] \tag{3.10}$$

where

$$\phi = \frac{4}{3}\left[\frac{1}{2}\left(\sqrt{C_T^2 + \mu^4} - \mu^2\right)\right]^{\frac{1}{2}} \tag{3.11}$$

and

$$\bar{C}_T = \frac{C_T}{a\sigma}. \tag{3.12}$$

The formulas were reported by Peters [23] to be "inadequate for the stalled conditions." It is interesting to observe that equations (3.8), (3.9), and (3.10) are independent of the rotor states.

3.2.3.2 Newton-like Methods with Time Marching

A common approach to the trim problem is to iteratively solve for the control vector and rotor (vehicle) response as follows:

1. Initialize approximate control vector c.
2. Compute the rotor (vehicle) response with c held constant by marching the system to "steady state".
3. Evaluate the trim constraint.
4. If the trim constraint is satisfied then stop.
5. Update the estimate for the control vector $\tilde{c}$ using a Newton-like scheme.
6. Return to step 2.

Mathematically speaking, this is equivalent to defining the function $X : \mathbb{R} \times \mathbb{R}^m \mapsto \mathbb{R}^n$ by the equation

$$X(t, c) = x(0) + \int_0^t f(x(\tau), \tau, c) d\tau \tag{3.13}$$

Then define the function $F : \mathbb{R}^m \mapsto \mathbb{R}^m$ by the equation

$$F(c) = \bar{y} - \frac{1}{T} \int_0^T y(X(t, c), c) dt, \tag{3.14}$$

and iteratively compute an approximate solution for c by applying a Newton-like Algorithm to equation (3.14). The rotor (vehicle) response x is determined as a side effect of the iterative solution for c.

The "loop" implied by step 6 is sometimes called the "trim loop", and the derivative $\frac{\partial F}{\partial c}$ is sometimes called the "trim Jacobian".

In practice, one does not typically use a closed form expression for $\partial F/\partial c$ but rather approximates this with a finite difference approximation([this is not Newton's method but is called...?]). Furthermore, equations (3.13) and (3.14) are written assuming that $x(0)$ is a point in the periodic attractor. In practice, this is not the case. Rather, one chooses $x(0)$ based on a convenient guess (e.g. zero, or the last known state from the previous iteration) rather than knowledge of the periodic attractor's location. This means that one needs to integrate equation (3.13) until the solution appears sufficiently periodic to approximate the periodic solution.

The principle advantage of this method is its ease of implementation, and modular architecture. That said, there are a number of detractors.

One detractor of this method may only be used to determine stable trim solutions[CITE ME]. This is a direct consequence of determining the rotor (vehicle) response by solving an Initial Value Problem. If the periodic attractor is unstable, then unless $x(0)$ is in the attracting set, the solution will diverge[CITE ME]. In practice, $x(0)$ is chosen based on a convenient guess–which rarely happens to coincide with the attracting set. Thus, if the penultimate goal is to support stability analysis, a different method will need to be used. Also, if one is interested in free-flight trim, the governing equations of the vehicle will need to be augmented with fictitious springs and dampers to stabilized the system[CITE ME].

Another detractor of this method is its computational cost. Depending on how $x(0)$ is chosen, the cost of evaluating equation (3.13) can be quite high. Furthermore,

if finite differencing is used to compute $\partial F/\partial x$, then the number of evaluations of equation (3.13) increases. One may try to offset the cost of the finite differencing by reusing the computed derivative during subsequent iteration steps (this is technically a chord method[CITE ME]). However, this strategy is appropriate when near the solution and it will reduce the convergence rate[CITE ME].

In general, Newton's Method is not a globally convergent method[CITE ME]. It has been empirically shown that this is indeed the case when Newton's Method is used in this application[CITE ME]. In fact, it has been shown that Newton's Method will fail even when the initial guess is close to the solution[CITE ME]. It has been speculated that $\partial F/\partial c$ can become singular or ill-conditioned when near the solution. This carries the implication that care is needed in choosing the initial estimate for the control vector c, and/or one needs to employ some kind of a globalization scheme. A globalized form of Newton's Method, known as "Damped Newton's Method" has been shown to overcome convergence issues related to ill-conditioning of $\partial F/\partial c$[CITE ME].

Sekhar et. al. have hypothesized that the basin of attraction for Newton's Method, when applied to the helicopter trim problem, is a fractal set [40]. This hypothesis has been extrapolated from the work of Epureanu [41], Haruta [42], and Jeong [43], which explored Newton's method in a different context. Sekhar performed a series of numerical experiments using a comprehensive (aeroelastic) rotor model with free-wake to explore this hypothesis. The test results support the hypothesis. However, the test results support other hypothesis as well; for example, that the basis of Newton's method is non-convex, or that is is not simply con-

nected. Neither of these alternative hypothesis imply that the basis of attraction have a fractal structure. Using similar numerical models, Gouravaraju et. al. have attempted to estimate the Hausdorff-Besicovitch dimension of the hypothesized fractal basin [44]. However, their conclusions are drawn based on the presupposition that the basin is in fact fractal. Either way, the work of Sekhar does demonstrate that the basin of attraction for Newton's Method is non-trival, and supports the notion that one must look elsewhere if a robust method is desired.

[Any citations for use of this method in industry codes such as RCAS or CAMRAD?]

3.2.3.3 Periodic Shooting

Periodic shooting is general purpose iterative method for finding periodic solutions to ordinary differential equations; i.e. finding a solution to equation [need to ref ode without control] (3.1) and (3.2). In its simplest form, periodic shooting seeks to find the initial condition x_0 as root of the residual function $r : \mathbb{R}^n \mapsto \mathbb{R}^n$ defined by the equation

$$r(x_0) = x(T) - x_0. \tag{3.15}$$

The solution to equation (3.15) is then iteratively computed by applying Newton's Method (or a variation thereof), with the evaluation of x_T by integrating the ODE from the current estimate for x_0. This may be expressed as the following algorithm:

1. Choose an initial estimate for x_0.

2. Compute x_T by integrating equation (3.1) from the initial condition x_0.

3. Evaluate $r(x_0) = x_T - x_0$.

4. If $r(x_0) = 0$ (or $r(x_0) < \epsilon$, for some small number $\epsilon > 0$) stop.

5. Update the estimate for x_0 using Newton's Method:

$$x_0 \leftarrow x_0 - \left[\frac{\partial x_T}{\partial x_0} - I\right]^{-1} r(x_0)$$

6. Goto step 2.

The Periodic Shooting method has been applied to both autonomous ODE's and ODE's with periodic coefficients. For autonomous ODE's, periodic shooting is ill-posed [1]. This is a consequence of the fact that any point on the periodic attractor is a solution. Reithmeier [2] has proposed strategies for overcoming this problem, as well as strategies for treating cases where the period T is unknown (e.g. finding limit cycles).

An alternative formulation of the Periodic Shooting method is based on the concept of a state transition matrix. Given a linear system with periodic coefficients

$$\dot{x}(t) = A(t)x(t), \tag{3.16}$$

there exists a matrix $\Phi(t, t_0)$ that maps $x(t_0)$ to $x(t)$, where x is a solution of (3.16). The matrix $\Phi(t, t_0)$ is called a state transition matrix. In the case of a nonlinear system, one can hope to locally approximate the nonlinear system by equation (3.16), and use the state transition matrix as a means to correct the initial condition $x(0)$ so as to satisfy the periodicity constraint (3.2). That is, given a means of determining

$\Phi(T,0)$, one can alternatively express step 5 as

$$x_0 \leftarrow x_0 - [\Phi(T,0) - I]^{-1} r(x_0).$$

This approach has been investigated by Friedmann [3] for computing rotor aeroelastic response and stability analysis. As shown by Friedmann, one of the noteworthy aspects of this alternative formulation lies in the fact that one can determine the stability characteristics of a rotor based on the Eigenvalues of $\Phi(T,0)$. [Mention Bauchau's stability analysis paper here?]

The periodic shooting method was first generalized as a method for finding solutions to the trim problem by Peters [4]. The principle modification is to augment the periodic constraint (3.2) with the trim constraint (3.3):

$$r\left(\begin{bmatrix} x_0 \\ c \end{bmatrix}\right) = \begin{bmatrix} r_1(x_0,c) \\ r_2(x_0,c) \end{bmatrix} = \begin{bmatrix} x_T - x_0 \\ \bar{y} - \frac{1}{T}\int_0^T y(x(\tau),c)d\tau \end{bmatrix} \tag{3.17}$$

Then apply Newton's method as before to the augmented system:

1. Choose an initial estimate for x_0 and control parameters c.
2. Compute x_T by integrating equation (3.1) from the initial condition x_0.
3. Evaluate residual (3.17)
4. If $r(x_0,c) = 0$ (or $r(x_0,c) < \epsilon$, for some small number $\epsilon > 0$) stop.
5. Update the estimate for x_0 and c using Newton's Method [check this]:

$$\begin{bmatrix} x_0 \\ c \end{bmatrix} \leftarrow \begin{bmatrix} x_0 \\ c \end{bmatrix} - \begin{bmatrix} \frac{\partial x_T}{\partial x_0} - I & \frac{\partial r_1}{\partial c} \\ \frac{\partial r_2}{\partial x_0} & \frac{\partial r_2}{\partial c} \end{bmatrix}^{-1} r\left(\begin{bmatrix} x_0 \\ c \end{bmatrix}\right)$$

6. Goto step 2.

In practice, the Jacobian matrix appearing in step 5 is computed using a finite difference approximation, rather than an exact formulation [4].[cite other PS refs here.]

As highlighted by Peters, one of the appealing aspects of Periodic Shooting is that the method may be applied to unstable systems, and upper left partition of the Jacobian appearing in step 5 is an approximation of the state transition matrix, and therefore can be used to deduce the stability characteristics of the rotor. Thus, Periodic Shooting provides is very attractive for comprehensive rotor analysis, where one is interested in predicting rotor response, loads and stability. Furthermore, Periodic Shooting allows one to reuse rotor models that have been formulated in the time domain. The disadvantage of Periodic Shooting is primarily its computational cost. Another potential issue is that Periodic Shooting cannot, in principle, be applied to models that posses hidden states (e.g. a dynamic stall model with pre-programed time delays).

A modified version of the Periodic Shooting algorithm for distributed parallel architectures was proposed by Subramanian et. al. [5]. The parallel algorithm assumes that one is using a finite difference approximation of the Jacobian, and that the Jacobian is a dense matrix. Given these assumptions, the algorithm works by associating each element of the Jacobian matrix with a distinct process, and exploiting the fact that each of these processes can compute the necessary function integrations for each of their respective state vectors and state perturbations inde-

pendently of other processes. It was shown that the resulting algorithm has a high degree of parallel efficiency. It should be noted, however, that this method assumes that the size of the rotor model is such that it can fit within the memory limits of each process. This may not be the case, for example, for a large CFD simulation.

It can be observed that the Periodic Shooting method assumes a system with a fixed number of state variables. For rotor aeromechanic models represented by PDE's, it may be desirable to incorporate solution adaptation in the numerical discretization. This would implies a variable number of state variables. A generalization of Periodic Shooting that can accomodate a variable number of states has yet to be developed.

[augment with other ps refs.]

[periodic shooting requires a fixed number of states. No go for adaptive PDE methods.]

[Elaborate on the basic (non-trim) method, example applications, and it's history (including generalizations to parallel algorithms)?]

[Clarify link between transition matrix and Newton Jacobian]

[Parallel Periodic Shooting Algorithm.]

3.2.3.4 Fast Floquet

The Fast Floquet method is a variation of Periodic Shooting that exploits the symmetry of rotor dynamics to reduce computational cost. Given a rotor with N_b identical blades, the response of each blade is identical to that of the blade that

precedes it (or follows it). Therefore, one can apply a similar solution strategy as outlined for Periodic Shooting, but instead of computing a full rotor revolution at each iteration, one only needs to integrate $1/N_b$ revolutions of the rotor. Furthermore, one can compute $\Phi(T, 0)$ directly from $\Phi(T/N_b, 0)$ using this symmetry[check this]. The Fast Floquet method was originally identified by McVicar [6]. Peters identified that the Fast Floquet Method generally gives improved numerical accuracy of the rotor Eigenvalues than the conventional Periodic Shooting Method [7].

A Parallel version of the Fast Floquet algorithm was proposed by Subramanian et. al. [8]. The method is a direct extension of their parallel periodic shooting algorithm [5] that exploits the symmetry relationship of the Fast Floquet method.

[Any missed points/refs from literature?]

3.2.3.5 Harmonic Balance

Like Periodic Shooting, Harmonic Balance is a general purpose family of methods for computing periodic solutions to differential equations. The principle strategy of Harmonic Balance is to approximate the solution as a partial sum of a Fourier Series,

$$x(t) = A_0 + \sum_{k=1}^{N} \left[A_k cos(k\omega t) + B_k sin(k\omega t) \right]. \tag{3.18}$$

Substitute equation (3.18) into the governing equation. After extensive application of trigonometric identities and asymptotic expansions of nonlinear terms, one then

should be able to algebraically express the result in the form

$$\begin{aligned} 0 =& f_0(A_0, \cdots, A_N, B_1, \cdots, B_N) \\ &+ \sum_{k=1}^{N} f_k(A_0, \cdots, A_N, B_1, \cdots, B_N) sin(k\omega t) \\ &+ \sum_{k=1}^{N} g_k(A_0, \cdots, A_N, B_1, \cdots, B_n) cos(k\omega t). \end{aligned} \tag{3.19}$$

Then one imposes the condition that each f_k and g_k are identically zero. This results a system of $2N + 1$ algebraic equations, plus an initial condition, that one tries to solve for $A_0, \cdots, A_N$ and $B_1, \cdots, B_N$, and ω. If ω is known, then one can drop the need for an initial condition.

Nayfeh and Mook [9] considered nonlinear single degree of freedom systems governed by an equation of the form

$$\ddot{x} + f(x + u_0) = 0, \tag{3.20}$$

Their solution procedure begins by using a Taylor series expansion to f to transform (3.20) into

$$\ddot{x} + \sum_{n=1}^{N} \alpha_n x^n = 0, \tag{3.21}$$

where each α_n is a Taylor series coefficient. Then a solution of the form

$$x(t) = \sum_{m=0}^{M} A_k cos(\omega m t + m\beta_0) \tag{3.22}$$

is assumed, where β_0 and A_1 are determined by initial conditions, and the remaining harmonic coefficients $A_0, A_2 \cdots, A_M$ and fundamental frequency ω are determined by harmonic balance. They remark that predicting ω with this method can be very sensitive to the number of harmonics included in the approximation.

The harmonic balance method, as an perturbation method, has been used to predict limit cycle flutter of two-dimensional airfoils. Shahrzad and Mahzoon [10] compared flutter speeds as predicted by a first order harmonic balance method, the center manifold theory, and theory of normal forms. They found that the harmonic balance can predict flutter speed for airfoil models with nonlinear stiffness with good accuracy, as compared to direct numerical integration with fifth order Runge-Kutta. Liu and Dowell [11] have investigated the application of harmonic balance to the dynamic response of an airfoil with freeplay control surfaces, with the aerodynamics modeled using a state-space unsteady aerodynamic model.

Cheung and Lau [12] proposed the Incremental Harmonic Balance (IHB) method for strongly nonlinear problems in structural mechanics. The IHB method is based on the Incremental Hamiltonian method together with what the authors call a "time-space strip procedure." However, it appears as though one can arrive at the same outcome by applying Newton-Homotopy method[CITE ME]to the standard Harmonic Balance equations, with the fundamental frequency ω and Fourier amplitudes of the generalized forces taken as homotopy parameters. (The IHB method presumes that one knows the fundamental frequency of the system.) The IHB method does not, in general, provide guidance on how to choose the size of the homotopy increments, nor how each homotopy parameter ought to be increased relative to one another. Cheung and Chen [13] later demonstrated the applicability of IHB to more general periodic systems with cubic non-linearities, and demonstrated the combined use of IHB and Floquet theory to determine the stability and response of a periodic system.[elaborate?] Raghothama and Nrayanan [14] demonstrated the

application of IHB to the limit cycle analysis of a pitching-plunging airfoil with cubic stiffness.[Great. What did they find?]

Hall et al. [15] investigated the application of the harmonic balance method to spatially and temporally periodic flows modeled by the Euler equations and the 2D RANS equations with the Spalart Almaras turbulence model. The fundamental excitation frequency was assumed to be known a-priori. Their initial approach was based on a direct application of the harmonic balance method to the governing equations with periodic boundary conditions. They reported that the "method produced accurate unsteady flow solutions." However, the cost of the analysis increased proportional to N^3, where N is the number of harmonics included in the analysis. Thus, they proposed a modified harmonic balance method that allows one work directly with the flow variables known on a temporal grid with $2N + 1$ points, but with a spectral approximation to the time derivative used in place of conventional finite-differencing. Despite working directly with physical flow variables, the method does require the use of Discrete Fourier Transforms to be used during the iteration process to enforce boundary conditions. A pseudo time marching method was used to solve the discrete equations. They applied the method to the flow of a front stage transonic rotor of a "modern" high-pressure compressor, for both steady and unsteady flow. The unsteady flow was excited by prescribed blade oscillations at a single frequency. The authors found that 3 to 5 harmonics were required to accurately predict the zeroth and first harmonics of the solution. The authors also reported that the method failed to converge when 7 harmonics were included. (The authors did not proceed to try including 8 or more harmon-

ics.) Despite this, the method was shown to be capable of capturing the combined presence of steady shocks waves and flow separation, and to predict the nominal chord-wise pressure distribution along the upper and lower surfaces of the blades (2D airfoils in this case). The unsteady analysis was reported to be approximately 10 times the cost of a steady-state analysis. No discussion or demonstration of the application of this method to parallel computer architectures was given. Thomas et at. [16] investigated the application of this method to the prediction of Limit-Cycle Ocillations arising from 2D airfoil flutter. The wing was modeled as a rigid 2D body mounted on a spring damper system, with the NLR 7301 airfoil geometry. Three harmonics were used in the analysis. The predicted LCO parameters showed commensurate agreement with the cited predictions of others, but the predicted LCO pitch amplitude did not appear to correlate well with the measured amplitude from wind tunnel experiment.

Liu et al. [17] restated the foregoing harmonic balance method in more general terms and called the method the "High Dimensional Harmonic Balance" method or "HDHB" method, and compared the HDHB method to the classical Harmonic Balance (HB) method using the Duffing equation as a model problem. It was shown that the HB and HDHB methods are identical when applied to linear systems (or nonlinear systems whose amplitude of motion is small), but nonlinear terms in the governing equations can lead to significant differences in approximation for the same finite number of harmonics. Despite the computational benefits of the HDHB method, it was shown that the HDHB method can yield non-physical solutions. Furthermore, it was concluded that, in order to obtain the accuracy of the HB

method, one needs to use approximately twice as many harmonics in the HDHB method as used in the HB method. Liu et al. [18] arrived at these same conclusion when using the Van Der Pol oscillator as a model problem. Application of the HDHB method to rotorcraft aerodynamics was investigated by Ekici et al. [19]. The flow was modeled using the Euler equations (inviscid, compressible flow), and periodic boundary conditions in space and time were used. They showed that it is possible to obtain reasonable predictions of the chordwise pressure distribtions of the Carodona-Tung rotor in hover and non-lifting forward flight when at least 5 harmonics are used. They also demonstrated the application of the HDHB method to lifting forward flight with prescribed blade motions and control inputs. Trim analysis was not performed. Application of the HDHB method to parallel computer architectures was not addressed.

Generalization of the Harmonic Balance method to the trim problem generally follows the same strategy as other generalizations of periodic solution techniques to trim: apply the standard discretization or approximation techniques as one would if one were directly solving for a periodic solution, augment the resulting system of algebraic equations with the trim constraint, then solve the augmented system using standard iterative methods for nonlinear equations. An example application of the harmonic balance to trim analysis is demonstrated by the work of Hsu and Peters [20]. Despite Harmonic Balance being frequently mentioned in the literature on helicopter trim, there does not appear to be any work published in the last 30 years that directly investigates the computational effectiveness of various generalizations of the Harmonic Balance method to the trim problem for high fidelity rotor models

and/or modern computer architectures. This may be due to the fact that Harmonic Balance does not scale well with respect to the number of harmonics, risk of spurious solutions, difficulties formulating the Harmonic Balance equations for complex models, or lack of a perceived path to realizing an efficient parallel algorithm.

3.2.3.6 Finite Element in Time

Application of the Finite Element in the time domain, often refered to as "Finite Element in Time" or simply "FET", was introduced for rotorcraft applications by Borri [21]. Since then it has gained some popularity for directly computing periodic response of rotors. While one could simultaneously solve the FET discretized rotor equations and the trim constraints, a more common approach is to solve the trim constraints using a separate Newton-like iteration.

[THIS SECTION COULD USE SOME ELABORATION. What is the appeal of FET? Has anyone investigated simultaneous FET/Trim solution?]

3.2.3.7 Auto Pilot

The Auto Pilot is a family of methods that approach the trim problem by modeling the rotor (vehicle) controls as time dependent functions governed by a feedback controller. The feedback controller is designed in such as way as to enforce the trim constraint in the limit of $t \to \infty$. The entire rotor (vehicle) system, together with the control laws, are then integrated from a user specified initial condition until the rotor (vehicle) states and control values reach a periodic state (or a reasonable

approximation there of). Mathematically speaking, one would solve an initial value problem of the form

$$\dot{x}(t) = f(x(t), \theta(t), t) \tag{3.23}$$

$$\tau\ddot{\theta} + \dot{\theta} = ACM\left(y_* - y(x(t))\right) \tag{3.24}$$

$$x(0) = x_0 \tag{3.25}$$

$$\theta(0) = \theta_0 \tag{3.26}$$

where x are the system states and θ is the control vector, y_* is the desired trim targets, and y is the observed values of the trim targets. There have been a number of variations on this theme proposed throughout the literature.

The Auto Pilot method of helicopter trim was first proposed by Peters [22]. Peters proposed the following control relation

$$\tau\ddot{\theta} + \dot{\theta} = ACM\Delta, \tag{3.27}$$

where $\Delta \in \mathbb{R}^m$ is the observed difference between the trim metrics (e.g. instantaneous rotor thrust and hub moments) and their respective target values, $CM \in \mathbb{R}^{m\times m}$ is a "coupling matrix", $A \in \mathbb{R}^{m\times m}$ is gain matrix, and $\tau \in R^{m\times m}$ is a diagonal matrix of time constants. Peters stated that the coupling matrix CM is formulated based on an a-priori understanding of the system behavior, but it does not need to be exact; it only needs to cause θ to move in the correct direction. The term $ACM\Delta$ may be thought of as defining a nominal rate of change in θ as a function of Δ. The term $\tau\ddot{\theta}$ serves to filter changes in θ that would otherwise be driven by the vibratory components of Δ (i.e. one only wishes the time averaged value of Δ to become zero). The foregoing control scheme was tested for a

simple rotor model with a rigid blade with flap and torsion degrees of freedom (no lag), and a quasi-steady aerodynamic model with uniform inflow. Only one blade was modeled. The governing equations were integrated using a "predictor corrector method." For purposes of numerical testing, the time constants and gains for θ_c and θ_s were taken to be equal, thereby reducing the number of control parameters to be empirically determined (i.e. "tunned"). Peters then proceeded to graphical and formal numerical optimization methods to determine the optimum time constants and gains. When manually tunning an autopilot, one increases the gains to increase the speed at which the controller converges, while increasing the time constants as needed to ensure stability and eliminate control oscillations. Peters found that the optimum value for the time constants and gains were approximately equal to the period of one rotor revolution. However, these values do not in general ensure that the control parameters reach constant values; small oscillations in the controls may still be present. One can reduce control oscillations by increasing the time constants, but this comes at the price of increasing the time required to trim the rotor. Peters also compared the autopilot to periodic shooting. He concluded that the autopilot was computationally more efficient than periodic shooting when the rotor system is moderately to heavily damped. (In this case, the rotor did not have a lag degree of freedom, so this condition was satisfied.) However, this cost comparison does not include the cost of tunning the autopilot, and presumes that a sufficiently accurate CM matrix can be derived. Furthermore, the cost analysis of Periodic Shooting is based on the assumption that finite differencing and dense matrix operations are used. Peters noted that the autopilot is limited to stable rotor

systems, whereas periodic shooting may be used for unstable systems. Thus, if one is ultimately interested in performing rotor stability analysis, the autopilot method cannot be used.

Peters and Chouchane [23] investigated the effect of dynamic stall on trimmed rotor response using both the auto-pilot method and a set of closed form relations derived using the harmonic balance method. The Unified Lift model was used to model the aerodynamics of the rotor blades, including the effects of dynamic stall. The Unified Lift model is based on an ODE formulation for the unsteady lift of an airfoil. The aerodynamic model does not model the unsteady drag that often accompanies dynamic stall. The authors formulated an elastic model for the rotor blades, but only used rigid flap and lag modes in their numerical study. A "predictor corrector" method was used to integrate the system equations. The authors explored a range of thrust and advance ratios. The authors found that the approximate closed form relations that they had derived using the harmonic balance method were generally inadequate for trimming the rotor, and that the Auto Pilot method was comparatively more reliable.

Investigation of the effect of dynamic stall on trimmed rotor response was continued by Peters et. al. [24]. In this follow on study, they included torsional degree-of-freedom (in addition to lag and flap). As in the previous study, they considered a moment trim of the rotor. The auto-pilot was tunned using a combination of gradient based numerical optimization and graphical methods. The authors observed that the settling time of the controller can exhibit discontinuous behavior as a function of the time constants and gains. This made it difficult to apply gradient

based numerical optimization methods to tune the controller. The authors suggested that if one wished to manually tune an auto-pilot, a good strategy is to first increase the gains of the controller until stability limits are reached, then proceed to increase the time constants to eliminate control oscillations. It was also stated that one generally wants to keep the time constants of the controller small enough so that the natural period of the controller is about 3-4 rotor revolutions. The authors found that the optimal control parameters for hover differed significantly from those for forward flight. Thus, auto-pilot parameters depend on both the rotor design and flight condition. The authors also reported difficulties tuning the auto-pilot for rotors that were soft in torsion. They speculated that part of the difficulty may be related to some of the simplifying approximations used to formulate the CM matrix. However, they also speculated that an adaptive control scheme may ultimately be needed. Finally, the authors concluded that the auto-pilot method is not appropriate for effecting trim solutions near stability boundaries.

Enns [25] demonstrated the feasibility of training a neural network for trimming a model of an Apache helicopter. Their principle aim was to investigate the application of a neuro-control scheme as part of a flight control system for a physical helicopter. One might consider directly using this strategy as a means of overcoming the limitations of auto-pilot method proposed by Peters. However, the cost of training the neural network is quite high. It was reported that the neural network required 1000 training epochs to effect a trim solution, where the time duration for each training epoch needs to be long enough to observe the steady response of the rotor for a given control input. That said, the control architecture and training

methods used by Enns is just one of many possibilities.

Riviello [26] proposed an neuro-autopilot scheme that is a composition of a low cost analytical reference model together with a neural network that is trained online to learn the error between the reference model and the detailed helicopter model to be trimmed. In principle, if the reference model is properly formulated, the amount of input-output data samples required to train the neural network ought to be small. This implies that the total time required to effect a trim solution ought to be small. The control scheme proposed by Riviello is a form of Nonlinear Model Predictive Control (NMPC). The rotor model to be trimmed is presumed to be a high-fidelity flexible-multibody dynamics model with the aerodynamics modeled using unsteady lifting models or CFD. The principle strategy is to determine an "optimal" control schedule that maneuvers the rotor (vehicle) from an initial given state into the desired trim state by solving an optimal control problem using a computationally inexpensive model of the rotor (vehicle) input-output behavior. The model used by the control optimizer is formulated as a composition of a simple reduced order model and a Neural Network. The purpose of the Neural Network is to correct the error between the simple reference model and the high fidelity rotor (vehicle) model. Training of the Neural Network is performed online using the standard back-propagation training algorithm. The purpose of the analytically formulated reference model is to reduce the amount of input-output samples that need to be collected to train the Neural Network. The Neural Network is updated at the end of each prediction horizon, during which the error between predicted plan input-output behavior and the actual system are monitored and used in subsequent

Neural Network training updates. After each Neural Network update, a new control schedule is recomputed, and the simulation resumes using the new control schedule. Riviello demonstrated the NMPC scheme for a flexible multi-body dynamics model of the UH60 rotor. The rotor controls were modeled using direct actuation of the pitch bearing; the details of the swash-plate and pitch-link control system were not modeled. The aerodynamics were modeled using simple strip theory. The numerical tests demonstrated that the NMPC method can effect a trim solution of the foregoing UH60 rotor model at advanced ratios of 0 and 0.35. It was also demonstrated that is is possible to tune a classical auto-pilot to yeild the same performance as the NMPC controller. However, this requires a great deal of user trial-and-error. Furthermore, optimal parameter selection for a classical auto-pilot is typically sensitive to flight condition. The NMPC scheme, on the other hand, is self "tunning" and is easier to use in practice. These conclusions were later shown to also hold for a flexible multi-body dynamic model coupled with a more detailed lifting line analysis and dynamic inflow model by Bottasso and Riviello [27].

3.2.3.8 Hybrid Schemes

A principle strategy used in engineering design is to identify primitive solutions to a problem, catalog their respective strengths and weaknesses, then seek to create a new solution as a composition of these primitive solutions in a way that retains their combined strengths while eliminating their weaknesses. This design strategy has been applied to a trim methods; however, research along these lines has not

been exhaustive.

Kim and Celi [28] explored a hybridization of the harmonic balance and periodic shooting method. The method proceeds by using harmonic balance to compute an approximate trim solution. Then, using the approximation solution computed using harmonic balance, the periodic shooting method is used to refine the estimate. This composition serves to mitigate the computational cost of periodic shooting and the inherent accuracy limitations of the harmonic balance method. This presumes that the cost of Periodic Shooting is dominated by the number of iterations, and that Since neither harmonic balance nor periodic shooting are limited to stable systems, the composite method is in principle capable of determine a solution to an unstable system. They successfully demonstrated their composite method for a full helicopter model in forward flight. The model was consisted of rigid body blade and fuselage models with blade-element theory and dynamic inflow aerodynamics. It was observed that relatively few iterations of periodic shooting were required to realize an a solution of acceptable accuracy; however, they did not provide a direct comparison between the cost of the hybrid method vs the cost of exclusively using either Periodic Shooting or Harmonic Balance.

Peters and Li [29] proposed a hybrid algorithm based on periodic-shooting and the auto-pilot method. The method composes Periodic-Shooting and the Auto-Pilot methods by way of a decision function that adaptively toggles between the two methods based on measured computational time and solution convergence. Periodic Shooting is used by default for the first iteration. The CM matrix of the Auto-Pilot in subsequent iterations is taken from the Jacobian computed for the most recent

application of Periodic Shooting. The time constants and gains of the Auto-Pilot must still be chosen by the analyst. The hybrid method was demonstrated for a "rigid blade flap-lag model of a rotor with pitch and hub degrees of freedom." By varying the hub stiffness, it was possible to simulate (approximately) the effects of free-flight and fuselage elastic degrees of freedom on rotor trim. (The hub did not have translational degrees of freedom.) The hybrid method was succesfully effect a trim solution at an advance ratio of 0.3 and a thrust coefficient of 0.08 for the fixed-hub and simulated elastic-hub test cases. The hybrid method was found to yield a factor of 3 speed up for the test cases considered.

3.2.3.9 Optimal Trim

Advanced rotorcraft configurations such as compound helicopters or variable RPM helicopters often provide additional control parameters in excess of the number of trim constraints. This implies the existence of multiple trim solutions at each flight condition. In the case of variable RPM rotors with swashplate control there are infinitely many solutions.

Shank [30] proposed using a numerical optimization approach to solve this class of problems. It was shown that one can find trim solutions that minimize power. It was not shown (nor claimed), however, that this approach necessarily yields a unique optimal solution.

[ELABORATE. What about Peter's work? RPI guy's work?]

3.2.4 Trim of Non-Periodic Systems

Very little work has been done in the area of trim methods for non-periodic systems. It has been well established that multi-rotor systems, where each rotor operates at a non-integer frequency multiple of the other rotors, will exhibit a non-periodic response. In these cases, it has become common practice to simply treat the problem as having a periodic solution with a fundamental frequency equal to the average fundamental frequency of the rotors [31].

3.2.5 The Delta-Coupling Method

Application of Computational Fluid Dynamics (CFD) to problems in helicopter aeromechanics has exclusively been by way of the delta-coupling method. The delta-coupling method is not a trim method in of itself. It is merely an iterative means of correcting the airload models used by legacy comprehensive rotorcraft analysis tools based on CFD airload predictions. The comprehensive analysis tools solves the trim problem using one of the methods previously discussed.

The delta-coupling method was first introduced by Tung et. al. [32]. They sought a means of improving unsteady lift predictions on rotors in high speed forward flight, where unsteady transonic flow is significant. The delta-coupling method provided a convenient way to use Finite Difference approximations to the Transonic Small Disturbance equations to predict the lift on the advancing blade, and incorporate this information in CAMRAD, an existing comprehensive rotorcraft analysis tool. Potsdam et. al. [33] successfully demonstrated the application of the

delta-coupling method to prediction of UH-60 airloads and structural loads for high-speed, low-speed, and high-thrust flight conditions. In this case, the CFD analysis was OVERFLOW-D (an overset URANS solver) and the comprehensive analysis was CAMRAD II. The CAMRAD II aerodynamic model used a lifting line analysis method combined with free-wake, airfoil tables, and a dynamic stall model. A detailed review of the delta-coupling method and the beneficial impact of CFD analysis on rotorcraft aeromechanic is provided by Datta et. al. [34]. The delta-coupling method is currently used in the Helios program [35].

3.3 Theory

An adaptation of the forgoing general purpose methodology is presented for the special class of problems commonly referred to as the "rotor trim problem". The principle strategy is to convert the governing equations of the rotor system to an equivalent autonomous ODE. When properly converted, one can directly apply the Periodic Multiple Shooting method to determine the states and control values of the rotor system. Furthermore, one can realize a non-trivial cost reduction by modifying the method slightly to exploit symmetries within the state-space of the converted system. Such modifications can realize a cost reduction on the order of the number of blades in the rotor.

3.3.1 Equivalent Autonomous First Order System

The governing equations for a rotor system generally consist of a second order differential equation with periodic forcing. In addition, there is a trim constraint, which relates control parameters (e.g. collective and cyclic control angles) and system states to a set of desired, time averaged, system observations (e.g. rotor thrust and hub moments).

Reduction of high order differential equations to an equivalent first order system is a straightforward process that has been well documented in textbooks (e.g. the book by Hirsch et al. [36]). The treatment of periodic forcing terms has received some attention in the literature (e.g. for example Chapter 7 and 10 of Wiggens [37]). However, there is no canonical approach, and each approach introduces its own modeling tradeoffs. The treatment of integral constraints does not appear to have received any attention in the literature. Thus, the subsequent two sections will address each of these later two issues in turn.

{Do I need to emphasis that the equivalence only holds for the long-term dynamics, and not the transient behavior?}

{Do I need to address error bounds for a solution that is near the attractor of the equivalent DS and the original differential model?}

3.3.1.1 Treatment of Periodic Forcing Terms

Consider a non-homogeneous first order system governed by the equation

$$\dot{x} = f(x, \cos(\omega t), \sin(\omega t)). \tag{3.28}$$

The objective is to identify an autonomous system

$$\dot{y} = g(y) \tag{3.29}$$

together with a map χ such that $\chi(y)$ is a solution to (3.28) (i.e. $\chi(y) = x$), and y is both periodic and a continuous on $\mathbb{R}$. Futhermore, if equation (3.28) posses a single periodic attractor, then so should (3.29). {show that these requirements are sufficient to enable application of PMS?}

A strategy used by Wiggens [37] is based on the observation that the function $\theta : \mathbb{R} \to \mathbb{R}$, given by the equation

$$\theta(t) = \omega t \mod 2\pi, \quad \forall t \in \mathbb{R} \tag{3.30}$$

is the solution to the differential equation

$$\dot{\theta} = \omega. \tag{3.31}$$

Thus, one may rewrite (3.28) as

$$\dot{x} = f(x, \cos(\theta), \sin(\theta)), \tag{3.32}$$

$$\dot{\theta} = \omega. \tag{3.33}$$

The above system may be rewritten in the same concise form as (3.29), by taking $y = (x, \theta)$, and χ as the relation $\chi((x, \theta)) = x$ for all $(x, \theta) \in \mathbb{R}^n \times \mathbb{R}$. However, y is not continous on $\mathbb{R}$, because θ is not continuous on $\mathbb{R}$. If one attemps to avoid the discontinuity by taking $\theta(t) = \omega$, then θ fails to be periodic. Thus, one cannot use this strategy as a means of enabling the application of PMS. {Clarify that Wiggens did not have multiple shooting methods in mind when using this approach.}

Another strategy is to look for dynamical system representations of sine and cosine.

Proposition 4. *Suppose $g, h : \mathbb{R} \to [-1 : 1]$ are functions given by the equations*

$$g(t) = \cos(\omega t) \tag{3.34}$$

and

$$h(t) = \sin(\omega t), \tag{3.35}$$

for $\omega \neq 0$. Then, for all $t \in \mathbb{R}$, f and g may also be expressed as

$$g(t) = s_1(t), \tag{3.36}$$

$$h(t) = -s_2(t)/\omega, \tag{3.37}$$

where s_1 and s_2 are the coordinate functions of the trajectory, passing through the point $(1, 0)$ at $t = 0$, for the dynamical system given by

$$\begin{bmatrix} \dot{s_1} \\ \dot{s_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}. \tag{3.38}$$

Proof. Substitute equation (3.34) into (3.36), (3.35) into (3.37), and then solve for s_1 and s_2 to yeild

$$s_1 = \cos(\omega t), \tag{3.39}$$

$$s_2 = -\omega \sin(\omega t). \tag{3.40}$$

Setting $t = 0$ in the above equations shows that $(s_1, s_2) = (1, 0)$. Taking derivatives

with respect to time gives

$$\begin{aligned}\dot{s}_1 &= -\omega \sin(\omega t) \\ &= s_2 \\ \dot{s}_2 &= -\omega^2 \cos(\omega t) \\ &= -\omega^2 s_1.\end{aligned}$$

Writing the above equations in matrix form yeilds (3.38).

Conversly, it can be shown that equations (3.39) and (3.40) constitute a unique solution to (3.38) for initial conditions $(s_1, s_2) = 0$ [36]. □

By Proposition 4, one may then rewrite (3.28) as the autonomous system of differential equations

$$\begin{aligned}\dot{x} &= f(x, s_1, -s_2/\omega) \\ \dot{s}_1 &= s_2 \\ \dot{s}_2 &= -\omega^2 s_1\end{aligned}$$

The above system is periodic and continuous on $\mathbb{R}$. The only issue is that the above system does not posses a unique periodic attractor. In fact, there is a continuum of periodic solutions. Given a small perturbation in the initial condition, and solution will move to a higher (lower) orbit. {clarify? incorporate non-uniqueness into proposition?}

The proposed solution is to use a limit cycle model, to ensure that the periodic attractor of (3.28) (if it exists) remains unique.

Proposition 5. *Suppose $g, h : \mathbb{R} \to [-1 : 1]$ are functions given by the equations*

$$g(t) = \cos(\omega t) \tag{3.41}$$

and

$$h(t) = \sin(\omega t), \tag{3.42}$$

for $\omega \neq 0$. Then, for all $t \in \mathbb{R}$, f and g may also be expressed as

$$g(t) = s_1(t), \tag{3.43}$$

$$h(t) = -s_2(t)/\omega, \tag{3.44}$$

where s_1 and s_2 are the coordinate functions of the trajectory, passing through the point $(1, 0)$ at $t = 0$, for the (nonlinear) dynamical system given by

$$\begin{bmatrix} \dot{s_1} \\ \\ \dot{s_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ \\ s_2 \end{bmatrix} - \begin{bmatrix} 0 \\ \\ 2\omega\zeta(s_1, s_2) \end{bmatrix}, \tag{3.45}$$

$$\zeta(s_1, s_2) = \left((s_1)^2 + (s_2/\omega)^2 - 1\right) c. \tag{3.46}$$

Furthermore, the foregoing dynamical system has only one periodic attractor.

Proof. Equations (3.41), (3.42), (3.43), and (3.44) may be combined and solved for s_1 and s_2 to give

$$s_1 = \cos(\omega t),$$

$$s_2 = -\omega \sin(\omega t).$$

Substituting the above results into (3.46) gives

$$\zeta(s_1(t), s_2(t)) = 0$$

for all $t \in \mathbb{R}$. Thus, equation (3.45) reduces to (3.38). As was already shown for Proposition 4, the above expressions for s_1 and s_2 satisfy (3.38). Futhermore, setting $t = 0$ gives $(s_1, s_2) = (1, 0)$.

Uniqueness of the periodic attractor is now shown.{finish}

□

{Show how to treat phase difference between blades}

{Disscuss setting of initial conditions here, or in separate section?}

3.3.1.2 Treatment of Trim Constraints

3.3.2 Seeding Initial Conditions

{Discuss initial conditions here, or as paragraph in section on autonomous system?}

3.3.3 Exploiting Symmetries

It is common practice to design helicopter rotors with identical, equally spaced, blades. This design practice serves to mitigate aircraft vibration. A beneficial side-effect of this design practice, is that one can exploit the symmetry of the system to simplify the analysis. A classical example of this strategy put to practice is the Fast Floquet Method [6, 7], which was discussed in Chapter 1.

For the sake of brevity, a rotor system that consists of $N_b \in \mathbb{N}$ identical blades, equally spaced around the rotational axis of the rotor, will be said to be a "symmetric rotor."

Recall from Section 3.3.1 that we can generally construct a dynamical system model of a rotor system, which posses the same long-term dynamics as the original differential model; i.e. the periodic attractor, and flow on the periodic attractor, are the same.

Proposition 6. *Suppose $M \subset \mathbb{R}^n$ is the state space of a dynamical system $\phi : M \to M$, with periodic attractor $A \subset M$, whose behavior on A models the long-term dynamics of a symmetric rotor.* {*finish*}

Proof. {finish} □

3.4 Numerical Tests

3.4.1 Rigid Bladed Rotor with Flap DOF

As a first step to understanding the application of PMS to a rotor model and assesing its utility, a test using a simple model of a rotor in forward flight has been performed. Such a model is not adaquate for "real-world" engineering analysis of rotor loads or vibration. Key physical behaviors, such as flap-torsion coupling, and unsteady aerodynamic effects, are not represented. Nevertheless, use of simple models does help to facilitate an understanding of how a method succeeds (or fails) on a fundamental level. Futhermore, well-documented examples, which range in complexity, provide useful pedagotical waypoints for those who wish to assimilate these ideas and subsequently improve upon them or put them to practice.

Two methods were applied to this rotor model: a conventional time marching

solution strategy with an autopilot, and the PMS method (with autopilot). The results of the two methods are reported below. It will be seen that the PMS method yeilds an identical solution as the time marching method, with the added benefit of enabling parallel computation.

3.4.2 Rotor System Model and Flight Condition

The rotor system is an articulated rotor with 4 identical blades, equally spaced around the azimuth. The rotor radius was $R = 26.83$ ft. The rotational speed of the rotor was a fixed rate of $\omega = 27.0$ rad/sec.

Each blade had a single flap-wise degree-of-freedom. The offset of the flap hinge was $e = 2.683$ ft. The mass distribution of the rotor is a uniform value of $m = 1$ slug/ft^2. The platform was rectangular with a chord length of $c = 2.0$ ft. A linear lift coefficient was assumed with $c_l(\alpha) = 2\pi\alpha$, where α is the local angle of attack.

The aerodynamic loads acting on each rotor blade was modeled using quasi-steady strip theory. A uniform inflow model was used to model the influence of the rotor wake on the local angle of attack at each blade section.

The rotor system was "flown" at an advance ratio of $\mu = 0.2$ at standard sea level conditions. The shaft angle of attack was fixed at 6.0° (tilted forward).

3.4.2.1 Mathematical Model

The only degree-of-freedom in this model is the flap angle of each blade, β_i, $i \in \{1, \cdots, N_b\}$. Each β_i is assumed to belong to the interval $(-b, b)$, where $b \ll 1$ (rad). Thus, $\cos(\beta_i) \approx 1$ and $\sin(\beta_i) \approx \beta_i$.

Invoking the principle of angular momentum balance about the flap hinge yeilds the following second order differential equation for the flap response:

$$\ddot{\beta}_i + \Omega^2 \nu_\beta \beta_i = \frac{1}{I_\beta} M_\beta(\beta_i, \dot{\beta}_i, \theta_i, \cos(\psi_i), \sin(\psi_i)) \tag{3.47}$$

where $I_\beta \in \mathbb{R}$ is the (second) moment of inertia of the blade about the flap hinge, $S_\beta \in \mathbb{R}$ is the first moment of inertia about the flap hinge, $\Omega \in \mathbb{R}$ is the rotational speed of the rotor, $\nu_\beta \in \mathbb{R}$ is the non-dimensional flap frequency

$$\nu_\beta = 1 + \frac{eS_\beta}{I_\beta}, \tag{3.48}$$

and $M_\beta : \mathbb{R}^5 \to \mathbb{R}$ is the function that maps local flap angle β_i, local flap rate $\dot{\beta}_i$, local pitch angle $\theta_i \in \mathbb{R}$, and local azimuth position, represented by $\cos(\psi_i)$ and $\sin(\psi_i)$, to the instaneous aerodynamic moment about the flap hinge. (The function M_β is mathematically defined over $\mathbb{R}^5$, but is only physically meaningful over small subset of $\mathbb{R}^5$.)

Assuming a strip theory model of aerodynamic forces, the aerodynamic moment is given by

$$M_\beta(\beta_i, \dot{\beta}_i, \theta_i, \cos(\psi_i), \sin(\psi_i)) = \frac{1}{2}\rho c c_{l\alpha} \Omega^2 R^4 \left(A_\theta \theta_i + A_\lambda \lambda + A_{\dot{\beta}} \dot{\beta}_i + A_\beta \beta_i \right), \tag{3.49}$$

where

$$A_\theta = \left(\frac{1}{4} - \frac{1}{3}\frac{e}{R}\right)(1 + 2\mu\sin(\psi)) + \left(\frac{1}{2} - \frac{e}{R}\right)(\mu\sin(\psi_i))^2, \tag{3.50}$$

$$A_\lambda = -\frac{1}{3} + \frac{1}{2}\frac{e}{R} - \mu\sin(\psi_i)\left(\frac{1}{2} - \frac{e}{R}\right), \tag{3.51}$$

$$A_{\dot\beta} = -\frac{1}{\Omega}\left(\frac{1}{4} - \frac{1}{3}\frac{e}{R} + \mu\sin(\psi_i)\left(\frac{1}{3} - \frac{1}{2}\frac{e}{R}\right)\right), \tag{3.52}$$

$$A_\beta = -(\mu\cos(\psi_i))\left(\frac{1}{3} - \frac{1}{2}\frac{e}{R}\right) - \left(\mu^2\sin(\psi_i)\cos(\psi_i)\right)\left(\frac{1}{2} - \frac{e}{R}\right). \tag{3.53}$$

The instanteous out of plain force $F_{z,i}$ contributed by blade i is given by

$$F_{z,i}(\beta_i, \dot\beta_i, \theta_i, \cos(\psi_i), \sin(\psi_i)) = \frac{1}{2}\Omega^2 R^3 \rho c c_{l_\alpha}\left(B_\theta\theta_i + B_\lambda\lambda + B_{\dot\beta}\dot\beta_i + B_\beta\beta_i,\right) \tag{3.54}$$

where

$$B_\theta = \frac{1}{3}\left(1 - \frac{e}{R}^3\right) + \left(1 + \frac{e}{R}^2\mu\sin(\psi_i)\right) + \left(1 - \frac{e}{R}\right)(\mu\sin(\psi_i))^2, \tag{3.55}$$

$$B_\lambda = -\frac{1}{2}\left(1 - \frac{e}{R}^2\right) + \left(1 - \frac{e}{R}\right)\mu\sin(\psi_i), \tag{3.56}$$

$$B_{\dot\beta} = -\frac{1}{\Omega}\left(\frac{1}{3}\left(1 - \frac{e}{R}^3\right) + \frac{1}{2}\left(1 - \frac{e}{R}^2\right)\mu\sin(\psi_i)\right), \tag{3.57}$$

$$B_\beta = -\frac{1}{2}\left(1 - \frac{e}{R}\right)\mu\cos(\psi_i) - \left(1 - \frac{e}{R}\right)\mu^2\sin(\psi_i)\cos(\psi_i). \tag{3.58}$$

The total instantaneous thrust is then

$$T = \sum_{i=1}^{N_b} F_{z,i} \tag{3.59}$$

Since the rotor is articulated (and there is no hinge spring or damper), the instan-

taneous hub moment is given by

$$M_x = \sum_{i=1}^{N_b} eT_i \sin(\psi_i), \tag{3.60}$$

$$M_y = -\sum_{i=1}^{N_b} eT_i \cos(\psi_i). \tag{3.61}$$

A simple swash-plate control model was used. For $i \in \{1, \cdots, N_b\}$, the local pitch angle θ_i is given by the equation

$$\theta_i = \theta_0 + \theta_{1c} \cos(\psi_i) + \theta_{1s} \sin(\psi_i). \tag{3.62}$$

The autopilot model is based on the original linear model proposed by Peters [24]. Denote the target thrust as T^* and target hub moments about the x and y axes as M_x and M_y, respectively. The feedback control law is given by

$$\tau \begin{bmatrix} \ddot{\theta}_0 \\ \ddot{\theta}_{1c} \\ \ddot{\theta}_{1s} \end{bmatrix} + \begin{bmatrix} \dot{\theta}_0 \\ \dot{\theta}_{1c} \\ \dot{\theta}_{1s} \end{bmatrix} = \begin{bmatrix} k_0 & 0 & 0 \\ 0 & k_c & 0 \\ 0 & 0 & k_c \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -b \\ 0 & -b & 1 \end{bmatrix} \begin{bmatrix} T^* - T \\ M_x^* - M_x \\ M_y^* - M_y \end{bmatrix}. \tag{3.63}$$

where

$$b = \frac{8\,(\nu_\beta - 1)}{\gamma}. \tag{3.64}$$

The control parameter τ is the time constant, and the control parameters k_0 and k_c are the gains corresponding to the collective and cyclic control parameters, respectively.

To facilitate the application of PMS, equations (3.47) and (3.63) were converted to equivolent first order differential equations, and $cos(\psi_i)$ and $\sin(\psi_i)$ was replaced with equivolent limit cycle model. {Give the resulting formula?}

3.4.2.2 Numerical Model

Trapezoidal integration was used for this test, as it is energy conserving. Use of a dissipative method, such as Implicit Euler, requires one to use an excessively small time step size to yeild a solution with acceptable accuracy. {Give the update formula? Describe how implicit system was solved?}

The linear state advancement model was based on a numerically consistent formulation.{elaborate, give formulation}

The time step size was fixed at $\Delta t = 2\pi/(\Omega n_s)$, where $n_s = 1024$. This time step corresponds to approximately 0.351° change in azimuth. Use of 1440 steps per cycle (a step size of 0.25° of azimuth), is not uncommon in rotorcraft analysis. The choice of 1024 steps per cycle was prefered over 1440 since it is a power of 2. Use of a power of 2 facilitates parallel scalability studies.

3.4.2.3 Implementation Notes

{Eliminate this section?}

3.4.2.4 Test Results

Direct integration of the govering equations was performed over a time interval equal to 100 rotor revolutions. Use of this method for computing rotor response in steady flight is well understood and commonly used in practice. Thus, the results of direct time integration may be used as a reference solution for assessing the relative accuracy and computational efficiency of subsequent Periodic Multiple Shooting

analysis.

The full time history of the flap response for all four blades is shown in Figure 3.1. It can be seen that exhibit qualitatively the same transient behavior, and that a periodic flap response is realized after 80 revolutions.

It can also be seen in Figure 3.1 that the local amplitude of the flap response passes through a minima near the 10th revolution. Looking at the corresponding control time history in Figure 3.2, one can see that autopilot initially moves the lateral cyclic in the wrong direction, and then begins to correct near rev 5. At rev 10, the lateral cyclic is passing through zero. During this time the nominal lateral tilt of the rotor is moving from left (port) to right (starboard). Thus the minima in local flap amplitude is due to the autopilot tilting the rotor laterally from left to right.

The autopilot requires about 80 rotor revolutions to determine the trimmed control angles. Figure 3.2 shows autopilot determining the lateral cyclic by revolution 30 (despite initially moving the lateral cyclic in the wrong direction), the collective by revolution 60, and longitudinal cyclic by revolution 80.

The correctness of the autopilots decision can be verified by inspecting the time histories for the thrust, hub moment about the x-axis, and hub moment about the y-axis, in Figures 3.3, 3.4, and 3.5, respectively.

The flap response over the final revolution is shown in Figure 3.6. Consistent with theory, the flap response of each blade has the same waveform as all other blades, and differs only in phase. The phase shift of each blade is 90° relative to the blade that precedes it. This is the value that one would expect for a 4 bladed rotor.

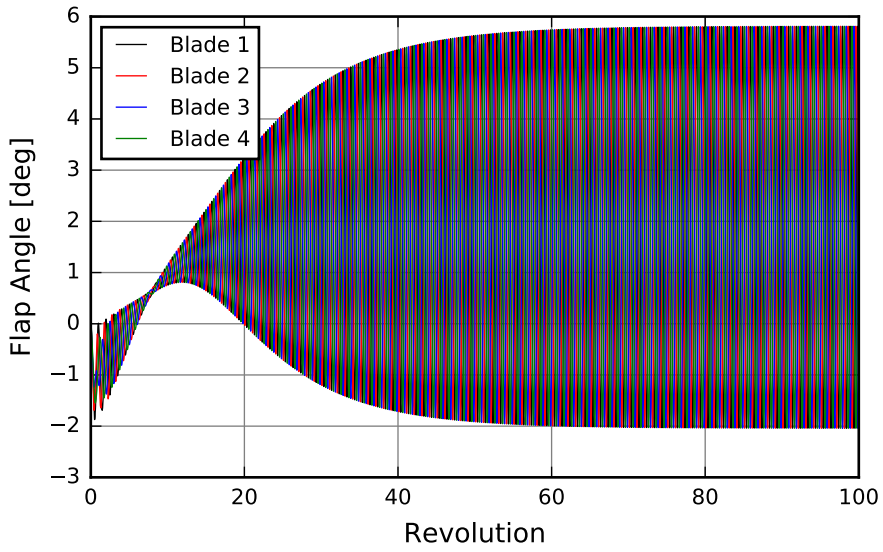


Figure 3.1: Flap response time history for all 4 blades.

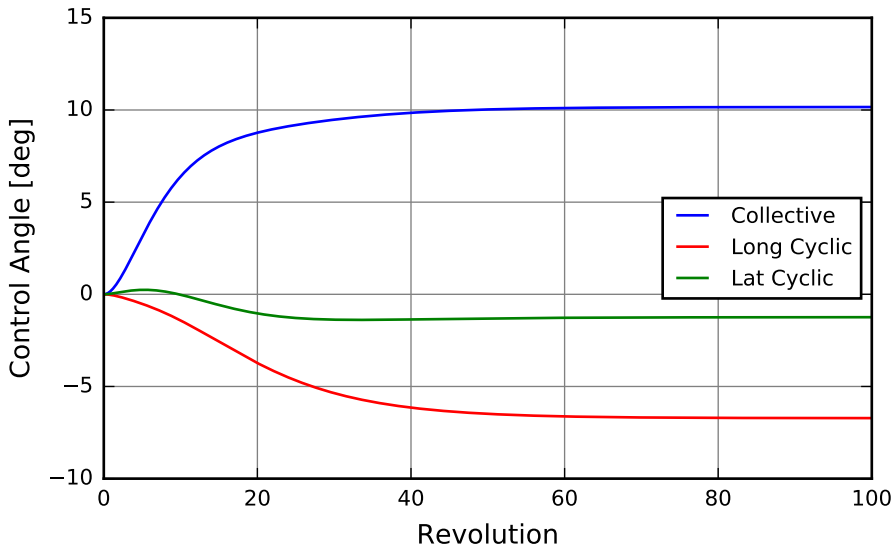


Figure 3.2: Control angle time histories.

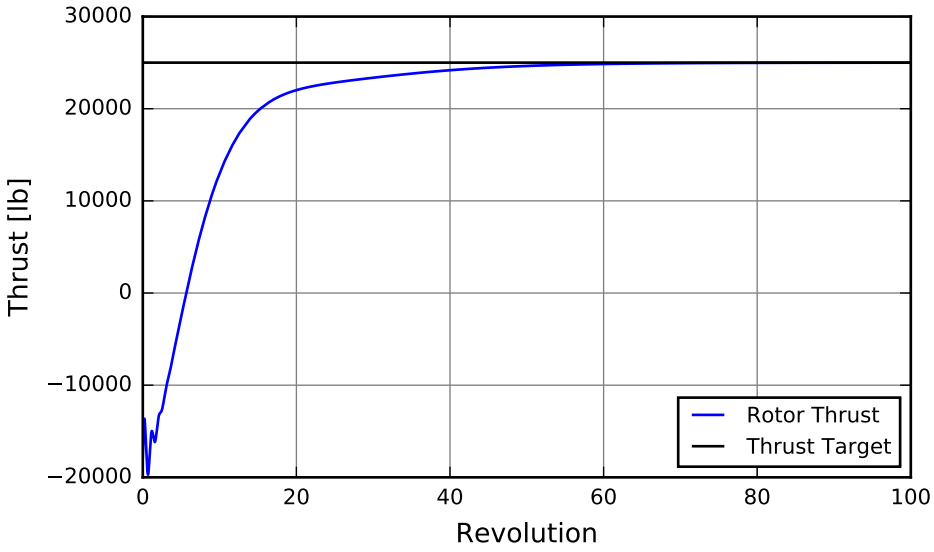


Figure 3.3: Thrust time history.

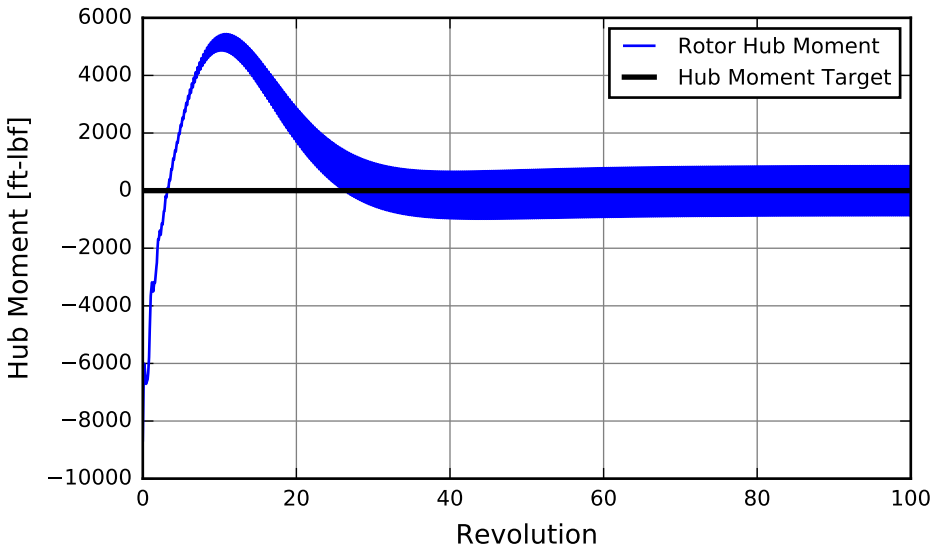


Figure 3.4: Hub moment time history about the x-axis.

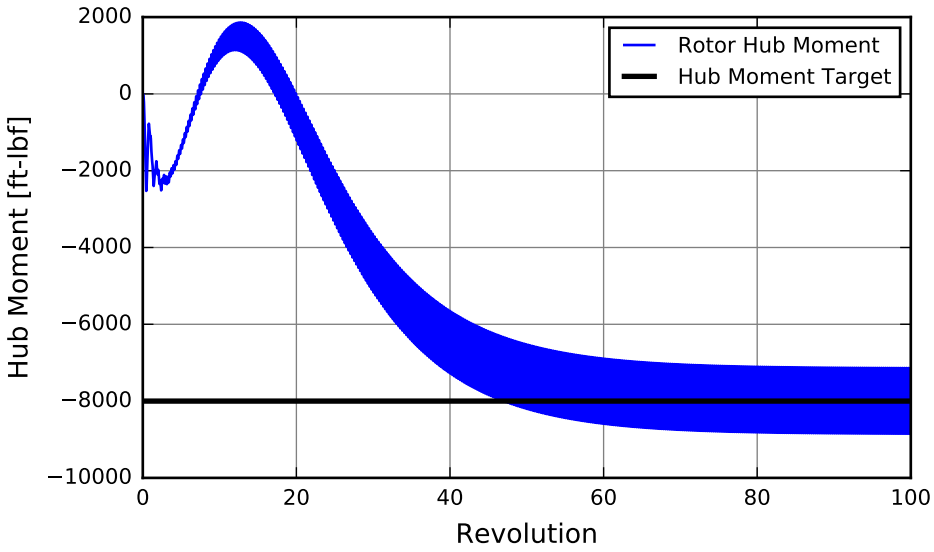


Figure 3.5: Hub moment time history about the y-axis.

The autopilot does converge and yeild constant control values, as shown in Figure 3.7. Figures 3.8, 3.9, and 3.10 confirm that the mean thrust, and hub moments were driven by the autopilot to the specified target values. Futhermore, it can be seen that the thrust and hub moments contain a 4p frequency, which is consistent with theory for a 4 bladed rotor.

The results of applying Periodic Multiple Shooting (PMS) are now considered. Four state-space partitions were used, and each partition was associated with a distinct computational process. Figure 3.11 shows the azimuth angle. Figures 3.12, 3.13, 3.14, and 3.15 show the flap response for blades 1, 2, 3, and 4, respectively. The control angles θ_0, θ_{1c} and θ_{1s} are shown in Figures 3.16, 3.17, and 3.18, respectively. The thrust is shown in Figure 3.19, and hub moment about the x-axis in Figure 3.20 and y-axis in 3.21. Each connected curve in the foregoing figures represents an intermediate solution of the PMS method on a partition of the state-space. There

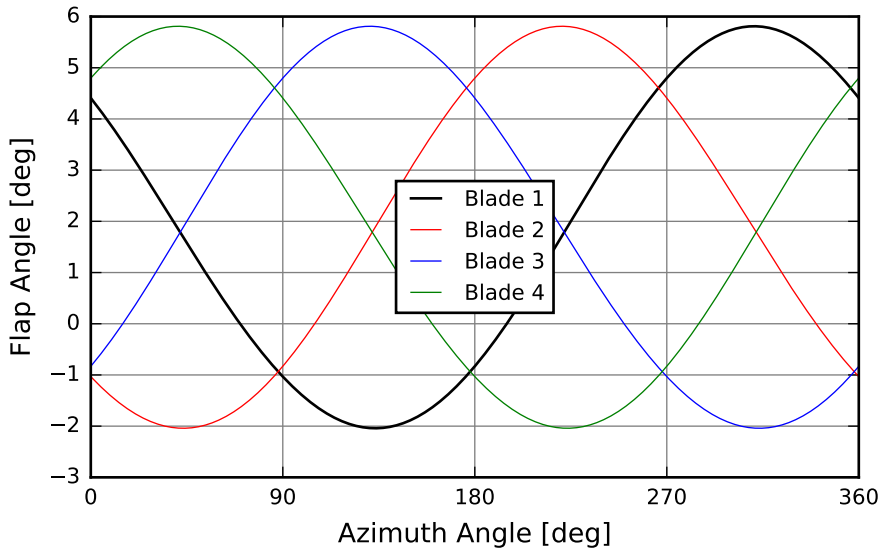


Figure 3.6: Converged flap response.

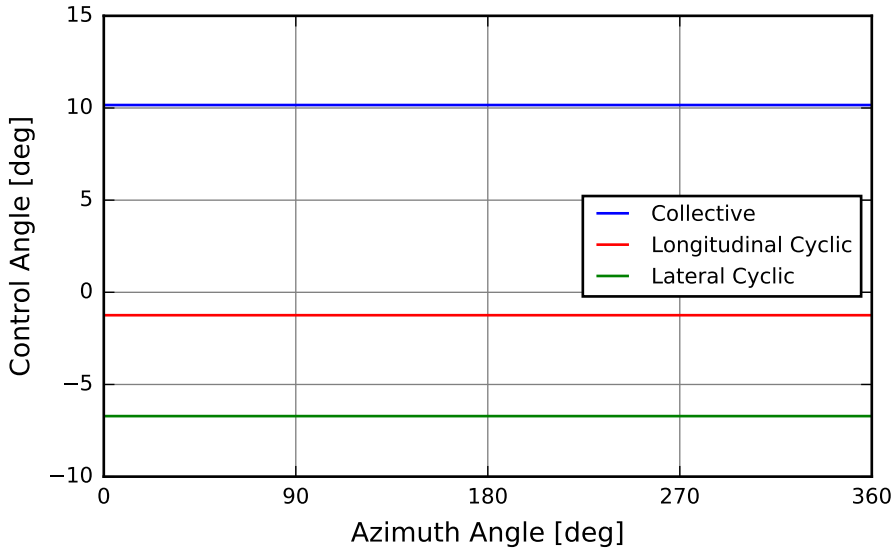


Figure 3.7: Converged control angles.

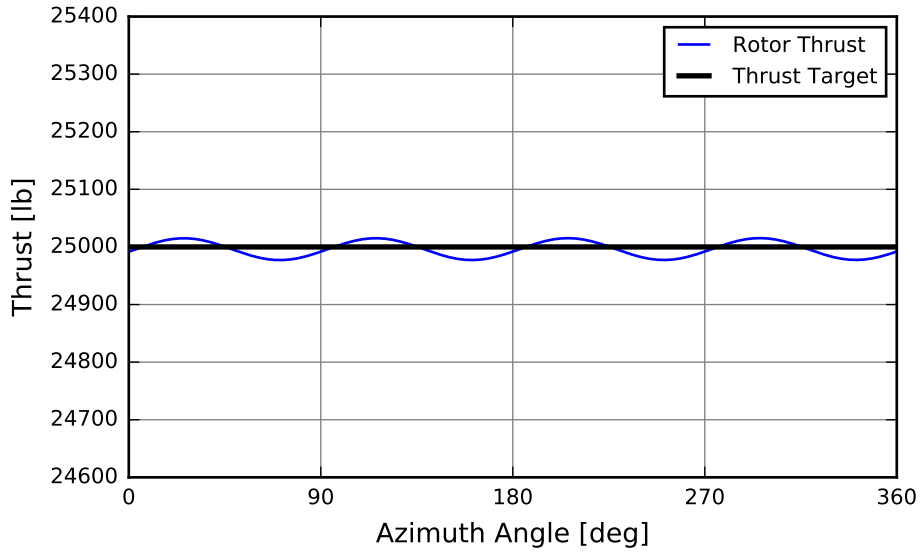


Figure 3.8: Converged thrust.

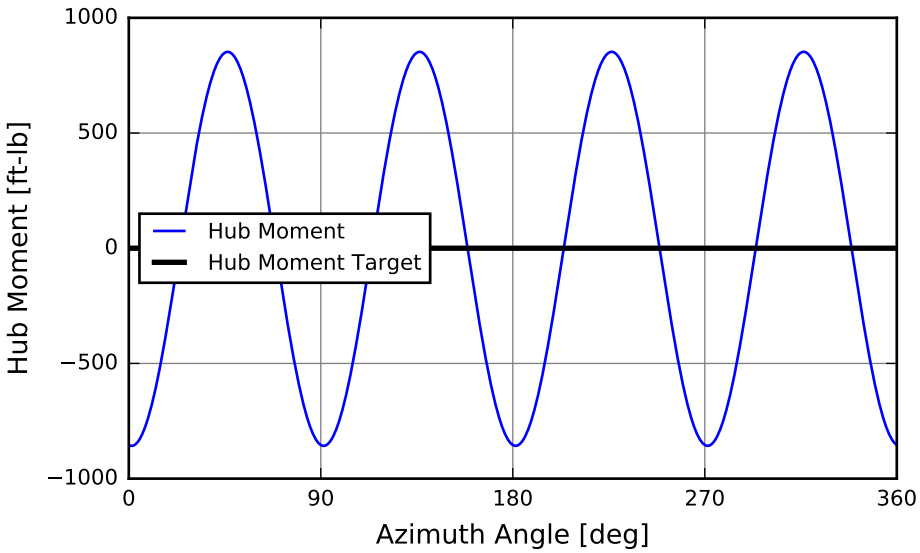


Figure 3.9: Converged hub moment about x-axis.

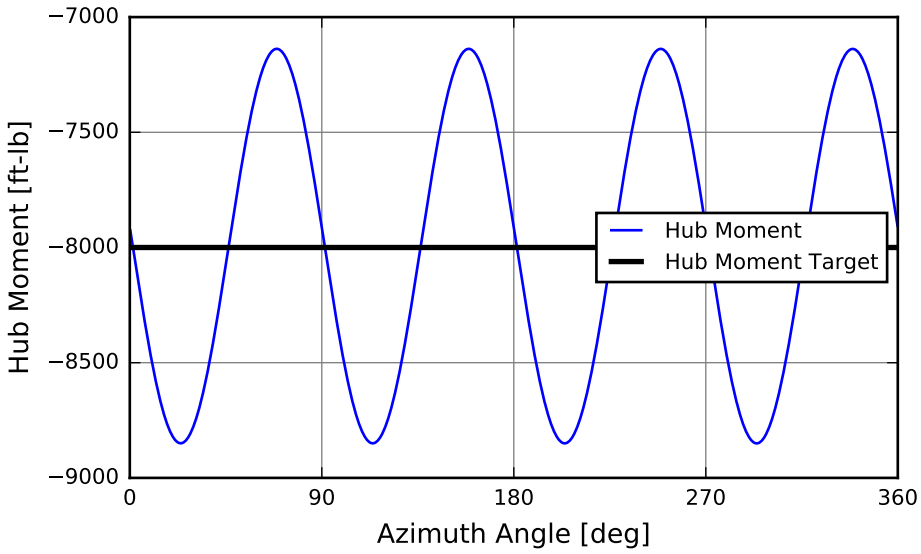


Figure 3.10: Converged hub moment about y-axis.

are a total of 100 iterations shown. The curves are colored by partition (process) number.

Recall that the azimuth angle is not prescribed as a direct function of time, but rather as a function of two state variables whose governing equation corresponds to a dynamical system with a single periodic attractor. The governing equation was designed with the intent of yeilding an azimuth angle that is equivolent to the function $\psi(t) = \psi_0 + \Omega t \mod 2\pi$, where $\psi_0 \in [0, 2\pi]$ is the value of ψ at $t = 0$. Figure 3.11 confirms that the azimuth angle of the rotor model does behave as intended. It can be seen that at the begining of each shooting phase, ψ begins at the azimuth angle associated with the "left" boundary of each respective partition (process), and increases linearly until the azimuth angle of the "right" boundary is reached. Futhermore, the starting azimuth angle returns to the correct value after each correction step, despite the fact that the other state variables are changing (as

will become subsequently evident). This suggests that once a state variable reaches its converged value, it remains at that value for subsequent iterations of PMS, even if the other state variables have not yet converged.

Now, consider the flap response of blade 1 (Figure 3.12). Each partition begins with the initial guess of zero, and the "shooting" step results in a decreasing flap angle. In fact, the results of the first shooting step are nearly identitical on all partitions for this case. The initial prediction of a downward trend is correct on partition 0, but incorrect on all other partitions. By the second iteration, however, the incorrect prediction of a downward trend is corrected, and by the 10th iteration, the solution begins to take on a qualitatively correct shape. By the 20th iteration, the intermediate solutions beging to monotonically approach the final converged solution. The rate at which the solver converges appears to be approximately the same on each partition (process). These same trends can also be seen in the flap response for blades 2, 3, and 4 (Figures 3.13, 3.14, 3.15, respectively). The solution for all four blades appears to progress at approximately the same rate.

Throughout the iteration process, the collective can be seen to monotonically trend towards the converged solution (Figure 3.16), likewise for the Logitunidal cyclic (3.18). The Lateral cyclic, on the other hand, shows an initial trend that progresses in the wrong direction, but is corrected by the 5th iteration (3.17). This behavior of the lateral cyclic mimicks that seen for the direct time marching solution. Likewise, the collective appears to converge the fastest, followed by the lateral cyclic, then the longitudinal cyclic. This convergence trend mimicks the behavior seen for the direct time marching solution. As was seen for the flap response, the rate at

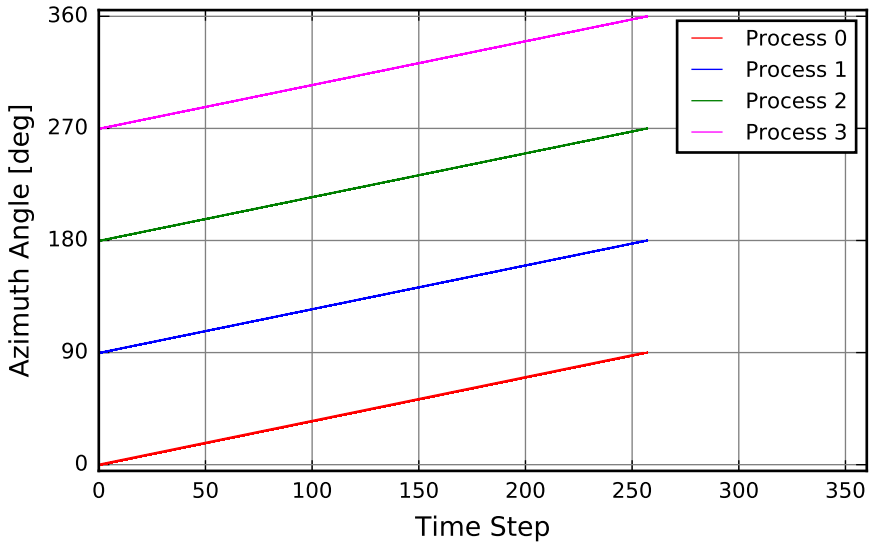


Figure 3.11: Azimuth angle.

which a control parameter converges appears to be indepenent of the partition it is associated with.

The thrust can be seen to steadily converge towards the target value of 25,000 lb (Figure 3.19). The hub moment about the x-axis (Figure 3.20) and y-axis (Figure 3.21) show an initial trend that progresses in the wrong direction, but this trend corrects itself by about the 10th iteration. From that point on, one can see a steady convergence trends towards the mean target values of zero and -8000, respectively.

A comparision between the converged solution obtained using PMS and that obtained using direct time marching is shown in Figures 3.22 through 3.31. In each of these plots, the "reference solution" is the last revolution of the time marching solution (revolution 100). The reference solution is show by a thick black line in all plots. The converged PMS solution is shown by the thin colored curves, where each color corresponds to a state-space partition (process). In all figures, it can be seen

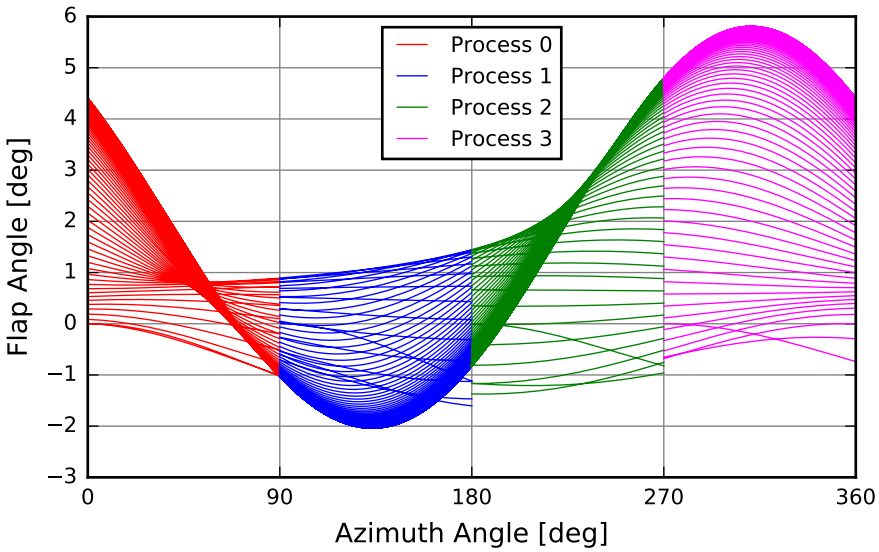


Figure 3.12: Blade 1 flap angle.

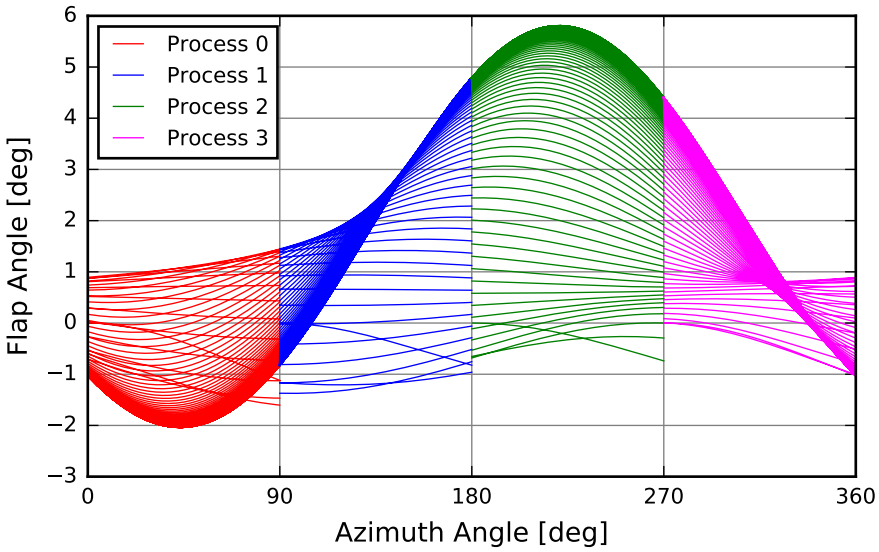


Figure 3.13: Blade 2 flap angle.

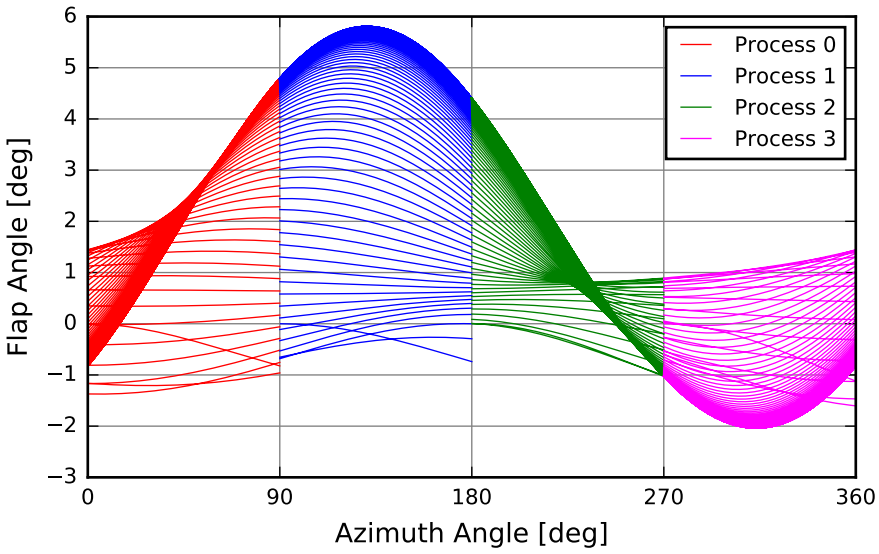


Figure 3.14: Blade 3 flap angle.

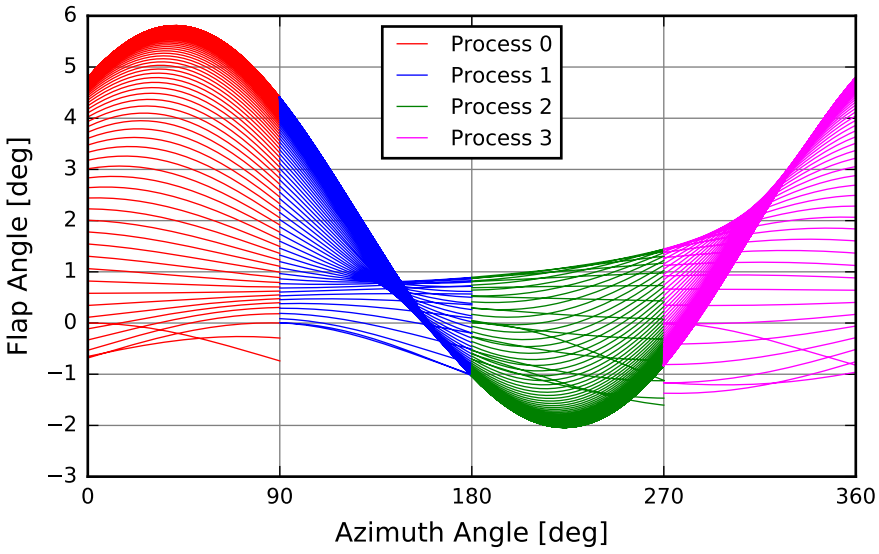


Figure 3.15: Blade 4 flap angle.

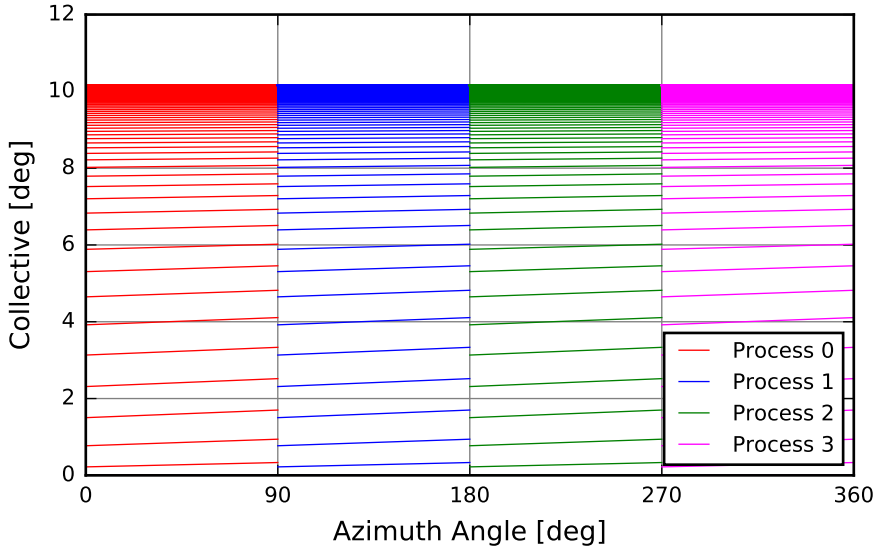


Figure 3.16: Collective.

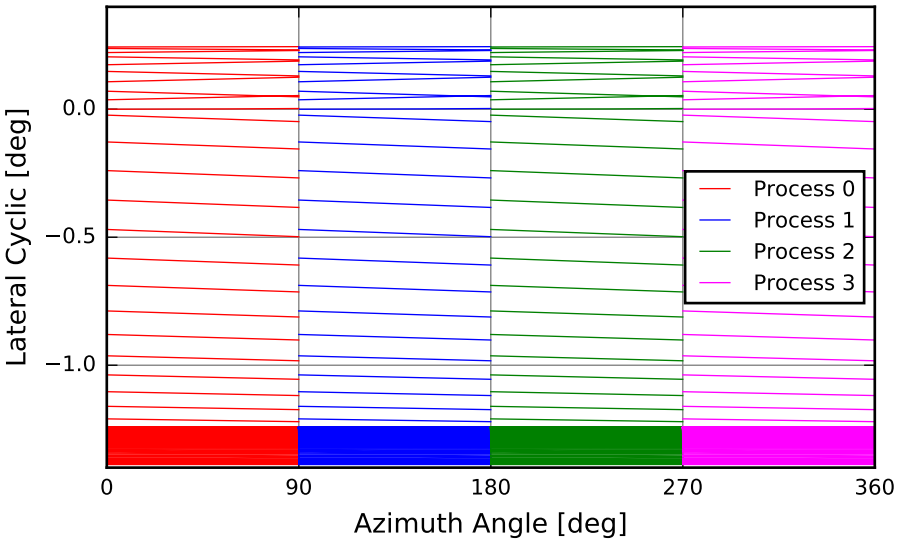


Figure 3.17: Lateral Cyclic.

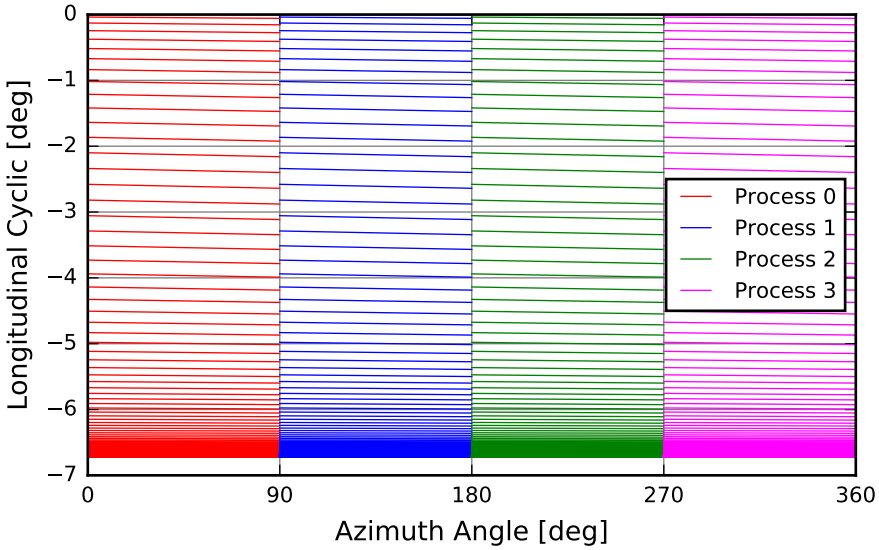


Figure 3.18: Longitudinal Cyclic.

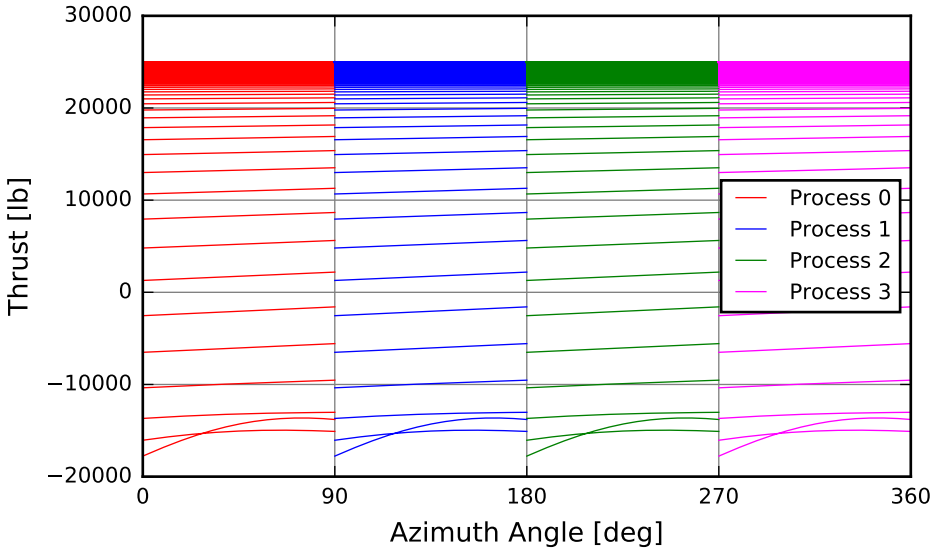


Figure 3.19: Rotor thrust.

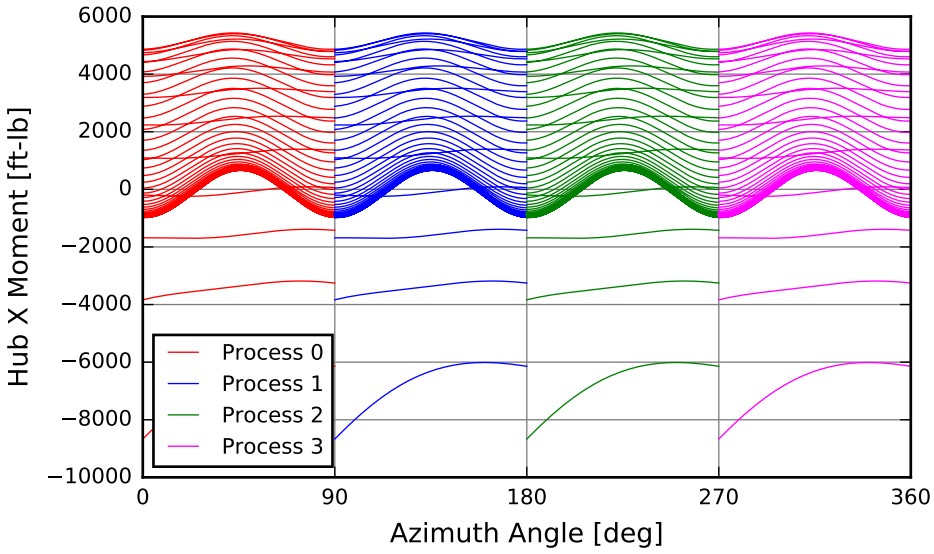


Figure 3.20: Hub moment about x-axis.

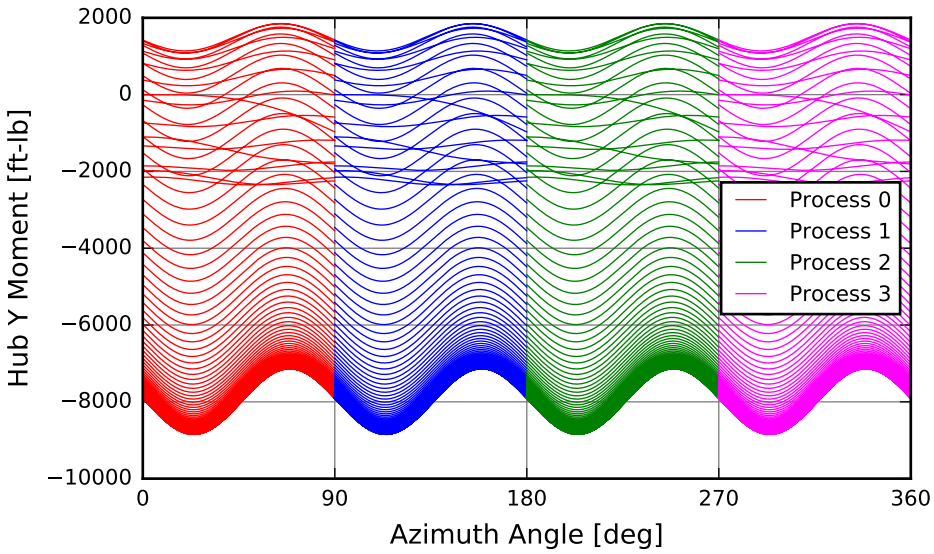


Figure 3.21: Hub moment about y-axis.

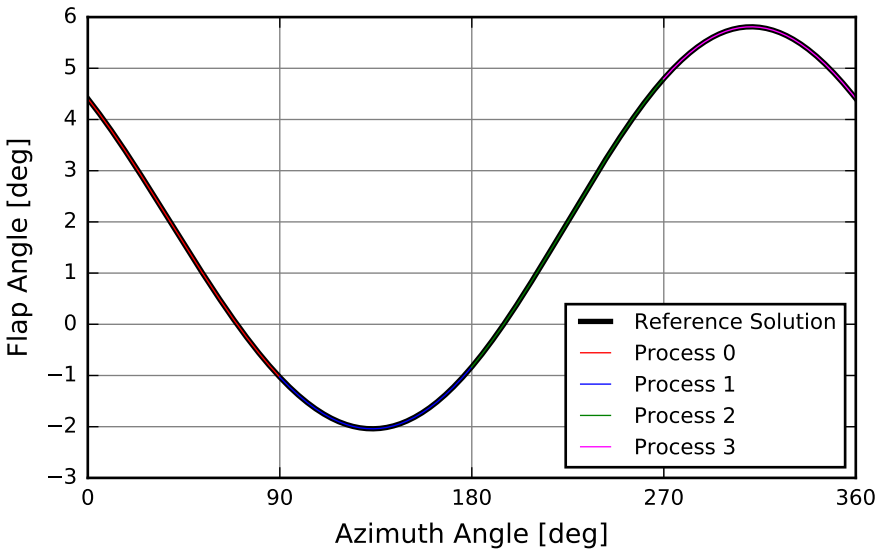


Figure 3.22: Blade 1 flap angle.

that the PMS solution directly overlays the reference solution. This suggest that PMS converges to the same solution as direct time marching, and therefore does not impact the temporal accuracy of the solution.

Figure 3.32 shows the l2-norm of the difference between each periodic solution estimate generated by PMS as a function of iteration. It can be seen that the change in the solution remains nearly constant for the 20 iterations, and then begins to dimish with each iteration.

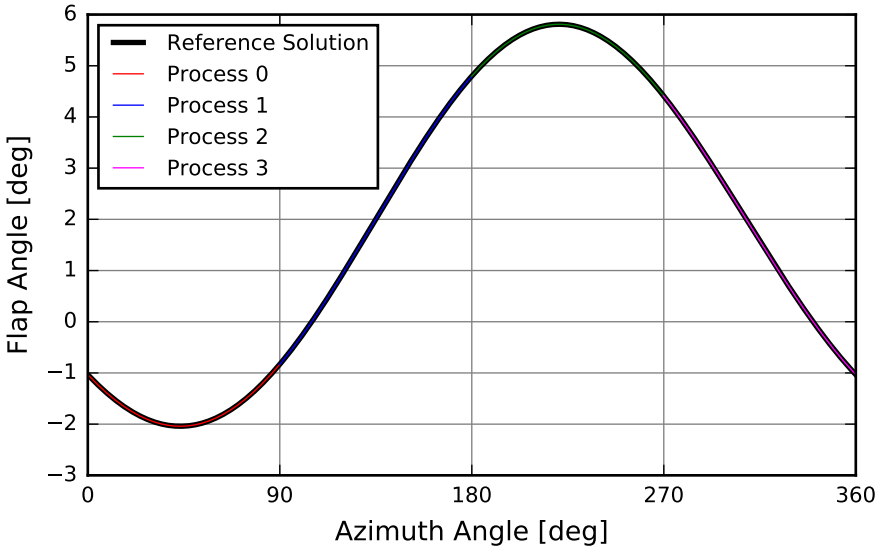


Figure 3.23: Blade 2 flap angle.

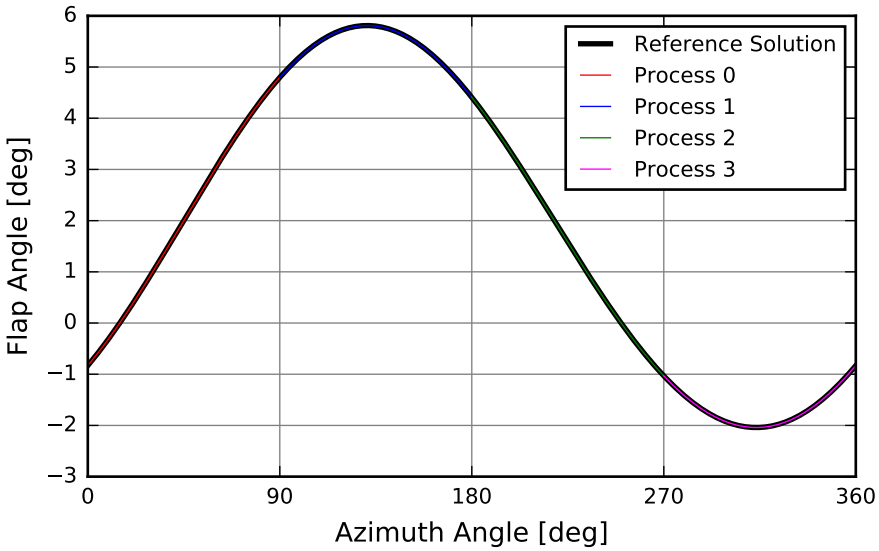


Figure 3.24: Blade 3 flap angle.

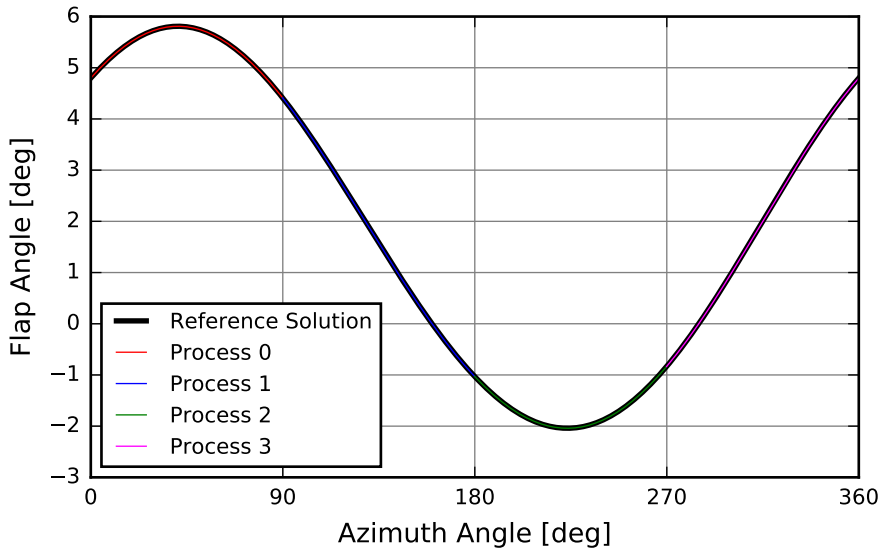


Figure 3.25: Blade 4 flap angle.

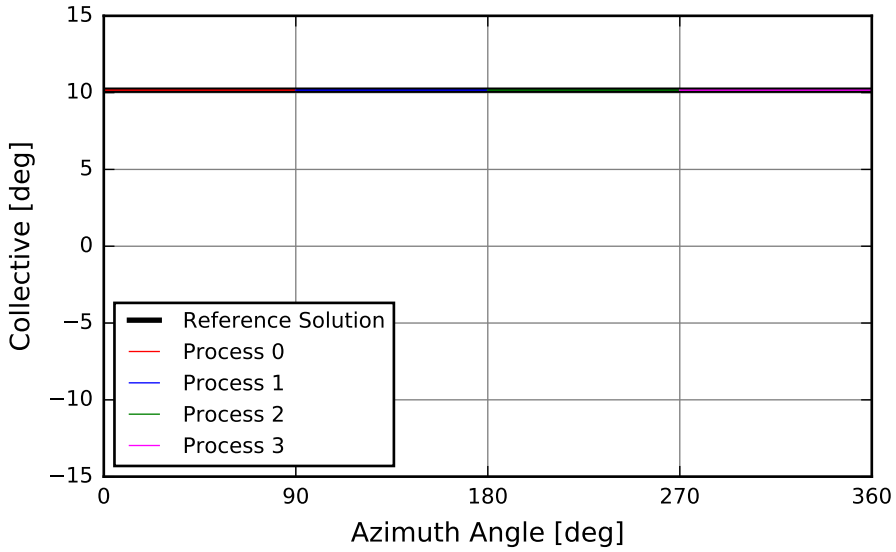


Figure 3.26: Collective.

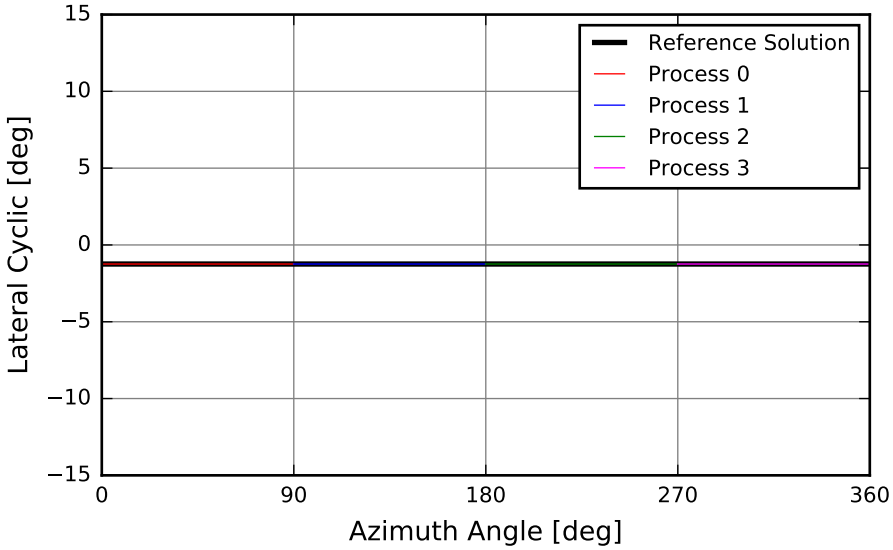


Figure 3.27: Lateral Cyclic.

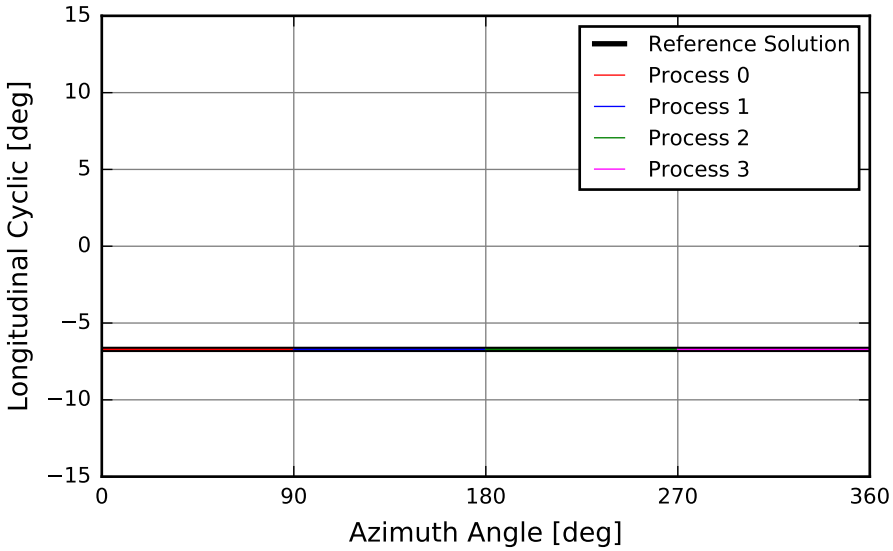


Figure 3.28: Longitudinal Cyclic.

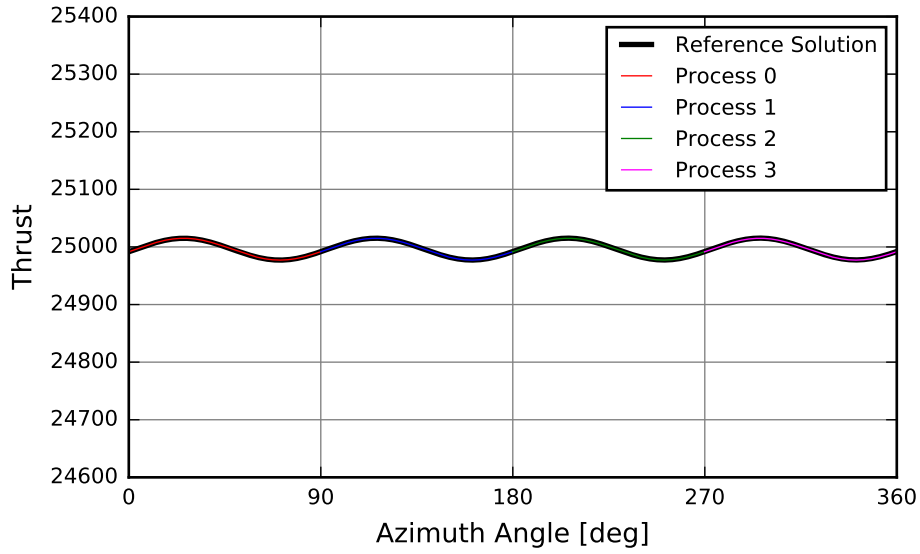


Figure 3.29: Rotor thrust.

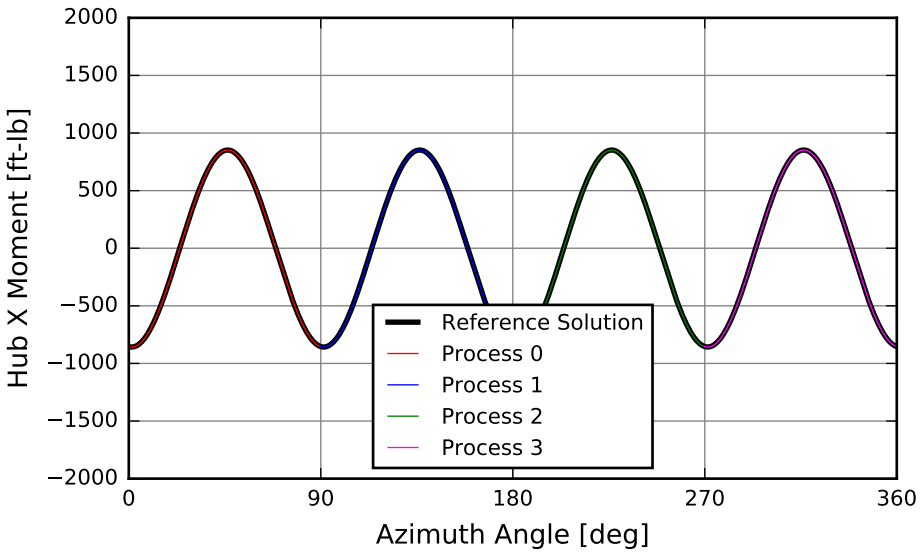


Figure 3.30: Hub moment about x-axis.

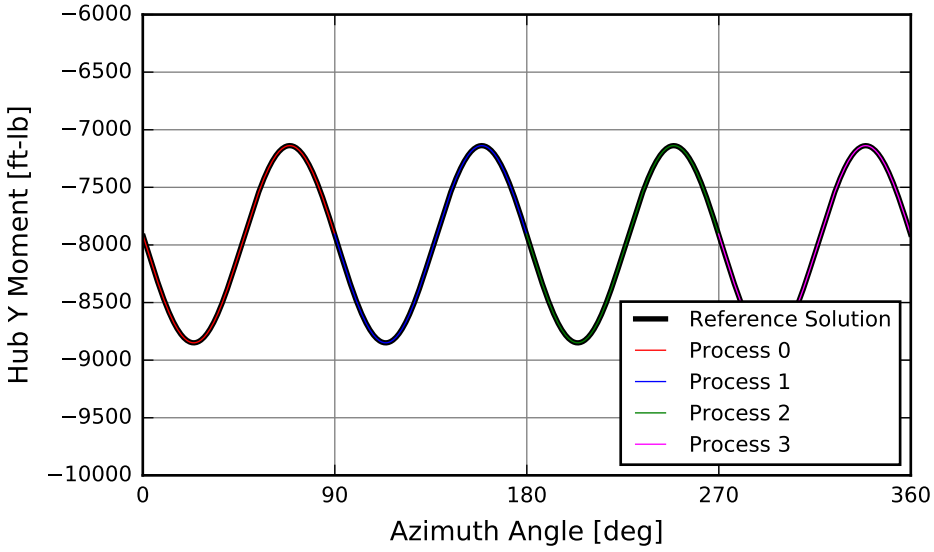


Figure 3.31: Hub moment about y-axis.

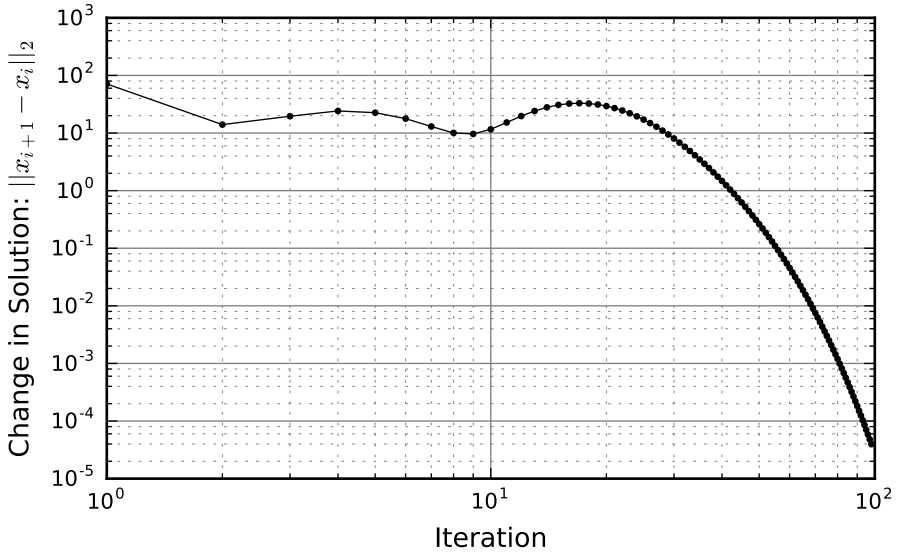


Figure 3.32: Change in solution as a function of iteration.

Algorithm 3 Parallel Symmetric Periodic Multiple Shooting

Require: ϵ, $kmax$, N_b, N_c, INITIALIZE, TIMESTEP, ATPOINCARESECTION, SHIFTRIGHT

```
1: procedure SymBC(X)
2:     n ← (size(X) − N_c)/N_b
3:     Cycle(X[0 : nN_b − 1], n)
4: end procedure

5: procedure Shoot(X, E)
6:     x ← X
7:     repeat
8:         x ← TimeStep(x)
9:     until AtPoincareSection(x)
10:    ShiftRight(x)
11:    if rank = 0 then
12:        SymBC(X)
13:    end if
14:    E ← x − X
15: end procedure

16: procedure Correct(X, E)
17:    Δx ← 0
18:    k ← 0
19:    r ← E
20:    loop
21:        ShiftRight(r)
22:        Δx = Δx + r
23:        k = k + 1
24:        if k > kmax then
25:            Break
26:        end if
27:        r = Sr
28:    end loop
29:    X ← X + Δx
30: end procedure

31: Initialize(X)
32: loop
33:    Shoot(X, E)
34:    if max_{0≤i<p} ‖E[i]‖ < ε then
35:        exit
36:    end if
37:    Correct(X, E)
38: end loop
```

Chapter 4: Application of Periodic Multiple Shooting Method to CFD/CSD Rotor Models

4.1 Theory

{Outline Krylov subspace projection strategy, and its theoretical basis.}

{Address PDE aspects of applying Periodic Multiple Shooting.}

{Discuss Symmetry conditions. In particular, structural states need to be permutated, as well as do the states associated with body fitted meshes (and any other meshes that rotate or follow the rotor blades, but not the states of the stationary background meshes. In the case of body fitted meshes, one may alternatively re-associate each body mesh with the permutated structural states (i.e. reassign each blade mesh to the preceeding blade.}

4.2 Numerical Tests

Appendix A: Dynamical Systems Theory

[TODO: Summarize key concepts and results]

Appendix B: CFD/CSD Analysis Framework

[TODO]

B.1 Rodymol

[TODO]

B.2 OVERTURNS

[TODO]

B.3 TIFAS

[TODO]

B.4 CFD/CSD Verification Results

[TODO]

Bibliography

[1] Morris W. Hirsch and Stephen Smale. *Differential Equations, Dynamical Systems, and Linear Algebra*. Pure and Applied Mathematics. Academic Press, Inc., San Diego, 1974.

[2] Stephen Wiggins. *Introduction to Applied Nonlinear Dynamical Systems and Chaos*, volume 2 of *Texts in Applied Mathematics*. Springer-Verlag, New York, second edition, 2003.

[3] V.I. Arnold. *Ordinary Differential Equations*. MIT Press, Cambridge, 1973.

[4] Walter Gander and Gene H Golub. Cyclic reduction–history and applications. *Scientific computing (Hong Kong, 1997)*, pages 73–85, 1997.

[5] David A Peters and Dinesh Barwey. A general theory of rotorcraft trim. *Mathematical Problems in Engineering*, 2(1):1–34, 1996.

[6] DA Peters and M Chouchane. Effect of dynamic stall on helicopter trim and flap-lag response. *Journal of Fluids and Structures*, 1(3):299–318, 1987.

[7] D Chandra Sekhar and R Ganguli. Fractal boundaries of basin of attraction of newton–raphson method in helicopter trim. *Computers & Mathematics with Applications*, 60(10):2834–2858, 2010.

[8] Bogdan I Epureanu and Henry S Greenside. Fractal basins of attraction associated with a damped newton's method. *SIAM review*, pages 102–109, 1998.

[9] Mako Haruta. Newton's method on the complex exponential function. *Transactions of the American Mathematical Society*, 351(6):2499–2513, 1999.

[10] Moonja Jeong, Gi Ok Kim, and Seong-A Kim. Dynamics of newton's method for solving some equations. *Computers and Graphics*, 26(2):271 – 279, 2002.

[11] Saipraneeth Gouravaraju and Ranjan Ganguli. Estimating the hausdorff–besicovitch dimension of boundary of basin of attraction in helicopter trim. *Applied Mathematics and Computation*, 218(21):10435–10442, 2012.

[12] Peter Kunkel. *Differential-algebraic equations: analysis and numerical solution*. European Mathematical Society, 2006.

[13] Eduard Reithmeier. *Periodic Solutions of Nonlinear Dynamical Systems*. Number 1483 in Lecture Notes in Mathematics. Springer-Verlag, Berlin, 1991.

[14] Peretz P. Friedmann. Numerical methods for determining the stability and response of periodic systems with applications to helicopter rotor dynamics and aeroelasticity. *Computers and Mathematics with Applications*, 12(1):131 – 148, 1986.

[15] David A Peters and Amir P Izadpanah. Helicopter trim by periodic shooting with newton-raphson iteration. In *American Helicopter Society, Annual Forum, 37 th, New Orleans, LA, Proceedings*, 1981.

[16] S Subramanian, GH Gaonkar, J Nagabhushanam, and RM Nakadi. Parallel computing concepts and methods for floquet analysis of helicopter trim and stability. *Journal of the American Helicopter Society*, 41(4):370–382, 1996.

[17] James SG McVicar and Roy Bradley. Robust and efficient trimming algorithm for application to advanced mathematical models of rotorcraft. *Journal of aircraft*, 32(2):439–442, 1995.

[18] David A Peters. Fast floquet theory and trim for multi-bladed rotorcraft. *Journal of the American Helicopter Society*, 39(4):82–89, 1994.

[19] Shyam Subramanian, S Venkataratnam, and GH Gaonkar. Parallel fast-floquet analysis of trim and stability for large helicopter models. *Mathematical and computer modelling*, 33(10):1155–1176, 2001.

[20] Ali H. Nayfeh and Dean T. Mook. *Nonlinear Oscillations*. Wiley-VCH, 2004.

[21] P. Shahrzad and M. Mahzoon. Limit cycle flutter of airfoils in steady and unsteady flows. *Journal of Sound and Vibration*, 256(2):213 – 225, 2002.

[22] Liping Liu and Earl H. Dowell. Harmonic balance approach for an airfoil with a freeplay control surface. *AIAA Journal*, 43(4), 2005.

[23] Y. K. Cheung and S. L. Lau. Incremental time-space finite strip method for non-linear structural vibrations. *Earthquake Engineering and Structural Dynamics*, 10:239 – 253, 1982.

[24] Y. K. Cheung and S. H. Chen. Application of the incremental harmonic balance method to cubic non-linearity systems. *Journal of Sound and Vibration*, 140(2):273 – 286, 1990.

[25] A. Raghothama and S. Narayanan. Non-linear dynamics of a two-dimensional airfoil by incremental harmonic balance method. *Journal of Sound and Vibration*, 226(3):493–517, 1999.

[26] Kenneth C. Hall, Jeffery P. Thomas, and W.S. Clark. Computation of unsteady nonlinear flows in cascades using harmonic balance technique. *AIAA Journal*, 40(5), 2002.

[27] Jeffrey P. Thomas, Earl H. Dowell, and Kenneth C. Hall. Modeling viscous transonic limit-cycle oscillation behavior using a harmonic balance approach. *Journal of Aircraft*, 41(6), 2004.

[28] L. Liu, J.P. Thomas, E.H. Dowell, P. Attar, and K.C. Hall. A comparison of classical and high dimensional harmonic balance approaches for a duffing oscillator. *Journal of Computational Physics*, 215:298 – 320, 2006.

[29] Liping Liu, Earl H. Dowel, and Kenneth C. Hall. A novel harmonic balance analysis for the van der pol oscillator. *International Journal of Non-Linear Mechanics*, 42:2–12, 2007.

[30] Kivanc Ekici, Kenneth C. Hall, and Earl H. Dowell. Computationally fast harmonic balance methods for unsteady aerodynamic predictions of helicopter rotors. *Journal of Computational Physics*, 227:6206 – 6225, 2008.

[31] T-K Hsu and David A Peters. Coupled rotor/airframe vibration analysis by a combined harmonic-balance, impedance-matching method. *Journal of the American Helicopter Society*, 27(1):25–34, 1982.

[32] M Borri. Helicopter rotor dynamics by finite element time approximation. *Computers & mathematics with applications*, 12(1):149–160, 1986.

[33] David A Peters, BS Kim, and H-S Chen. Calculation of trim settings for a helicopter rotor by an optimized automatic controller. *Journal of Guidance, Control, and Dynamics*, 7(1):85–91, 1984.

[34] David A Peters, Mnaouar ChouChane, and Mark Fulton. Helicopter trim with flap-lag-torsion and stall by an optimized controller. *Journal of Guidance, Control, and Dynamics*, 13(5):824–834, 1990.

[35] Russell Enns and Jennie Si. Helicopter trimming and tracking control using direct neural dynamic programming. *IEEE Transactions on Neural Networks*, 14(4):929–939, 2003.

[36] Luca Riviello. *Rotorcraft trim by a neural model-predictive auto-pilot.* PhD thesis, Georgia Institute of Technology, 2005.

[37] Carlo L Bottasso and Luca Riviello. Trim of rotorcraft multibody models using a neural-augmented model-predictive auto-pilot. *Multibody System Dynamics*, 18(3):299–321, 2007.

[38] Frederick D Kim, Roberto Celi, and Mark B Tischler. Forward flight trim and frequency response validation of a helicopter simulation model. *Journal of Aircraft*, 30(6):854–863, 1993.

[39] David A Peters and Si-Hao Li. A combined periodic-shooting, auto-pilot technique for rotorcraft analysis. In *Computational Mechanics' 95*, pages 654–659. Springer, 1995.

[40] Troy Shank. *Optimal aeroelastic trim for rotorcraft with constrained, non-unique trim solutions*. PhD thesis, Georgia Institute of Technology, 2008.

[41] Chee Tung, Francis X. Caradonna, and Wayne R. Johnson. The prediction of transonic flows on an advancing rotor. *Journal of American Helicopter Society*, 31(3):4 – 9(6), 1986.

[42] Mark Potsdam, Hyeonsoo Yeo, and Wayne Johnson. Rotor airloads prediction using loose aerodynamic/structural coupling. *Journal of Aircraft*, 43(3), May-June 2006.

[43] Anubhav Datta, Mark Nixon, and Inderjit Chopra. Review of rotor loads prediction with the emergence of rotorcraft cfd. *Journal of the American Helicopter Society*, 52(4):287–317, 2007.

[44] Jayanarayanan Sitaraman, Mark Potsdam, Buvaneswari Jayaraman, Anubhav Datta, Wissink Andrew, Dimitri Mavriplis, and Hossein Saberi. Rotor loads prediction using helios: a multi-solver framework for rotorcraft cfd/csd analysis. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 1123, 2011.