AN EFFICIENT FLUID-RIGID BODY INTERACTION SIMULATION OF A BIOMIMETIC MICRO AERIAL VEHICLE

By

DAVID JOSEPH TALARICO

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Mechanical and Aerospace Engineering

Written under the direction of

Mitsunori Denda and Aaron Mazzeo

And approved by

____

____

____

____

New Brunswick, New Jersey

May, 2019

ABSTRACT OF THE DISSERTATION

AN EFFICIENT FLUID-RIGID BODY INTERACTION SIMULATION OF A BIOMIMETIC MICRO AERIAL VEHICLE

by DAVID JOSEPH TALARICO

Dissertation Directors:

Mitsunori Denda and Aaron Mazzeo

Advancements in aerospace technologies which rely on unsteady fluid dynamics are being hindered by a lack of easy to use, computationally efficient unsteady CFD software. Flapping in nature is ubiquitous, yet modern day micro air vehicles (MAVs) based on flapping are in their infancy. The most successful MAVs to date

are much less maneuverable and efficient than their natural counterparts, partly due to the fact that they are based on the relatively simple aerodynamics of propeller blades. Similarly, mainstream wind turbines are based on rotary blades and relatively simple aerodynamics. In fact, flapping wing energy harvesters have been increasingly investigated as a possible alternative to traditional wind and tidal turbines after several studies highlighted their unique capabilities and exceptional efficiencies.

Flapping wing energy harvesting has been shown in recent years to be reaching efficiency levels "comparable to the best performances achievable with modern rotor blade turbines" [1]. Simultaneously, advancements in experimental and computational abilities are bringing a comprehensive understanding of the underlying mechanisms of flapping wing insect flight within reach. These mechanisms may hold the key to highly maneuverable and resilient MAVs [2], and some of them have been shown to enhance the performance of flapping foil power generators as well [3].

The major reason for the scarcity of flapping devices is the difficulty involved in the design of such devices. Existing CFD platforms are capable of handling unsteady flapping, but the time, money, and expertise required to run even a basic flapping simulation makes design iteration and optimization prohibitively expensive for the average researcher. For the design of a MAV capable of the complex maneuvers observed by natural flyers, this lack of computational efficiency makes the successful implementation of a viable design nearly impossible.

However, by utilizing a novel unsteady vortex method which has been designed specifically to handle the highly unsteady flapping wing problem, it has been shown [4] that the time to compute a solution is reduced by a factor of 20, and the level of skill to operate the software is reduced so much that an undergraduate engineering student can easily produce accurate results.

Despite the success of the original vortex method from [4], especially for a small number of flapping cycles, the solution deteriorates as the number of flapping cycles increases due to the inherent lack of viscosity in the vortex method. This would make it challenging to couple the fluid solver to a rigid body solver, as the increasing moment would cause the MAV to tumble. In addition, the original vortex method from [4] does not utilize parallel processing to increase the computational speed and wastes a large amount of computational time in the far field wake for simulations involving multiple flapping cycles. Finally, the original method assumed the motion of the MAV to be fully prescribed, whereas a more valuable tool for MAV designers would allow for the motion of the MAV body to be predicted concurrently with the fluid forces given a specified wing motion.

It is therefore the goal of the present study is to create a faster, suitably accurate fluid-rigid body interaction simulation for flapping wing MAV designers. This is accomplished by further improving the computational efficiency of the solver and by adding the capability for the simulation to solve for the flight trajectory and aerodynamic forces of a flapping wing MAV concurrently given only the flapping kinematics. This simulation thereby allows for testing virtual flight stability, maneuver flapping sequences, and performance testing.

First, the code will be shown to run orders of magnitude faster by being modified to allow the GPU to compute vortex velocity contributions in a massively parallel configuration.

In addition, a remedy which models the effect of viscosity is introduced into the original vortex method. The new approach proposed herein lumps far field vortices to simulate viscosity-induced vortex decay, which will be shown to improve the accuracy of the solution while maintaining the pitching moment amplitude. This is especially important for simulations involving many flapping cycles, which is the case when predicting the flight path of an MAV. In addition to improving the accuracy of the solution, the new method greatly reduces the computation time for simulations involving many flapping cycles. Several different incident flow velocity angles were tested and the moment amplitude is shown not to increase for all cases.

Moreover, a novel fluid-rigid body interaction simulation is shown to leverage the improvements to the fluid model to allow for the equations of motion of a two-body flapping wing flyer to be solved. This new fluid-rigid body solver is then utilized to support the hypothesis that the position of a flapping wing insect's abdomen is carefully adjusted in order to balance the pitching moment created by aerodynamic forces generated from flapping. Additionally, a basic control feedback loop is introduced to simulate an MAV's on-board active flight stabilization control system. This simulated control system is shown to stabilize the MAV's main body, thereby creating a stable platform for mounting sensors, such as a camera. Finally, the solution of the original vortex method from [4] and the solution of the present method are compared to published data from a full Navier Stokes simulation [5] and show good agreement.

Acknowledgements

This dissertation is the result of many people's efforts and support. Thank you to my co-advisor Dr. Mitsunori Denda for working with me to apply to fellowships, for guiding me in the writing process, and for his selfless and tireless support in securing my teaching assistantship year after year. He has shown me how to push the boundaries of my field of research while remaining securely grounded in scientific law. Thank you to my co-advisor Dr. Aaron Mazzeo for all of the invaluable mentoring and writing advice. I feel extremely blessed to have had the honor of being mentored by two advisors who have taken countless hours out of their busy schedules to guide and support my research efforts. I truly appreciate the kindness and patience you both have shown me time and again. Thank you to my committee member Dr. Lisa Grega for preparing me for graduate school by mentoring my first research project during my undergraduate studies, and thank you for taking the long trip from The College of New Jersey to Rutgers each time we met to discuss my research. Thank you to my committee member Dr. Onur Bilgen for taking the time to be a part of my committee and for the help and guidance with my research.

I would like to thank all of the exceptional faculty and staff members of the Mechanical and Aerospace Engineering Department for making my time at Rutgers an enjoyable and exceptional learning experience.

Finally, I would like to thank my family and friends for all of their help and support. Thank you to my parents for their endless support and encouragement, and thank you to my wife, Karen, for always being there to lift my spirits when I needed it most.

Table of Contents

List of Figures

Chapter 1 | Introduction

1.1 Motivation

Currently, the most popular CFD software packages do an excellent job solving and optimizing steady or quasi-steady problems, but solving highly unsteady flow problems with them is so difficult and time consuming that only a few highly qualified individuals with access to expensive computational resources are capable of doing so. This is due in part to the overly generalized, steady or quasi-steady foundation which traditional Navier Stokes solvers are built upon. One example of an exciting field which could greatly benefit from an easy to use unsteady solver is the field of flapping wing aerodynamics.

DARPA defined micro air vehicles as aircraft which are smaller than 15 cm in size. They began to fund research in 2000 with the goal of creating ultra-small surveillance planes for military use. One of the first designs was created by a company called AeroVironment. This MAV is known as the Black Widow, and comprises a fixed wing design with an onboard camera that can operate for over a half an hour on one charge.

The problem that designers ran into was that very small, low speed aircraft such as these were very difficult to stabilize in high winds. However, biological flyers exhibit excellent control of their position under dynamic conditions. Insects, in particular, accomplish this with responsive flexible wings, a full array of motion and pressure sensors, coordinated body motions, and exceptional vision [6]. The highly complex aerodynamics that result from the coordination of these features gives rise to interconnected nonlinearities between the insects control systems, wing deformation, and the surrounding fluid's dynamics.

They turned to fluid dynamicists and biologists for an alternative approach inspired by nature. It is thought that flapping wings of insects allow for much more stability when faced with a cross wind, as insects have been observed to have exceptional control and stability in such situations. This, coupled with the desire for insect-sized flying surveillance vehicles, prompted many researchers to turn their attention to the problem.

Another example of a field which could benefit greatly from an easy to use unsteady fluids solver for flapping wings is flapping wing energy harvesting. There is an ever-expanding demand for energy on our planet today. Factors include the exponentially growing population of over 7 billion people and the industrialization of many developing countries. Besides being the world's largest market (~$6 trillion per year), energy is of great concern to humanity because the current prevailing methods of producing it are amongst the most ecologically destructive human activities.

A century ago, electrical power was just being introduced to the general population. Much like computers in the 1970's, electricity at the time likely seemed like a curious novelty for scientists to use. Once people began to realize the potential uses for electricity, many electrical power plants began to emerge. Most of these power plants used coal as fuel, and for many decades systems and standards were developed around the idea that the byproducts of combustion would be so diluted in the vast atmosphere that humans would not be able to produce any noticeable effect on the environment. Now, in the 21st century, we have come to terms with the fact that not only are the products of combustion harmful to the environment, humans are producing them at such high and accelerating rates that we are beginning to see their effects in the form of global climate change.

Disruptive renewable energy technology has the potential to free us from our dependence on these deleterious energy sources and prevent the mass destruction which would follow from climate change.

There are many forms of renewable energy available today. Amongst them, the closest candidate for market disruption is wind power, which as of 2016 has a lower cost of energy than a coal fired plant. Hydroelectric dams are also outcompeting coal, but almost every feasible geological site for one is already being utilized. Although not technically renewable, nuclear power is also slightly cheaper than coal, but it comes with hosts of its own environmental issues.

The cost-competitive wind power system referred to earlier is typically a "wind farm" consisting several dozen or even hundreds of horizontal-axis wind turbines (HAWT). While it is the most cost competitive in most cases, the HAWT has many drawbacks.

First, the blades are typically over 100 feet long, so even at moderate rotational speeds the tips reach tangential velocities approaching the speed of sound. This creates a loud humming noise which can be irritating and even harmful to local residents. This, combined with objections many have to their aesthetic appeal, forces wind farms to be constructed far from populated areas. As a result, long, high resistance power lines must be used thereby greatly reducing the overall system efficiency.

Second, to maintain uniform stress along the blade, the shape of the blade must twist and taper simultaneously along its length. The twist is necessary to maintain a uniform effective angle of attack along the length of the blade (due to the linearly varying tangential velocity). The taper is required to reduce the aerodynamic forces along the blade as these forces typically scale with the square of the local tangential velocity (which varies linearly along the blade). This complex geometry is achieved through the hand-layering of fiberglass. This imprecise manual manufacturing technique allows for small bubbles to form in the blade, which can cause high stress concentration and catastrophic blade failure up to 20% of the time [7].

Even with all of these drawbacks, the wind energy industry is the fastest growing of all of the renewables [8]. With plenty of room for improvement in wind harvesting designs and the encouraging growth of the industry, there is a unique opportunity for engineers to innovate and create novel wind energy harvesting systems.

Flapping wing energy harvesting systems, such as the one presented here, have many advantages over traditional rotary designs. For one, the span (which is typically the longest dimension of the wing/blade) is not directed radially or subject to rotational motion. This prevents the common noise pollution problem found with HAWT designs.

Next, for the same reason, the wing does not need to twist along its length, opening up the possibility of using the same manufacturing techniques used for commercial aircraft to construct the wing. Not only would this dramatically reduce the cost, it could virtually eliminate the high rate of catastrophic blade failure in HAWT designs. (e.g. When was the last time you heard of a 747's wing falling off from stress concentration?)

In addition, flapping wing systems can offer the flexibility to operate in places rotary turbines can't. A flapping wing system developed by Pulse Tidal is able to intercept the same cross-sectional area (and hence the same amount of available power) as a rotary design in shallower water.

Last and most importantly, flapping wing energy harvesters have been shown to be as efficient if not more efficient than HAWT's. For all of these reasons, the number of research papers in the field of flapping wing energy harvesters has grown almost exponentially over the past decade.

These are just two examples of applications where an easy to use, computationally efficient unsteady CFD solver for the flapping wing problem could benefit society. In order to develop such software, the present work began by building on groundbreaking research [4], and by conducting a thorough literature review to gain a comprehensive understanding of the physical phenomena behind flapping wing flight.

1.2 Flapping Wing Fundamentals

The flapping wing problem has been studied extensively over the past two decades due to the rise in available computational power to the average researcher. The following section highlights the key findings from these studies which pertain to the underlying physical phenomena responsible for efficient flapping wing designs.

1.2.1 The Limited Role of the Reynolds Number

The Reynolds number is a dimensionless number that compares inertial effects to viscous effects:

$$Re = \frac{UL}{\nu}, \tag{1.1}$$

where U is the velocity of the flow, L is a length scale, and ν is the kinematic viscosity of the fluid. A higher Reynolds number means the flow is dominated by inertia while a lower Reynolds number flow is dominated by viscosity. The vast majority of fluid dynamics problems rely on the Reynolds number for dynamic similarity. The study of flapping wings appears to be unusual in that the Reynolds does not play a significant role.

> "Flapping foil systems where the [leading edge vortex] is dominant are likely to be much less sensitive to Reynolds number effects than airfoils/hydrofoils in steady state flows, or other bluff bodies" – Young [9]

1.2.2 The Leading Edge Vortex

By far the most important mechanism of force production in efficient flapping-wing energy flight is the leading edge vortex. First suggested as a mechanism of enhanced force production for insect flight by Ellington [10], the leading edge vortex is also a phenomenon known by pilots of fixed wing aircraft. It occurs during rapid pitching of a wing above the stall angle. Once the wing is pitched upward, a large vortex forms on the leading edge of the wing, generating a high lift force momentarily before detaching from the wing's surface and convecting downstream. The wake evolves (in time) into a highly turbulent state and the wing is said to be "stalled" (i.e. loses lift and gains drag).

Ellington [10] purported that unlike in the fixed wing case, insects reduce the angle of attack of their wings just as the leading edge vortex detaches, allowing them to reattach the flow before the wing stalls. This hypothesis has been validated by numerous sources and has been shown to contribute the majority of the desirable unsteady aerodynamic force to the wing.

Like insect wings, efficient flapping-wing MAVs will rely heavily on the leading edge vortex to generate the desired lift. As the vortex grows and sheds, the angle of attack is reduced such that the wing never stalls. Therefore, the amount of time the vortex remains attached to the leading edge will determine the frequency of oscillation. This is why the reduced frequency is such an important dimensionless parameter in the flapping wing problem.

1.2.3 The Reduced Frequency: A Measure of Synchronicity

The reduced frequency is defined as follows:

$$f^* = \frac{fc}{U}, \tag{1.2}$$

where f is the frequency of oscillation of the airfoil, c is the chord length, and U is the oncoming flow velocity. Since the time required for a fluid particle to travel the length of the chord is $t_c = \frac{c}{U}$, and the time to complete one oscillation of the wing is $t_{osc} = \frac{1}{f}$, we have:

$$f^* = \frac{fc}{U} = \frac{t_c}{t_{osc}} = \frac{1}{r}, \tag{1.3}$$

where r is the number of chord lengths that a fluid particle in the free stream travels during one complete oscillation of the wing. For example, if the reduced frequency is 0.1, the fluid in the free stream travels 10 chord lengths in one flapping cycle. Therefore, the reduced frequency is simply the inverse of the number of chord lengths traveled by a fluid particle in the free stream during one flapping cycle.

The implications of the cited importance of the reduced frequency are central to gaining a comprehensive understanding of the flapping wing problem. To gain a qualitative understanding of why the reduced frequency is significant, let us fist consider the step response of a wing whose pivot point is fixed. Imagine the wing pitching rapidly from a zero angle of attack to 50 degrees (Figure 1.1). The flow at the leading edge separates, and a leading edge vortex begins to form. What happens next is the key to understanding the underlying principles involved with effective flapping wing systems. The vortex is a low-pressure region, so all of the fluid at the boundary of the vortex will have a pressure force directed towards the center of the vortex.

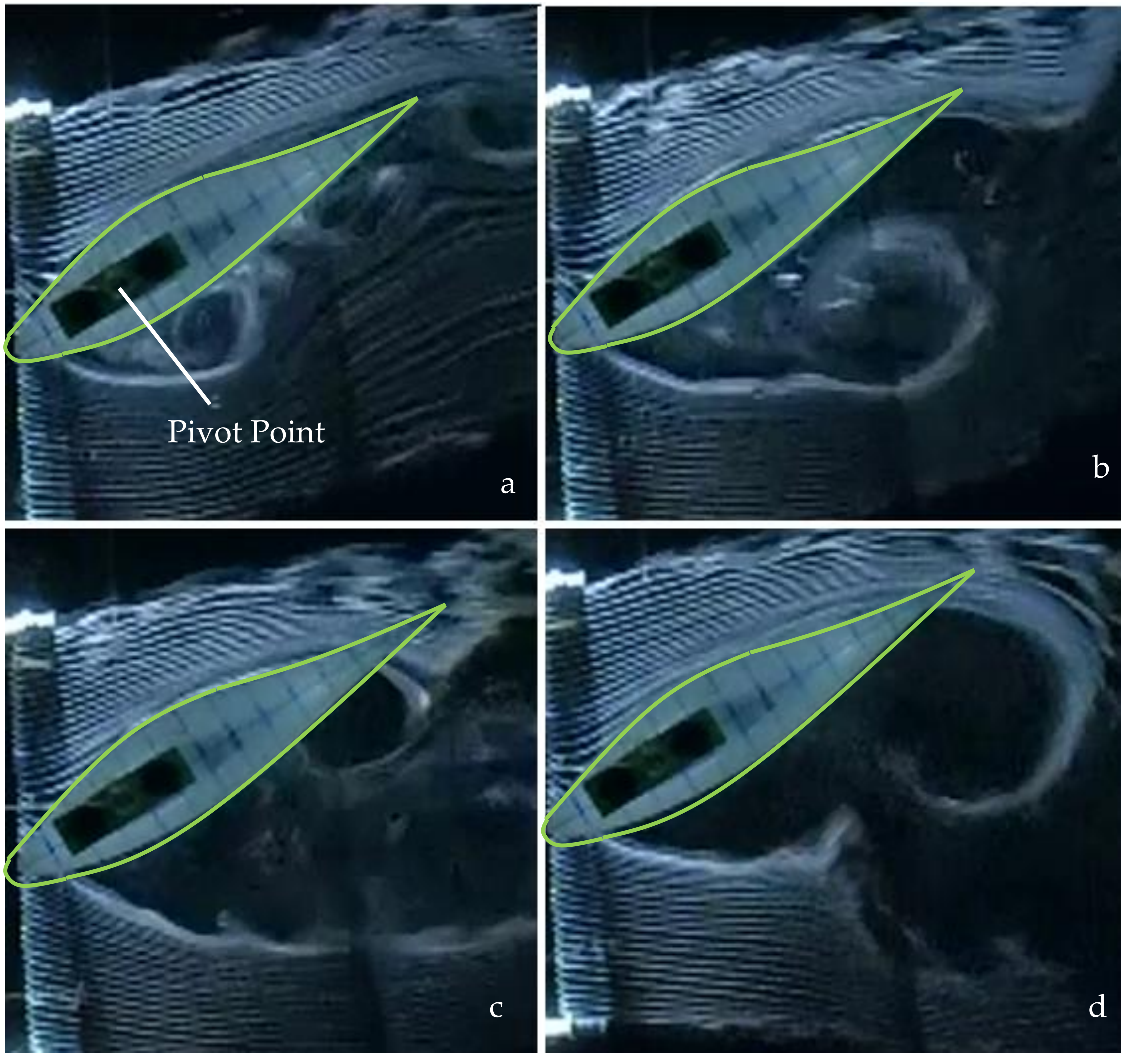


Figure 1.1 - Dynamic stall (wing outlined in green)[11]

First, let us focus on the fluid that was diverted downward by the leading edge at the boundary of the vortex in Figure 1.1a. The pressure force that acts on this fluid is towards the center of the vortex, but since it has momentum in the direction of the free stream, it follows a spiral path in towards the center of the vortex and becomes “entrained”. This feeds the vortex and as it grows (Figure 1.1b), and it becomes increasingly influential on its surroundings. By “increasingly influential", it is meant that the streamlines of the fluid which is proximal to the boundary of the vortex increasingly curve to match the curvature of the streamlines of the vortex.

Now, let us focus on the fluid that was diverted upward by the leading edge. Notice that from Figure 1.1a to Figure 1.1d, this fluid (as it leaves the trailing edge of the wing) is pulled downward by the leading edge vortex more and more, such that at some critical moment, the momentum that was carrying it upwards and to the right is overcome by the growing pressure force created by the vortex. As can been seen in Figure 1.1c and Figure 1.1d, this fluid eventually creates a new vortex, indicating the onset of a vortex street. Subsequent vortices are observed to be smaller and weaker than the first vortex formed, and the wing loses lift and gains drag.

The important thing to realize is that the fluid that was diverted upwards at the leading edge (the fluid travelling along the upper surface of the wing) had strong up-right momentum as the first vortex formed. If the experiment was repeated at a very high angle of attack, continuity would require the vortex to grow much larger and be shed much more rapidly. At lower angles of attack (but still large enough for flow separation to occur), the first vortex would form more gradually and would remain smaller as it shed, because the shallow angle would give less up-right momentum to the fluid on the upper surface of the wing, and therefore this momentum would be easier for the vortex to influence, thereby causing the vortex to shed without growing significantly.

Since the change in momentum is proportional to the force and the duration the force acts, there is an optimal angle of attack for which the vortex is sufficiently large and the duration which it remains attached is sufficiently long. At this angle of attack, the duration which the vortex remains attached defines the duration of the half-cycle for flapping. At higher free stream velocities, this duration would be reduced because the vortex would grow proportionally faster. For a wing with larger chord length, this duration would be increased, because the vortex would need to grow more before becoming close enough to on the fluid travelling along the upper surface of the wing to force it off of its trajectory and allow for vortex shedding to occur. Therefore, we find that the optimal duration (and hence optimal frequency) is affected by the flow velocity and the chord length, which could have been anticipated by the strong performance correlation that flapping wing systems have to the reduced frequency.

For example, in the case of flapping wing energy harvesting the reduced frequency plays a significant role. It has been shown, using the Orr-Sommerfeld equation, that the power output of a flapping wing energy harvester is maximized when the frequency of flapping matches the most unstable frequency of the wake [12]. This “foil-wake resonance” occurs at a reduced frequency of 0.15, which is (amazingly!) the same reduced frequency found to maximize efficiency using a DNS solver using a Monte Carlo Scheme in [3]

1.2.4 Tip Vortex

The tip vortex, typically associated with efficiency losses in fixed wing aircraft and mitigated by the use of winglets, has actually been shown to enhance some flapping wing insects' flight performance. For low aspect ratio wings especially (i.e. where span-wise flow is more dominant), the tip vortex has been shown to "anchor" the leading edge vortex to the wing longer, thereby allowing the wing to benefit from its enhanced lift for a longer duration each stroke [13]. Moreover, it has been shown that biological flyers with low aspect ratios which operate at (relatively) high Reynolds numbers rely heavily on the tip vortex to generate the necessary aerodynamic forces for flight [13]. This is a three-dimensional phenomenon, so the present study does not simulate it, however, it could be the subject of a future study to artificially force the leading edge vortex to stay attached longer in order to simulate this effect.

1.2.5 Rapid Pitching

Similar to the Magnus effect, the rapid pitching of the wing at the end of the stroke can enhance lift by generating additional vorticity [14]. The key to whether the pitching of the wing increases or decreases lift is the relative phase angle of the pitching motion to the stroke motion. If the pitch reversal begins slightly before the stroke reversal, the wing generates additional vorticity, thereby enhancing lift. If the wing pitches after the stroke reversal, however, the vorticity generated is of the opposite sense, thereby decreasing lift.

1.2.6 Wake Capture

As the wing begins to reverse its stroke, the wake from the previous stroke may increase the flow velocity around the wing, thereby increasing lift [14]. It has been shown using a dynamically scaled fruit fly wing in mineral oil that the effect of wake capture is dependent on when, where, and how great the vorticity magnitude is throughout the flapping cycle.

1.2.7 Clap and Fling

The clap-and-fling mechanism relies on the interaction between wings to create additional suction. As the two wings approach each other, the air between them is forced out, generating thrust. Once the wings begin to separate and pitch about the leading edge, air rushes in to fill the gap, generating a vortex pair that produces the correct circulation around each wing to generate lift [14]. With regards to the current study, this effect is not observed, as it is a highly three-dimensional effect for low aspect ratio wings such as the butterfly.

Chapter 2 | Vortex Method

An in house two-dimensional (2D) vortex method code [4] is used to calculate the forces and moments on the wing at each time step. The wing is modeled by a line which is the projection of an infinitely long wing onto the plane of analysis. This wing can be descried in one of three coordinate systems. The first coordinate system is the global space-fixed system $\tilde{O} - \tilde{\xi}\tilde{\eta}$. In simulations where the ambient wind velocity is zero, this system is also the fluid-fixed system. The second system in which the wind can be described is the wing-fixed system $O - \xi\eta$. This system is attached to and aligned with the wing. The origin of this system is placed at the midpoint of the chord of the wing and the ξ-axis remains aligned with the chord as the wing pitches. The third coordinate system employed is the wing-translating system $\hat{O} - \hat{\xi}\hat{\eta}$. This system has its origin at the center of rotation of the wing (which is a distance a from the mid-chord) and it does not rotate with the wing. Instead the $\hat{\xi}$-axis remains aligned with the global space fixed $\tilde{\xi}$-axis regardless of the pitch angle of the wing. These three coordinate systems are shown in Figure 2.1.

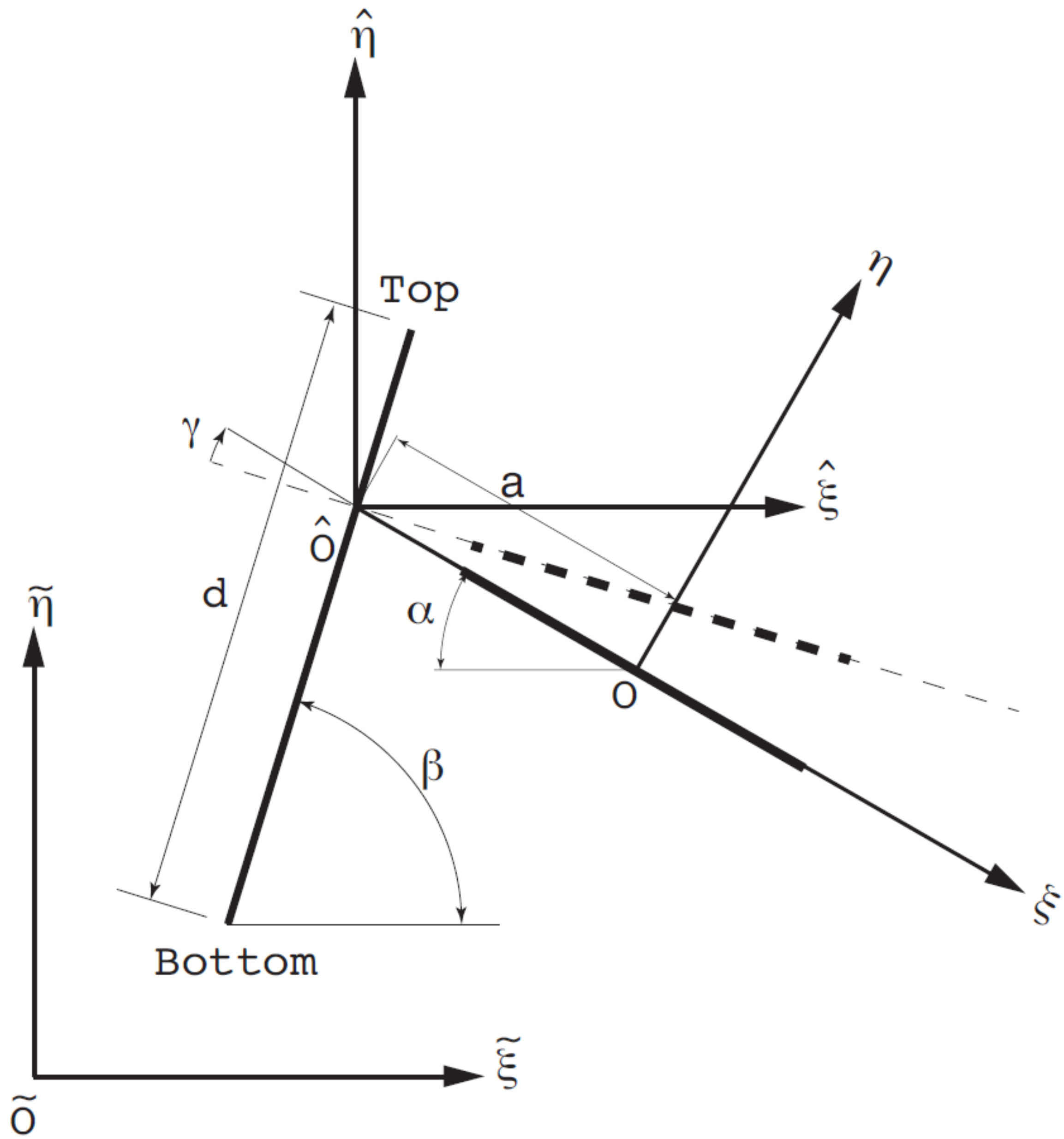


Figure 2.1 – Wing described in space-fixed, wing-fixed, and wing-translating coordinate systems

The wing of the MAV undergoes simultaneous pitching and heaving motions. The pitching motion is about the center of rotation ($\hat{O}$) and the heaving motion is along the stroke line of length d which makes a specified angle β with respect to the $\tilde{\xi}$-axis. The angle is known as the stroke plane angle, which in three dimensions is defined as the angle of the plane to which the center of rotation of the wing is constrained makes with the ground plane.

In Figure 2.1, it can be seen that there ar two different ways of defining the oreintation of the wing. The first is denoted as α, which is the angle the wing makes with respect to the $\tilde{\xi}$-axis. The second is denoted as γ, which is the angle the wing makes with respect to a line perpendicular to the stroke line (shown as a dashed line).

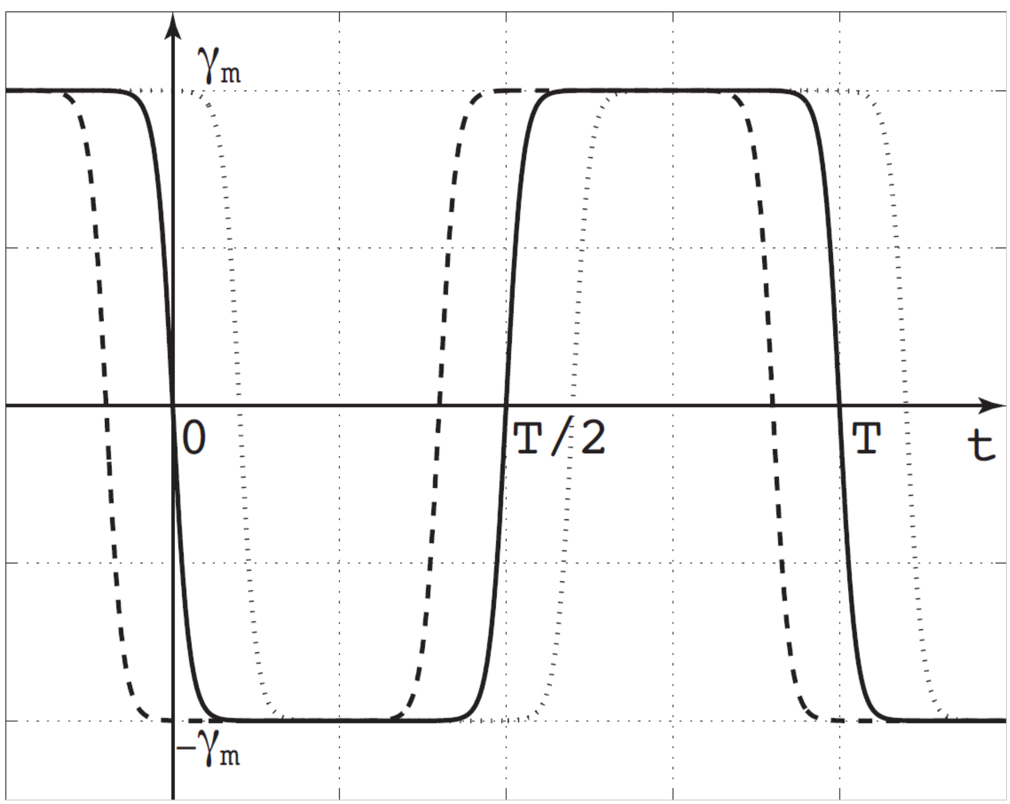


Figure 2.2 – Smoothed step function used to define pitch motion. The solid line represents symmetric pitching, the dashed line represents advanced pitching, and the dotted line represents delayed pitching.

The heaving motion is prescribed as a sinusoidal motion, while the pitching motion is a prescribed as a smoothed step function. This smoothed step function, shown in Figure 2.2, is used to reflect the sudden rotation exhibited by natural fliers such as flapping wing insects.

Figure 2.3 shows the view normal to the stroke plane. Φ_T indicates the three-dimensional angle that corresponds to the top-most heaving position of the wing, Φ_B indicates the three-dimensional angle that corresponds to the bottom-most heaving position of the wing, and l indicates the length of the wing. Therefore, the present 2D representation is actually a projection of the three-dimensional wing motion onto a vertical plane which is normal to the lateral direction of the body of the MAV.

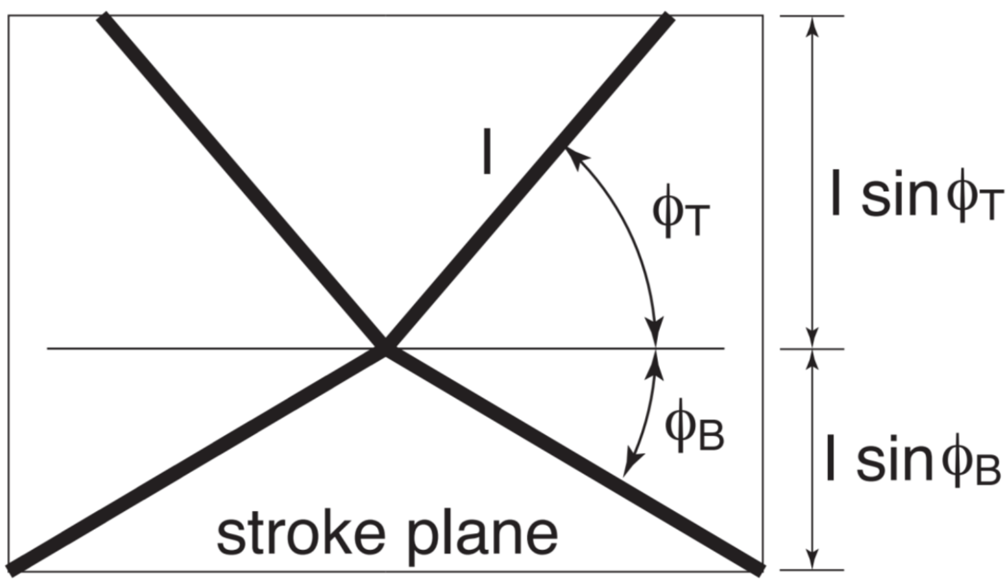


Figure 2.3 - View normal to stroke plane (from [4])

Vortices are placed on the wing surface to model the circulation-generating effect of viscosity in the boundary layer as in thin airfoil theory. Collocation points are spaced at the midpoint of the chord segment connecting adjacent vortices.

The complex potential of a line vortex is given as follows:

$$w(\tilde{\zeta}) = -\frac{i\Gamma}{2\pi} log(\tilde{\zeta} - \tilde{\zeta}_0) \ , \tag{2.1}$$

where $\tilde{\zeta}$ and $\tilde{\zeta}_0$ are global complex coordinates of the point of interest and the vortex, respectively.

Therefore, the complex velocity at a point $\tilde{\zeta}_i$ due to a vortex of circulation Γ_j at position $\tilde{\zeta}_{0j}$ is given by

$$\bar{v}_{ij} = -\frac{i\Gamma_j}{2\pi} \frac{1}{\tilde{\zeta}_i - \tilde{\zeta}_{0j}} \ . \tag{2.2}$$

The wing-normal component (indicated by superscript n) of the velocity at collocation point i due to a vortex j is given by,

$$v_{ij}^n = \frac{\Gamma_j}{2\pi r_i} \hat{\theta} \cdot \hat{n} = \left(\frac{1}{2\pi r_i} \hat{\theta} \cdot \hat{n}\right) \Gamma_j = a_{ij}^n \Gamma_j \ , \tag{2.3}$$

where Γ_j is the circulation of vortex j, r_i is the distance between the vortex and point i, $\hat{\theta}$ is the unit vector perpendicular to line ij, and $\hat{n}$ is the unit normal to the wing at collocation point i, and a_{ij}^n is the influence coefficient of wing-normal component of the velocity at collocation point i due to a vortex j. At each time step, vortices are shed from the leading and trailing edges and convected based on the local fluid velocity.

In order to solve for the strength of the bound vortices (Γ_{jb}) at each time step, the non-penetration condition and Kelvin's circulation theorem provide a set of linear equations, respectively,

$$\sum_{j=1}^{m} a_{ij}^n \Gamma_{jb} + \sum_{k=1}^{p} v_{ik}^n = V_i^n \ , \tag{2.4}$$

$$\sum_{j=1}^{m} \Gamma_{jb} + \sum_{k=1}^{p} \Gamma_{kw} = 0 \ , \tag{2.5}$$

where, a_{ij}^n and v_{ik}^n are the influence coefficients of wing-normal component of the velocity at collocation point i induced by the bound vortex Γ_{jb} and the wing-normal component of the velocity at collocation point i induced by the wake vortex k, respectively. In addition, V_i^n is the normal velocity component of the wing at collocation point i, m is the total number of bound vortices, p is the total number of wake vortices, and Γ_{kw} is the circulation of wake vortex k.

For m bound vortices with unknown circulations Γ_j , there will be *(m-1)* collocation points and thus *(m-1)* equations from Eqn. (2.4). Therefore Eqn. (2.5) allows for a unique solution to be determined for the Γ_j's by defining the m^{th} equation. Impulse-momentum theory is then used to calculate forces and moments on the wing in terms of the time derivatives of the linear and angular impulses.

The force and moment acting on the wing are determined by taking the time derivative of the linear and angular momenta,

$$L = -i\rho\Gamma z \ , \tag{2.6}$$

$$H = -\frac{1}{2}\rho\Gamma|z|^2 \ , \tag{2.7}$$

where L is the linear momentum of the fluid due to the vortex, i is the imaginary unit, ρ is the fluid density, Γ is the circulation of the vortex, z is the position of the wake vortex in the complex plane, and H is the angular momentum of the fluid due to the vortex.

Impulse-momentum theory is then used to calculate forces and moments on the wing. The sum of the linear momentum contributions and the sum of the angular momentum contributions of all of the bound and wake vortices are stored at each time step. A central difference scheme is then employed to calculate the linear and angular impulse. Dividing the impulse at each time step by the time step determines the force and moment on the fluid at each time step.

In order to produce the desired moment (about the center of rotation of the wing), the moment calculated from the time derivative of the sum of the angular momenta contributions from Eqn. (2.7) is adjusted using the standard formula for calculating the angular momentum about a non-stationary reference frame:

$$\overrightarrow{H_A} = \vec{H} + \vec{V}_A \times \vec{L} \ , \tag{2.8}$$

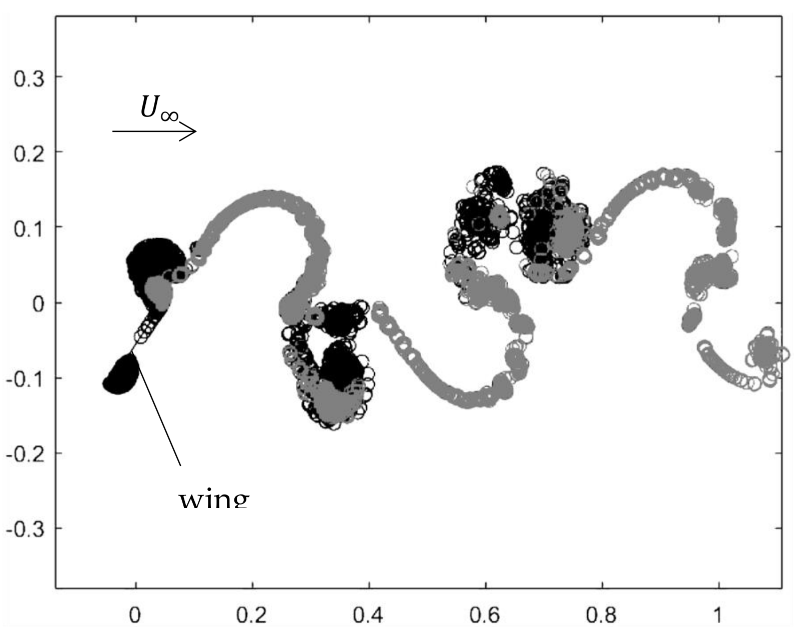
where $\overrightarrow{H_A}$ is the angular momentum with respect to point A (the center of rotation of the wing), $\vec{H}$ is the angular momentum with respect to the fluid frame, and $\vec{V}_A$ is the velocity of point A.

In complex notation, Eqn. (2.8) becomes

$$H_A = H + Im(\overline{V_A} * L) \ , \tag{2.9}$$

where Im() is "the imaginary component of" and the overbar indicates the complex conjugate. It is important to note that the resulting force and moment are by the wing and acting on the fluid. Therefore, the signs of the force and moment are reversed to obtain the force acting on the wing by the fluid.

The authors of [4] demonstrate good agreement between this discrete vortex method solver and the open source full Navier Stokes solver OpenFOAM over a wide range of kinematic parameters. Figure 2.4 shows an example of the wake developed by an oscillating flat plate wing after several cycles. The thin black line on the left is the wing (translating down), the fluid velocity is from left to right, and the black and grey circles represent vortices shed from the leading and trailing edges of the wing, respectively. The starting vortex can be seen at the far right of the figure, and the von Kármán vortex street can be clearly identified. Denda et al. [4] demonstrate good agreement between this vortex method solver and the open source full Navier Stokes solver OpenFOAM over a wide range of kinematic parameters.



2.1 Flow Separation Points

First, the thin line wing model in 2D allows for the assumption that separation occurs at the leading and trailing edges of the wing. This eliminates the need for computationally expensive high resolution boundary layer modeling to determine the separation point, such as the method described in [15].

2.2 Force Calculation

Next, an impulse-momentum is used to calculate the total force and moment on the wing. This method solves in much less time than traditional velocity-pressure

Figure 2.4 - Wake vortex plot of a flapping wing after several oscillation cycles.

based force calculation methods, at the expense of sacrificing knowledge of the local pressure distribution on the surface of the wing. However, for the flapping wing device designer looking for viable kinematics and power requirements, this pressure distribution would likely be integrated to obtain the net forces and moments anyway, so the reduction in computation time is preferred.

To calculate the force and moment acting on the fluid, a first order central difference scheme is employed as follows:

$$\vec{F}_n = \frac{\vec{P}_{n+1} - \vec{P}_{n-1}}{2 * dt} \ , \tag{2.10}$$

$$\vec{M}_n = \frac{\vec{H}_{n+1} - \vec{H}_{n-1}}{2 * dt} + \vec{V}_n \times \vec{P}_n \ , \tag{2.11}$$

where F is the force acting on the fluid, P is the linear momentum of the fluid, dt is the time step, M is the moment acting on the fluid, H is the angular momentum, V is the wing velocity, and n is the current time step.

2.3 Limitations of the Vortex Method

Despite the impressive performance of the vortex method presented in [4], there are several limitations. The first limitation is that the software does not utilize advanced computing techniques such as parallel processing. The Vortex Method produces hundreds or even thousands of point vortices, all which have to be tracked and included in the convection calculation at every other vortex location.

Given that velocity contributions can be calculated independently of one another, this computationally heavy step in the simulation is a prime candidate for parallel processing.

In addition, when simulating multiple oscillation cycles, one may find that the amplitude of the calculated pitching moment on the wing tends to slowly increase over time (Figure 2.5). This increasing pitching moment amplitude would make it challenging to couple the fluid solver to a rigid body solver, as the increasing moment would cause the MAV to tumble.

Therefore, it will be the goal of the present study to remedy these limitations in order to allow for a fluid-rigid body solver to be developed from the resulting fluid simulation software. First, the code will be analyzed for optimization using parallel processing. Next, the parallel processing algorithm will be tested against to serial algorithm to demonstrate the increase in computational efficiency. Finally, the root cause of the increasing pitching moment will be investigated and pin pointed so that a new approach can be implemented that does not include an increasing pitching moment.

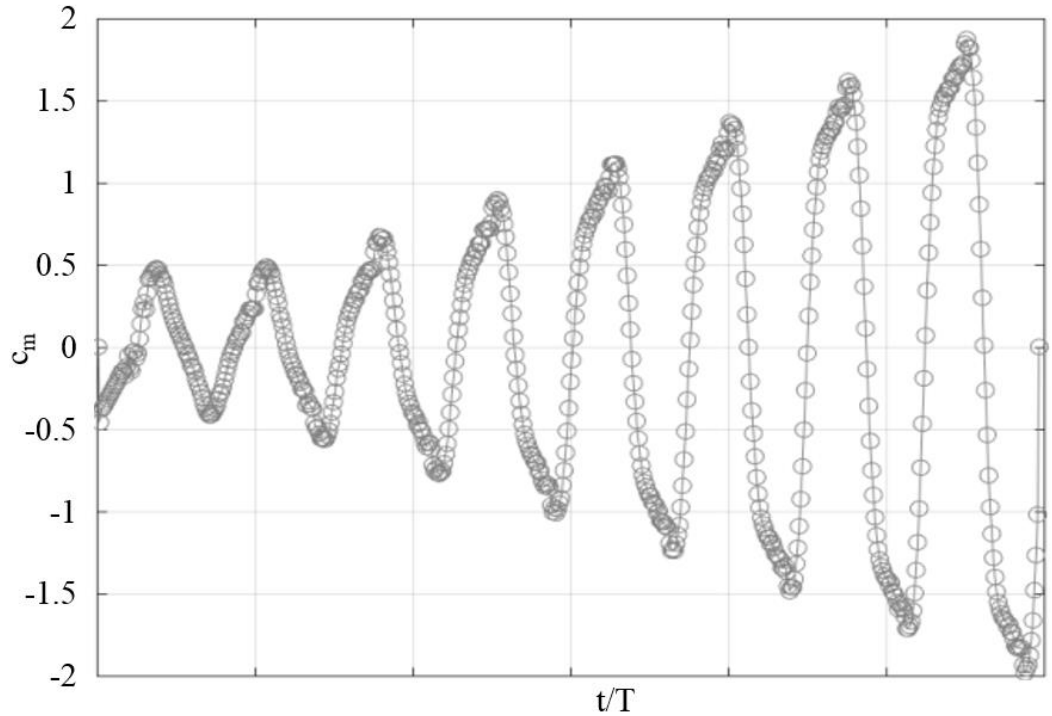


Figure 2.5 - Increasing pitching moment

Chapter 3 | Massively Parallel Implementation

The vortex method's speedup of 20X over traditional solvers is impressive, but one of the goals of the present study was to further improve the computational efficiency. One way this was achieved was by utilizing parallel processing. Parallel processing can be defined as the distribution of several repetitive mutually independent tasks to multiple processing compute cores. Each core effectively divides the time to complete the parallel task, producing dramatic improvements in computational efficiency in certain situations.

3.1 Hardware Selection

There are several different hardware options to implement parallel processing. Three examples are using multi-core CPU's, using a high-performance cluster, and using graphics processing units. Most consumer-end CPU's have only 4 cores (8 with hyperthreading) which run at a frequency on the order of 1 GHz. This number of cores does add some benefit, but for 1000's of operations per time step the vortex method could definitely utilize more cores. Rutgers does have a high-performance cluster with plenty of high frequency CPU cores, but access is limited, which makes rapid development difficult. Recently there has been a rise in the use of consumer grade graphics processing units (GPUs) for scientific computations.

Given that today's graphics cards are priced at less than $100 and have hundreds or even thousands of cores which each run at around 800MHz, it is easy to see why it was decided that the vortex method code would be modified to be implemented on a GPU. Even though each core of the GPU takes about 4 times longer to compute a given operation than a core on the CPU, the large number of them more than makes up for it.

To make the decision even easier, the code was already written in MATLAB, which has built-in functions that take advantage of NVIDIA's CUDA programming language to allow for the programmer to interface directly with the GPU using these built-in functions by simply reorganizing the computations into a GPU-friendly format.

3.2 Software Analysis

The vortex method code was analyzed to determine whether or not it could benefit from the highly parallel processing enabled by the GPU. Some of the functions did not take more time to run as the simulation progressed (i.e. with more wake vortices). Others only increased approximately linearly with the number of wake vortices ($\propto N$). However, one function in particular, the velocity calculation at each wake vortex point (for convection), increased with the square of the number of wake vortices ($\propto N^2$), which greatly reduced the efficiency of the solver after more than three cycles. Since the calculation of each velocity contribution is independent of the others, this function is a prime candidate for parallel implementation on the GPU. First, a pre-processor creates three $1 \times N^2$ arrays; it creates one for the location of the contributing vortex, one for the circulation of the contributing vortex, and one for the vortex location at which the velocity is to be determined. Therefore, the array indices represent one of the N^2 permutations required to calculate the velocity contribution from every vortex at every vortex point. Finally, once the computations have been performed on the GPU, a decoding function takes the output arrays and distributes them back to the individual velocity contribution arrays for each target location so that they can be summed.

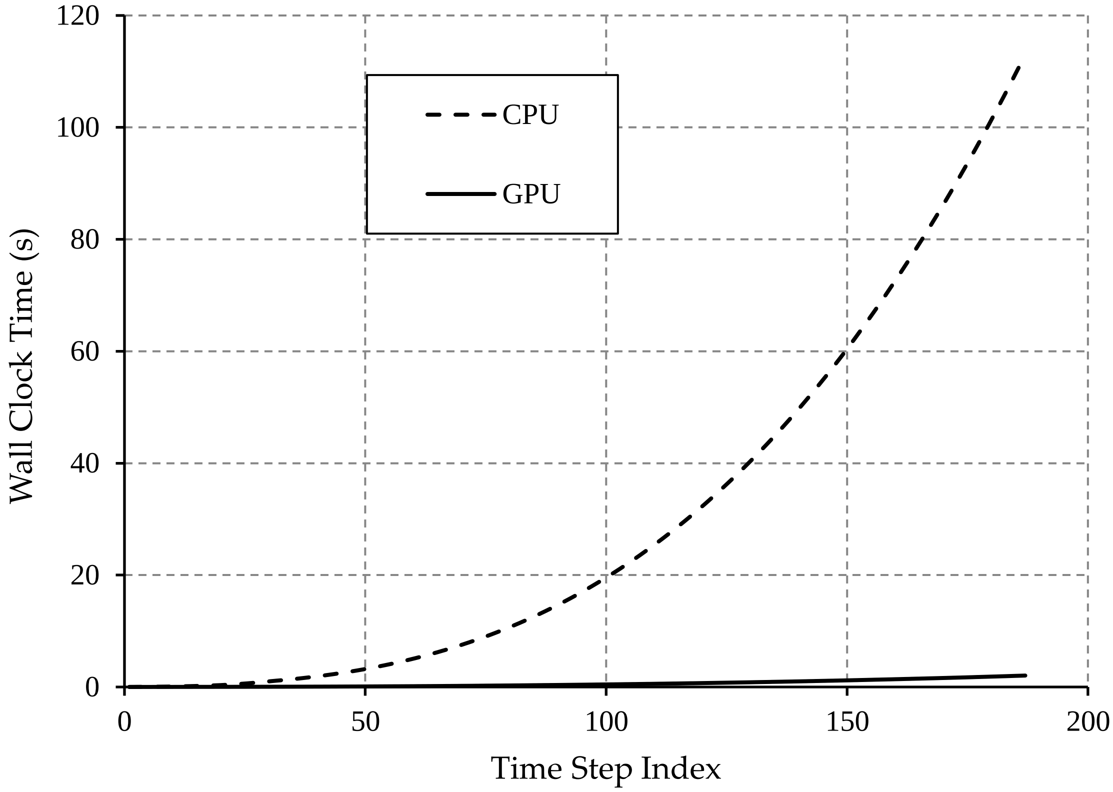


Figure 3.1 - Comparison of CPU vs. GPU Implementation

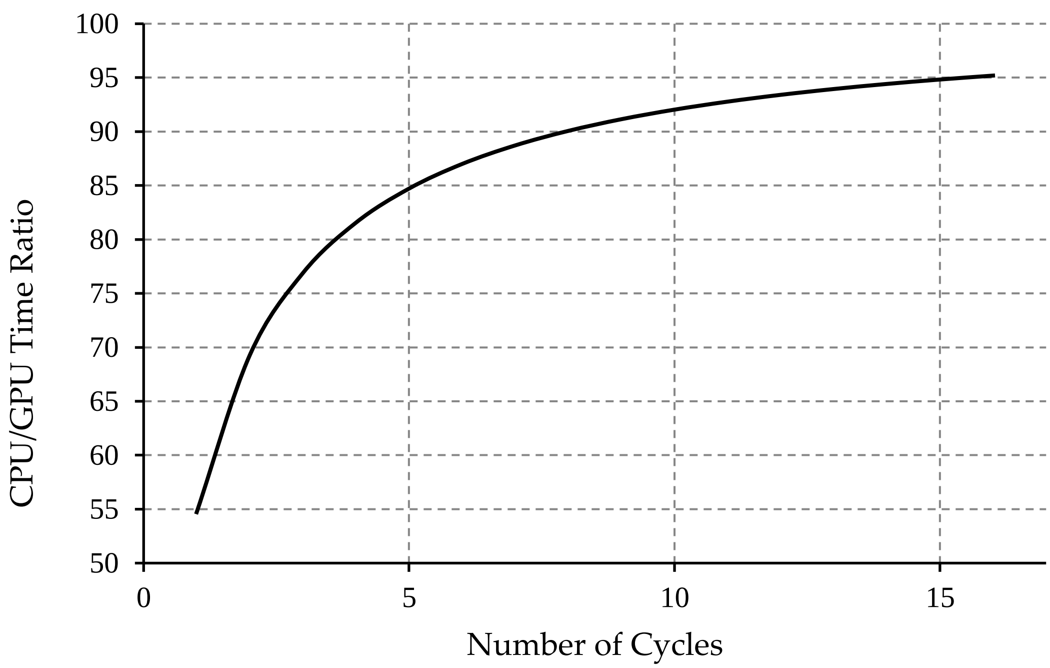


Figure 3.2 - Ratio of CPU to GPU Run Time vs. Number of Cycles Simulated

3.3 Results

The results of the GPU implementation were very encouraging. For simulations over 3 oscillation cycles the same output from the GPU runs over 80 times faster than on the CPU. Figure 3.1 shows the results of a simulation of only 1 cycle (55 times), because for larger numbers of cycles the time plot for the GPU is difficult to see. For example, the largest number of cycles tested on the GPU was 16.5 cycles, which took 1 hour and 15 minutes to run. By comparison, the same simulation would have taken just over 5 days on the CPU! Figure 3.2 illustrates the relationship between the number of cycles simulated and the ratio of the CPU time to the GPU time. Based on a regression analysis of the run time data (R^2~0.99999916), the ratio seems to asymptotically approach 101.07.

Chapter 4 | Reduced Order Analytical Solution

When simulating multiple oscillation cycles, one may find that the amplitude of the calculated pitching moment on the wing tends to slowly increase over time (Figure 2.5). In order to determine the root cause of the increasing moment, we first need to determine if the increase comes from the bound vortices or the wake vortices. Figure 4.1 shows the moment broken down into wake and bound

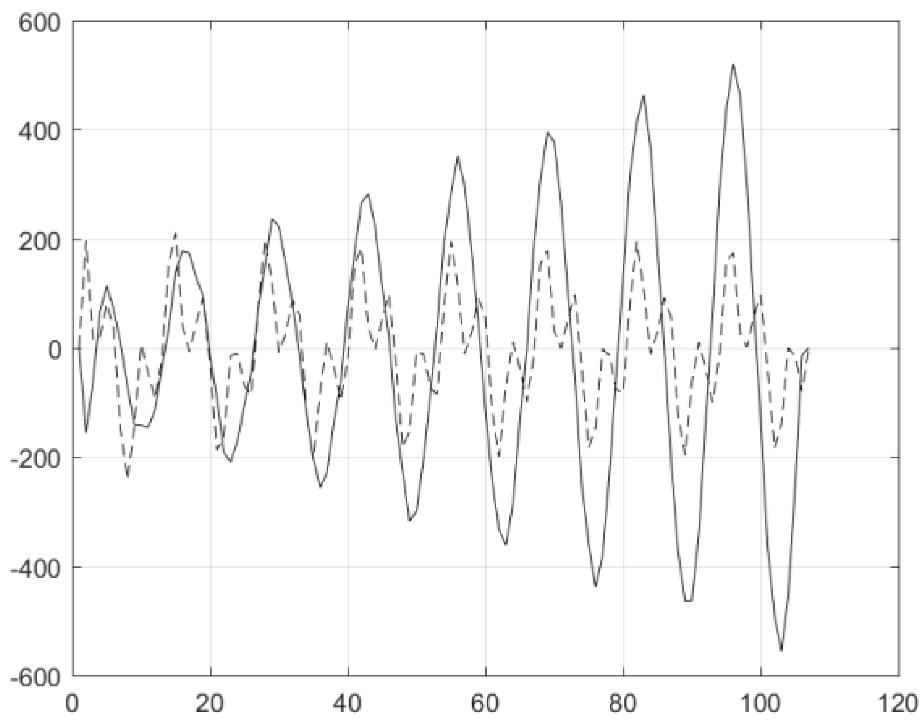


Figure 4.1 - Bound (dashed line) and wake (solid line) contributions to total moment

components. It is clear from this plot that the increase is coming from the wake vortices.

Next, a simplified series representation of the wake will show why the wake component of the moment calculation gives us this increasing amplitude, and how it might be attenuated in a real fluid. If we look at the diagram in Figure 4.2 and angular momentum contribution and location made by a single vortex in Eqn. (4.2) and Eqn. (4.2), respectively,

$$H^j = \sum_{i=1}^{j} -\frac{1}{2}\rho\Gamma_i\left|z_i^j\right|^2 \tag{4.2}$$

$$z_i^j = \binom{x}{y} \tag{4.2}$$

it may seem that the increasing distance from the wing in the x-direction is causing the increasing moment.

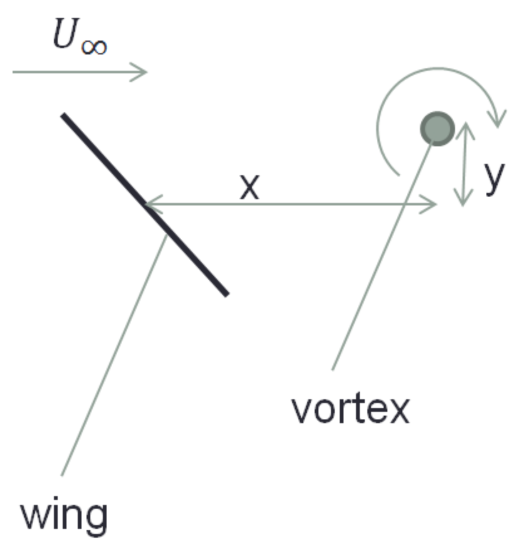


Figure 4.2 - Single wake vortex diagram

4.1 Reduced Order Model 1

In order to test this hypothesis, a reduced order model called Model 1 is constructed where the shed vortices move with the free stream but are constrained to the x-axis (Fig. 6). Note that the x-axis (parallel to the wind direction) was chosen out of convenience, and that any inclined wind direction would also produce valid results with the corresponding wake vortices distributing in the parallel direction.

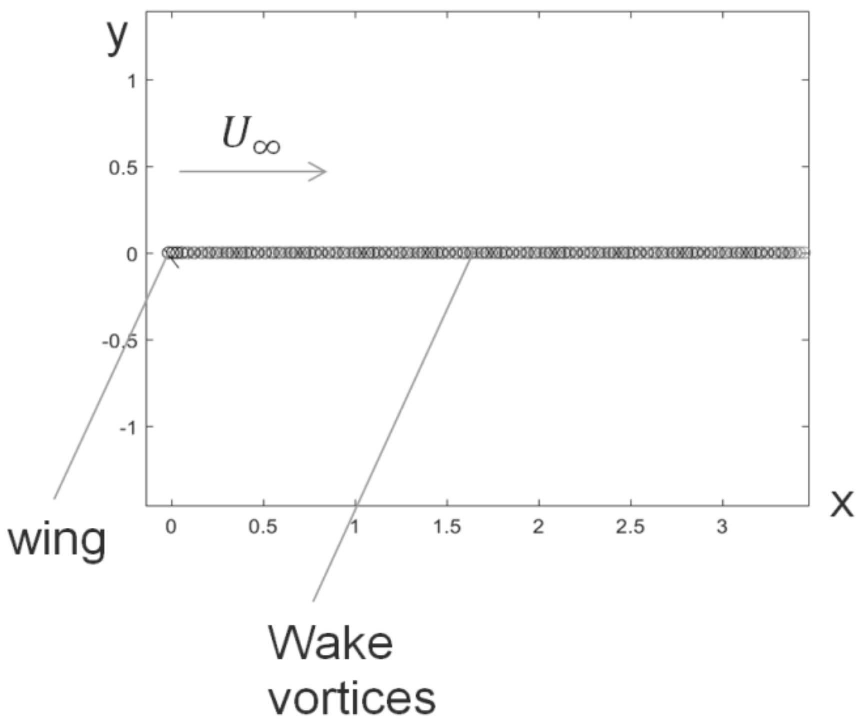


Figure 4.3 – Model 1

If the hypothesis is correct, the moment should increase due to the increasing x-distances of the wake vortices However, we find that even with the same circulation values as before, the moment doesn't increase despite the increasing x-distance of the far field vortices (Figure 4.4).

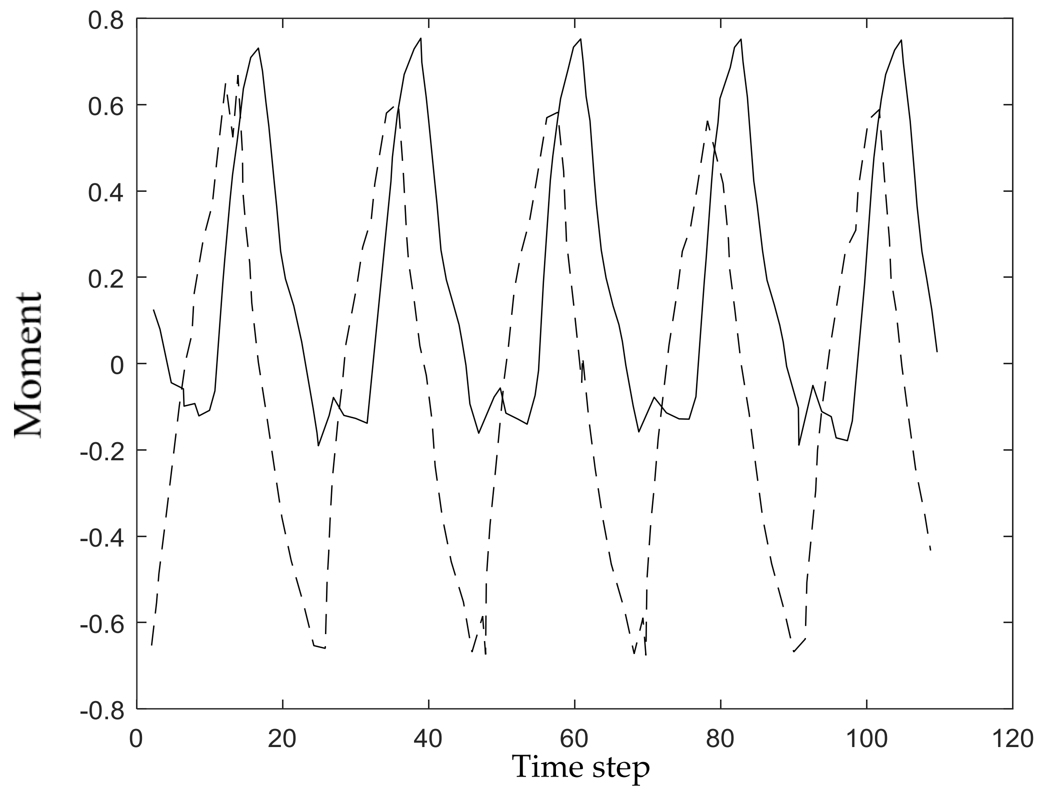


Figure 4.4 - Bound (dashed line) and wake (solid line) contributions to total moment for Model 1

It therefore appears that the periodic nature of the circulation allows neighboring vortices' angular momentum contributions to effectively cancel each other out when the wake is confined to the x-axis (Figure 4.5).

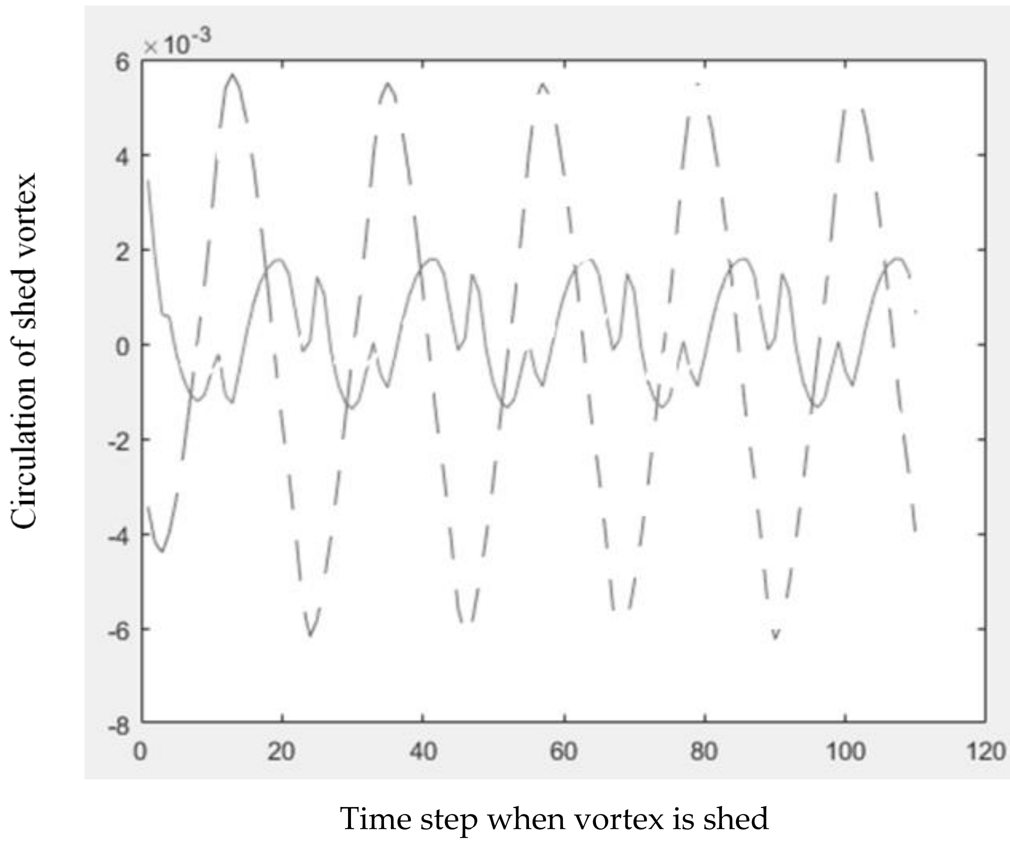


Figure 4.5 - Circulation values of wake vortices shed from the leading (dashed line) and trailing (solid line) edges

However, confining vortices to the x-axis is not physical. Therefore, the resulting constant moment amplitude (Figure 4.4) only goes to disprove the theory that the increasing distance of the average vortex is the root cause of the increasing moment.

4.2 Reduced Order Model 2

Consider a second simplified model called Model 2 of the vortex shedding procedure which adds a sinusoidal y-position to the wake vortices where once a given vortex is shed, it is fixed and not convected by the local fluid velocity (Figure 4.6).

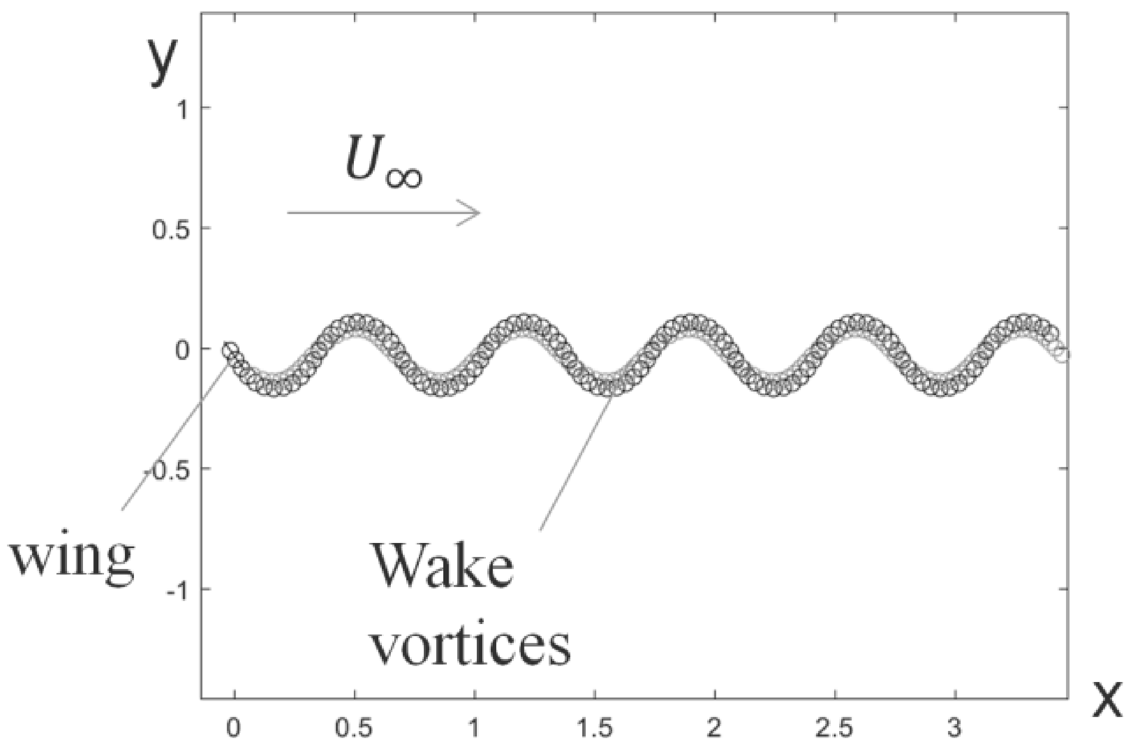


Figure 4.6 – Model 2

Surprisingly, this seemingly small change from the previous case (adding a sinusoidal y-position to the wake vortices) produces increasing moment amplitude, such as in Figure 2.5. Functional forms of the moment for Model 2 (Figure 4.6) can be derived as follows. The moment about the center of rotation of the wing is calculated as follows:

$$\vec{M} = \frac{d}{dt}\left(\overrightarrow{H_0}\right) + \vec{v} \times \vec{P} \ , \tag{4.3}$$

where $\overrightarrow{H_0}$ is the angular momentum of the fluid with respect to the center of rotation of the wing, $\vec{v}$ is the linear velocity of the center of rotation of the wing and $\vec{P}$ is the linear momentum of the fluid with respect to the space fixed frame.

The total angular momentum, $\overrightarrow{H_0}$, is the sum of the individual contributions of the wake and bound vortices. The contribution of the wake at time step j is defined as follows:

$$H^j = \sum_{i=1}^{j} -\frac{1}{2}\rho\Gamma_i\left|z_i^j\right|^2 \ , \tag{4.4}$$

where Γ_i is the circulation of vortex i, ρ is the density of the fluid, and $\left|z_i^j\right|$ is the distance from the center of rotation of the wing and vortex i at time step j. For simplicity, let us assume a flapping frequency and wing tip amplitude of : $\mathrm{f} = \frac{1}{2\pi}$ Hz and $A_x = 1$, respectively. For the case where vortices are not convected and remain where they were shed as in Figure 4.6 we have:

$$z_i^j = \begin{pmatrix} x \\ y \end{pmatrix} , \tag{4.5}$$

where x is the distance traveled by the wing since vortex i was shed and y is the difference between the y-coordinate of the vortex and the y-coordinate of the center of rotation of the wing at time step j. This gives (by the Pythagorean theorem),

$$\left|z_i^j\right|^2 = \left(U_\infty \Delta t(j-i)\right)^2 + (\sin(i\Delta t) - \sin(j\Delta t))^2 \ . \tag{4.6}$$

In addition, the circulation of a given wake vortex can be approximated as follows:

$$\Gamma_i \sim A \sin(i\Delta t) \ . \tag{4.7}$$

In general, the periodic circulation function would be a Fourier series, but as long as the Fourier coefficients are constant, each term of the Fourier series will produce the same functional form as the sine function will when multiplied by the magnitude of the distance. Therefore, the total angular momentum can be approximated by the following relations:

$$H^j = \sum_{i=1}^{j} -\frac{1}{2}\rho\Gamma_i\left|z_i^j\right|^2 \ , \tag{4.8}$$

$$H^j = -\frac{1}{2}\rho\sum_{i=1}^{j} A\sin(i\Delta t)\left[\left(U_\infty\Delta t(j-i)\right)^2 + (\sin(i\Delta t)-\sin(j\Delta t))^2\right] \ . \tag{4.9}$$

Note that the current time is simply $j\Delta t$ and that the time at which a given vortex i was shed is $i\Delta t$. Figure 4.7 shows a plot of the angular momentum of Model 2 given by Eqn. (4.9).

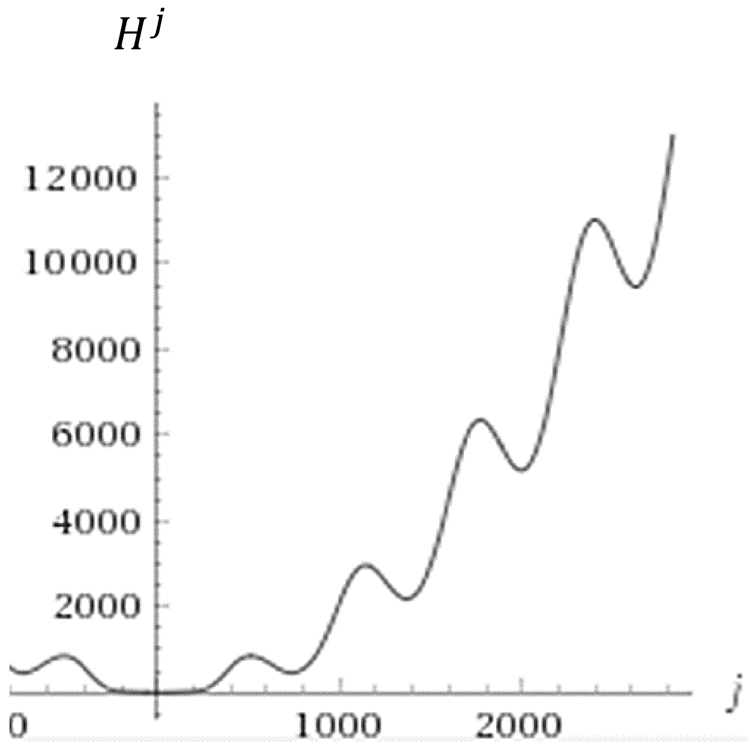


Figure 4.7- Angular momentum vs. time step (Model 2)

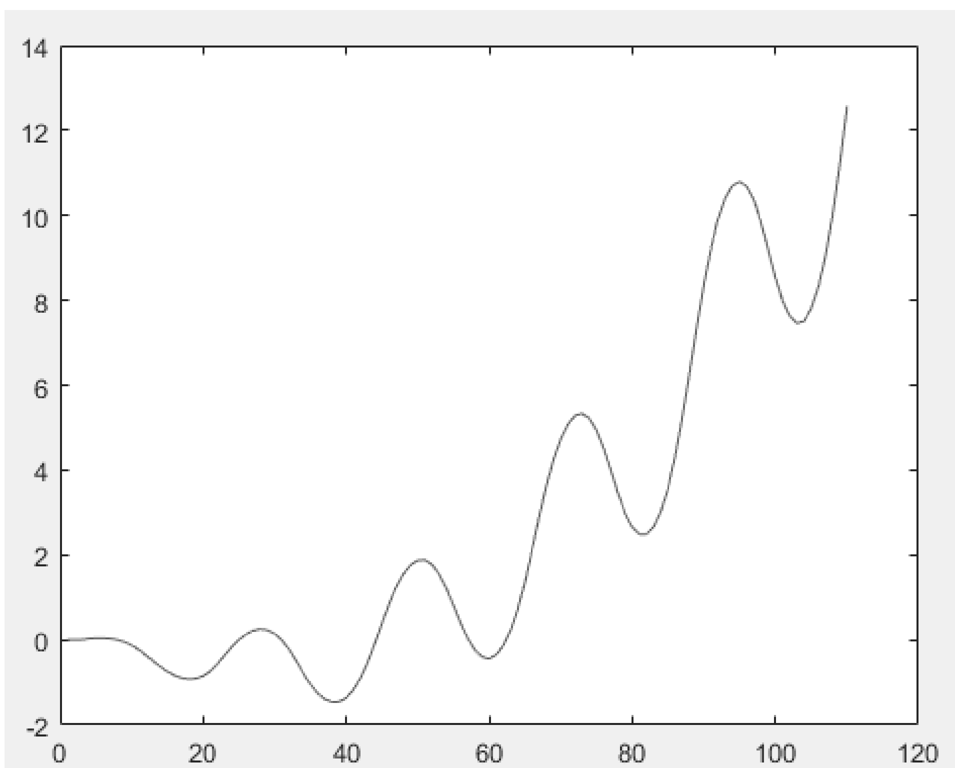


Figure 4.8 - Typical angular momentum from vortex method

Note the striking resemblance between the plot of Eqn. (4.9) (Figure 4.7) and a typical angular momentum plot of a fully convected wake from the original non-simplified code (Figure 4.8). Therefore, by comparing the wake of original code (Figure 2.4) and Model 2 (Figure 4.6) and by also comparing the angular momentum plots of Figure 4.7 and Figure 4.8, one may conclude that the Model 2 model provides an adequate first order representation of the physics of the problem. Moreover, Model 2 allows one to obtain an approximate continuous analytical solution for the pitching moment, instead of the nonlinear representation from the actual code. Therefore, we can examine the mathematics of this second model to understand more about the root cause of the increasing moment and how to mitigate it.

The $A\sin(i\Delta t) * (\sin(i\Delta t) - \sin(j\Delta t))^2$ term from Eqn. (4.9), when summed, behaves like $t sin(t)$, where $t = j\Delta t$.

This can be seen in the approximate functional form:

$$\int_{t_1}^{t_2} A sin(\tau)(\sin(t) - \sin(\tau))^2 \, d\tau \tag{4.10}$$

$$= \frac{1}{12} A(6\sin(t)(2t_1 - 2t_2 - \sin(2t_1) + \sin(2t_2)) + (15 - 6\cos(2t))\cos(t_1) + 3(2\cos(2t) - 5)\cos(t_2) - \cos(3t_1) + \cos(3t_2))\,.$$

Upon the expansion of Eqn. (4.10), we see that part of the results includes the following terms:

$$\frac{1}{12}A(6\sin(t)\,(2t_1 - 2t_2)\,. \tag{4.11}$$

In the original code, $t_1 = 0$ and $t_2 = j\Delta t$ (where j = current time step). Therefore, this produces the term that behaves like $t * \sin(t)$ which causes the increasing moment amplitude, even in this simplified case.

By eliminating the far field wake vortices, we effectively prevent the (t*sin(t)) term from growing in amplitude, by limiting the time t over which the sum is taken. For example, in Eqn. (4.11), $t_1 - t_2 = const.$ (e.g. the threshold time). In the discrete analog of Model 2 (Eqn. (4.9)), the removal of vortices would cause i to increment each time step, such that $j - i = const$. Physically, this represents the decay of old vortices due to viscosity. However, one issue with simply removing vortices is that it introduces spurious high frequency perturbations to the flow field. This tends to cause instability in the solution of the bound vortex circulation (Eqns. (2.4) and (2.5)).

Chapter 5 | The Lumped Wake Vortex Method

5.1 Vortex Lumping

In order to overcome this issue of solution instability, two lumped vortices are generated, one for removed vortices which were originally shed from the leading edge of the wing and one for those shed from the trailing edge. These lumped far field vortices approximate the effect that the removed point vortices have on the solution to Eqns. (2.4) and (2.5) by taking on the average position and circulation of their respective aggregated point vortices. In essence, instead of removing some wake vortices to mimic the vortex decay, we replace them by a pair of lumped wake vortices such that the lumped vortices take the place of the removed vortices in the linear and angular momentum calculations.

The positions and circulation magnitudes of the two lumped vortices are given by:

$$ {}^{LE}Z_{lumped}^{j} = \frac{1}{j}\sum_{i=1}^{j-k} z_i^j \ , \tag{5.1} $$

$$ {}^{LE}\Gamma_{lumped}^{j} = \frac{1}{j}\sum_{i=1}^{j-k} \Gamma_i^j \ , \tag{5.2} $$

$$ {}^{TE}Z_{lumped}^{j} = \frac{1}{j}\sum_{l=1}^{j-k} z_l^j \ , \tag{5.3} $$

$$ {}^{TE}\Gamma_{lumped}^{j} = \frac{1}{j}\sum_{l=1}^{j-k} \Gamma_l^j \ , \tag{5.4} $$

where k is the index corresponding to the time step when lumping begins, LE indicates the lumped leading edge vortex, TE indicates the lumped trailing edge vortex, i indicates that the vortex was shed from the leading edge, and l indicates that the vortex was shed from the trailing edge.

Another approach that was tested was to use the sum of the circulations of the removed vortices for the circulation of the lumped vortex instead of the average. Due to the periodicity of the circulation values of the shed vortices, the amplitude of the sum of the circulation values of the removed vortices remains a small constant over time. The solutions using each of the two approaches show no appreciable differences due to the fact that the removed vortices have a negligible effect on the solution just before they are removed. Therefore, it is better to use the sum to be consistent with the conservation of circulation (Kelvin's Theorem). The total linear and angular momenta due to the wakes vortices then becomes,

$$L_{total}^{j} = L_{LE}^{j} + L_{TE}^{j} + \sum_{i=j-k-1}^{j} L_{i}^{j} + \sum_{l=j-k-1}^{j} L_{l}^{j} \quad , \tag{5.5}$$

$$H_{total}^{j} = H_{LE}^{j} + H_{TE}^{j} + \sum_{i=j-k-1}^{j} H_{i}^{j} + \sum_{l=j-k-1}^{j} H_{l}^{j} \quad , \tag{5.6}$$

where L_{LE}^{j} and L_{TE}^{j} are the linear momentum contribution of the lumped leading and trailing edge vortices, respectively. Furthermore, L_{i}^{j} and L_{l}^{j} are linear momentum contributions of the leading (i) and trailing (l) edge vortices. The notation for the angular momentum contributions are similarly defined. In addition, the superscript j refers to the time step index j.

In order to simulate wake vortex decay, one could simply remove the oldest point vortices after a predetermined critical threshold time like in the initially attempted wake vortex removal model. This method of vortex removal also has the added benefit of reducing the number of calculations in the far field wake, where the additional computational effort would provide little value. However, since the present method of force calculation relies upon impulse momentum theory, the forces and moments are calculated as the differential of the momentum. Therefore, simply removing vortices will result in large spikes in the force and moment plots, as the removal process represents a step change in momentum.

5.2 Force and Moment Calculation

In performing the central difference calculation to get the derivatives of the momenta at step n, we use the momenta at time steps n+1 and n-1. The additional issue arises here since the removed vortices at n+1 and n-1 differ, i.e., more vortices have been removed at n+1 than at n-1. In order to overcome this issue, only the linear and angular momenta contributions of the point vortices remaining at time step n+1 are used in the calculation of the moment at time step n. In other words, the momentum contributions of the vortices which will be removed by time step n+1 are subtracted from the total momentum at time step n-1 before calculating the differentials (Eqns. (2.10) and (2.11)).

Physically, this adjustment to the momentum at time step n-1 can be explained as follows. Non-lumped vortices are closer to the wing than the lumped vortices and therefore have a stronger influence on the force acting on the wing. Therefore, if the adjustment was not made, the differential in the momentum calculation would be unbalanced, as it would be comparing two different sets of vortices. The large spike in the force amplitude one would obtain as a result of not making the adjustment would be due to the comparison of two dissimilar representations of the flow field.

To illustrate this process, Figure 5.1 shows example wake plots at time steps n-1 and n+1. The white vortices at time step n-1 will be eliminated by time step n+1, so they are excluded from the calculation of the moment at time step n.

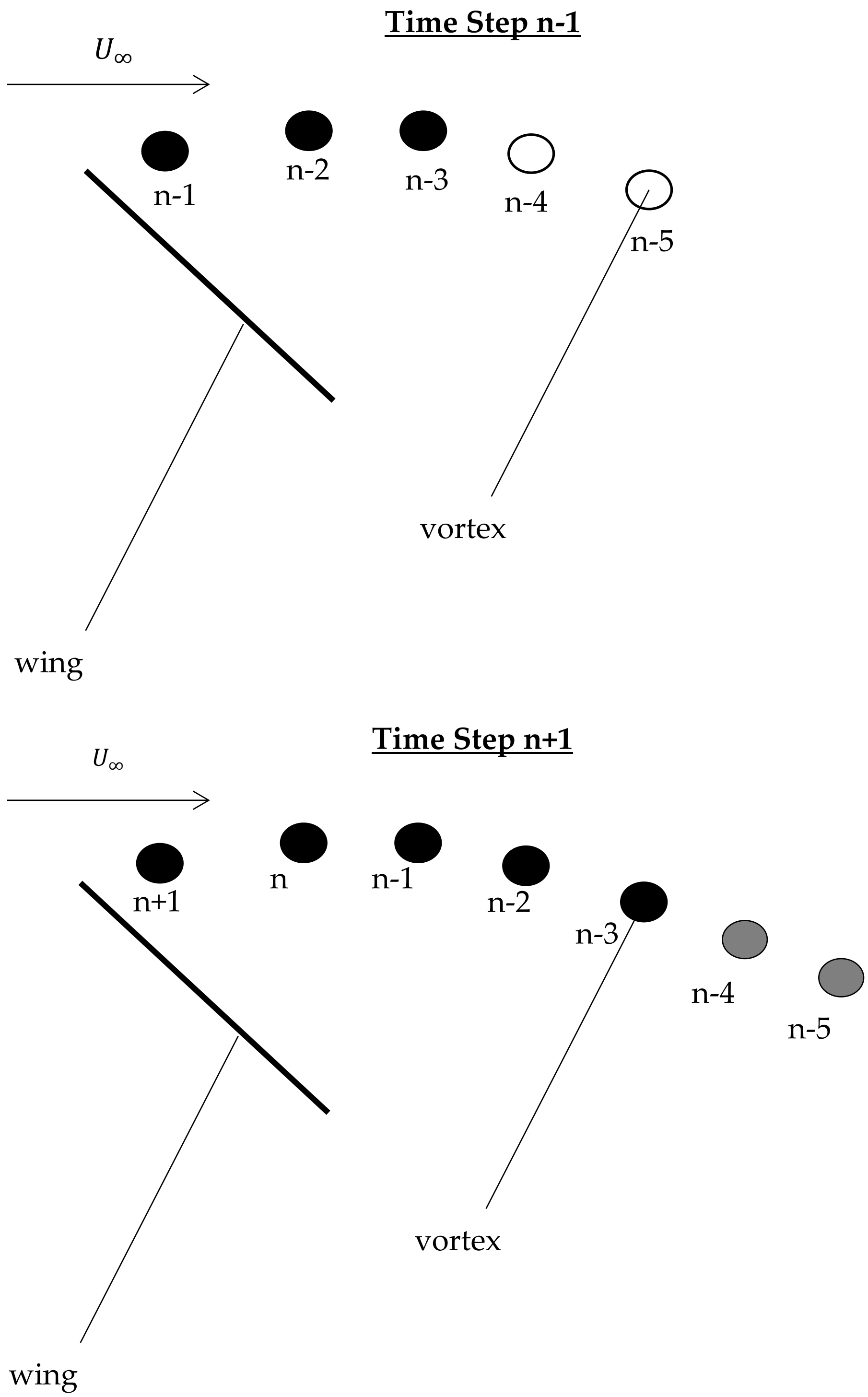


Figure 5.1 - Vortex wake plots for time steps n-1 (top) and n+1 (bottom). Black vortices are used for the moment calculation at n, since the grey vortices will be removed at time step n+1, white vortices (present in step n-1) are not included in the moment calculation at n.

The moment at time step n for this example (Figure 5.1) would be calculated as follows:

$$\vec{M}_n = \frac{\vec{H}_{n+1} - \vec{H}_{n-1} - \vec{h}_{n-4}^{n-1} - \vec{h}_{n-5}^{n-1}}{2 * dt} + \vec{V}_n \times \vec{P}_n \ , \tag{5.7}$$

and similarly, for the force,

$$\vec{F}_n = \frac{\vec{L}_{n+1} - \vec{L}_{n-1} - l_{n-4}^{n-1} - \vec{l}_{n-5}^{n-1}}{2 * dt}, \tag{5.8}$$

where $\vec{h}_{n-4}^{n-1}$ and $\vec{h}_{n-5}^{n-1}$ are the angular momentum contributions of wake vortices *n-4* and *n-5* at time step *n-1*, and l_{n-4}^{n-1} and $\vec{l}_{n-5}^{n-1}$ are the linear momentum contributions of wake vortices *n-4* and *n-5* at time step *n-1*. The momenta from these vortices are only subtracted for the moment calculation at time step n. All subsequent time steps still include their momenta.

5.3 Comparative Example

In order to further examine the physical implications of removing the momentum contributions of to-be-removed vortices at time step n-1, one may compare the results of a simulation which includes these contributions to one that does not.

Figure 5.2 and Figure 5.3 show the (incorrectly) calculated forces on the wing in the x and y directions, respectively, of a simulation that includes the momentum contributions of to-be-removed vortices at n-1. Compare these plots to Figure 5.4 and Figure 5.5, which show the (correctly) calculated forces on the wing in the x and y directions, respectively, of a simulation that removes the momentum contributions of to-be-removed vortices at n-1. Notice how the forces are much higher in the case where the momentum contributions at n-1 are included.

Physically, this is because the change in fluid momentum in this case is calculated using two dissimilar sets of vortices. Therefore, in order to correctly determine the forces on the wing while excluding the sum of the viscous forces, one must remove the momentum contributions of vortices which are removed by step n+1 at step n-1 when calculating the forces via impulse-momentum theory.

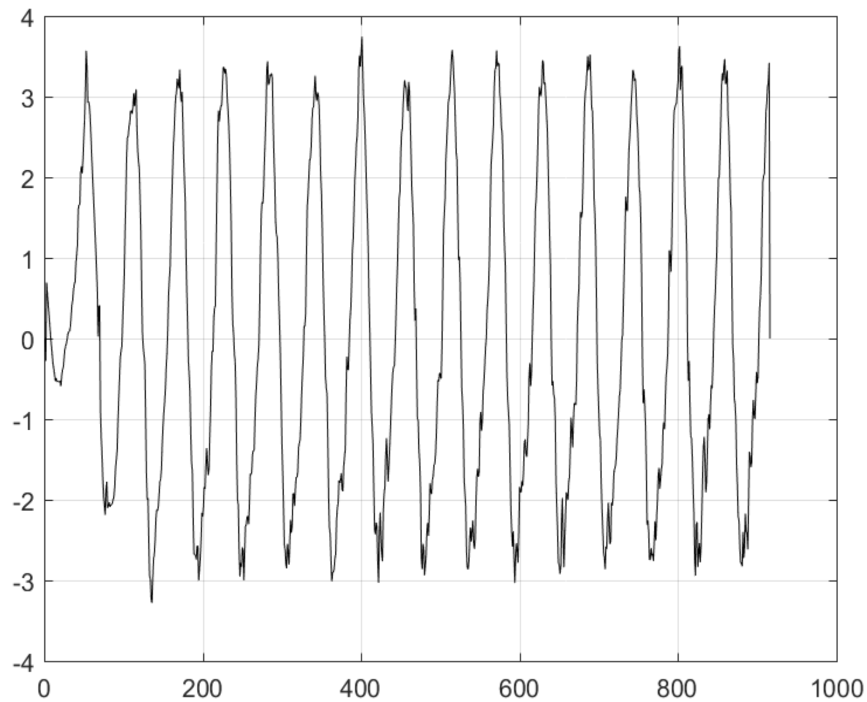


Figure 5.2 - Incorrectly calculated force on wing in x-direction

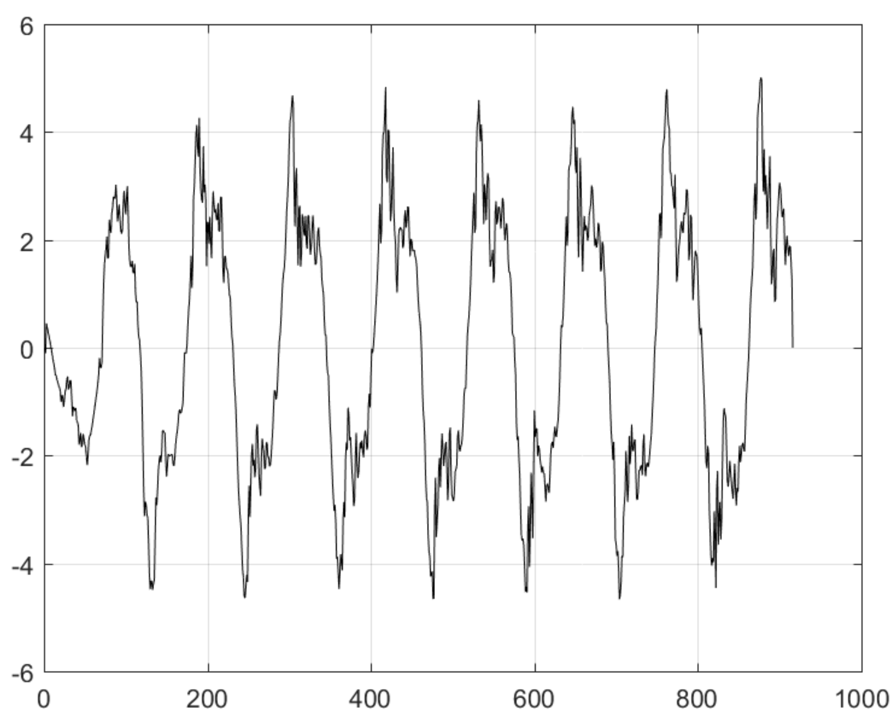


Figure 5.3 - Incorrectly calculated force on wing in y-direction

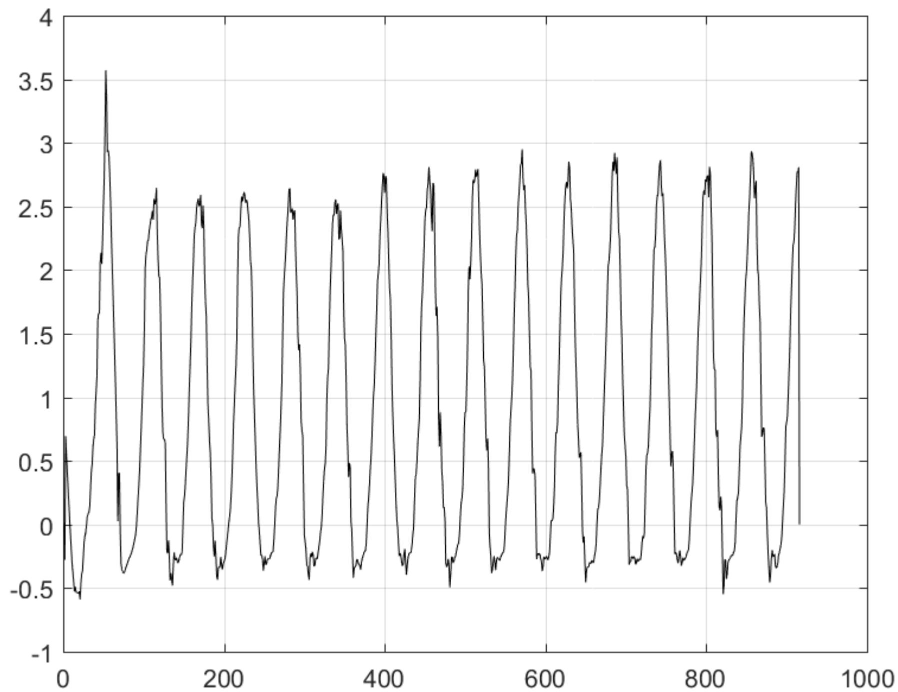


Figure 5.4 - Correctly calculated force on wing in x-direction

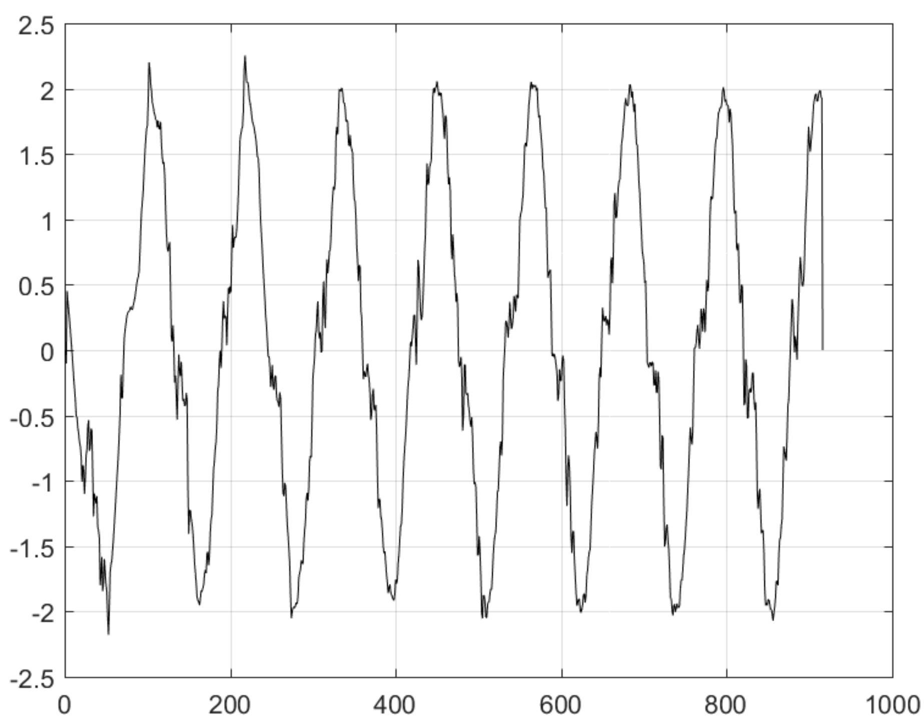


Figure 5.5- Correctly calculated force on wing in y-direction

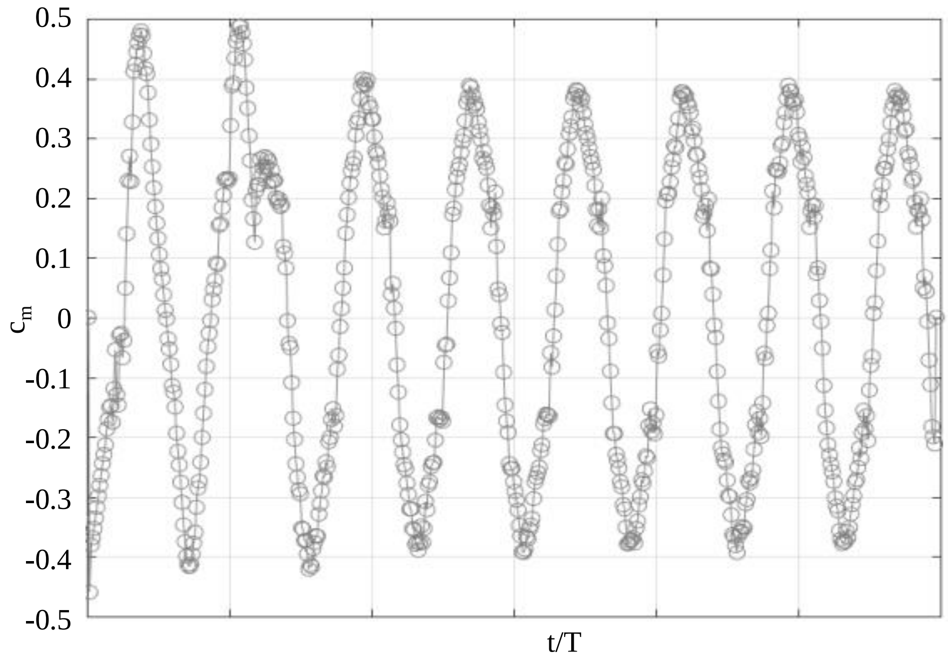


Figure 5.7 - Constant amplitude pitching moment

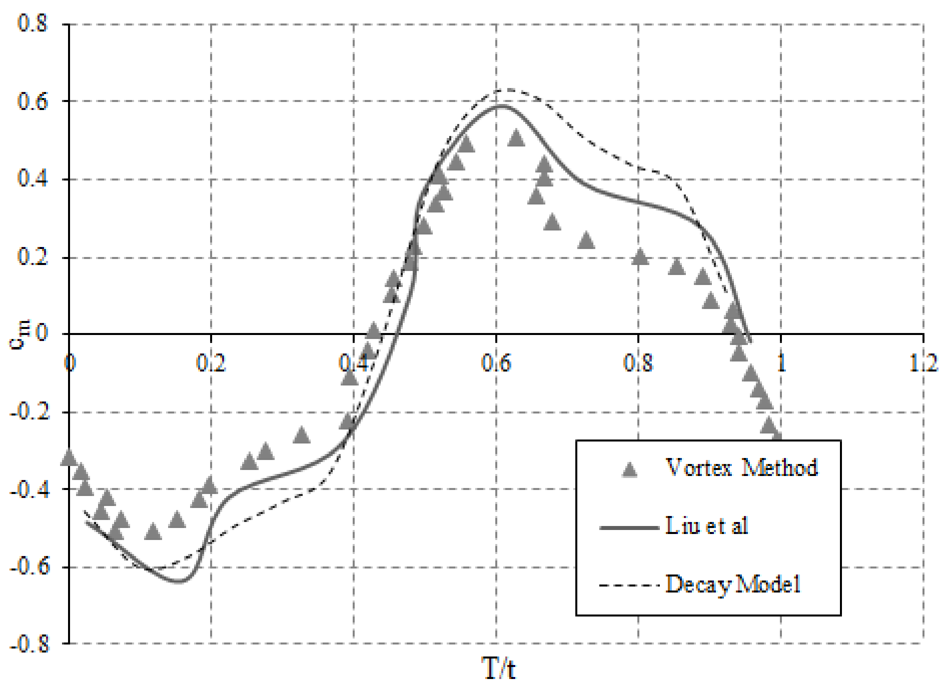


Figure 5.6 - Comparison of results with published data

5.4 Key Results

5.4.1 Constant Moment Amplitude

The resulting constant amplitude moment plot is shown in Figure 5.7. It can be seen in this figure that the moment amplitude remains very nearly constant at the value of the amplitude for the first flapping cycle from the original vortex method (Figure 2.5).

5.4.2 Comparison with 2D Navier Stokes Simulation

In order to further verify the moment coefficient results, the original method and the new lumped wake vortex method were compared to published results [5] of a full Navier Stokes solution of a flapping wing (Figure 5.6). As can be seen from the figure, the new model slightly over predicts the pitching moment while the original model slightly under predicts. Qualitatively, the models both agree well and would provide sufficient accuracy given the radically lower computation time.

5.4.3 Further Reduction in Computation Time from Vortex Removal

One major advantage of this strategy is that it not only better represents the physical system being modelled; it also reduces the number of calculations in the

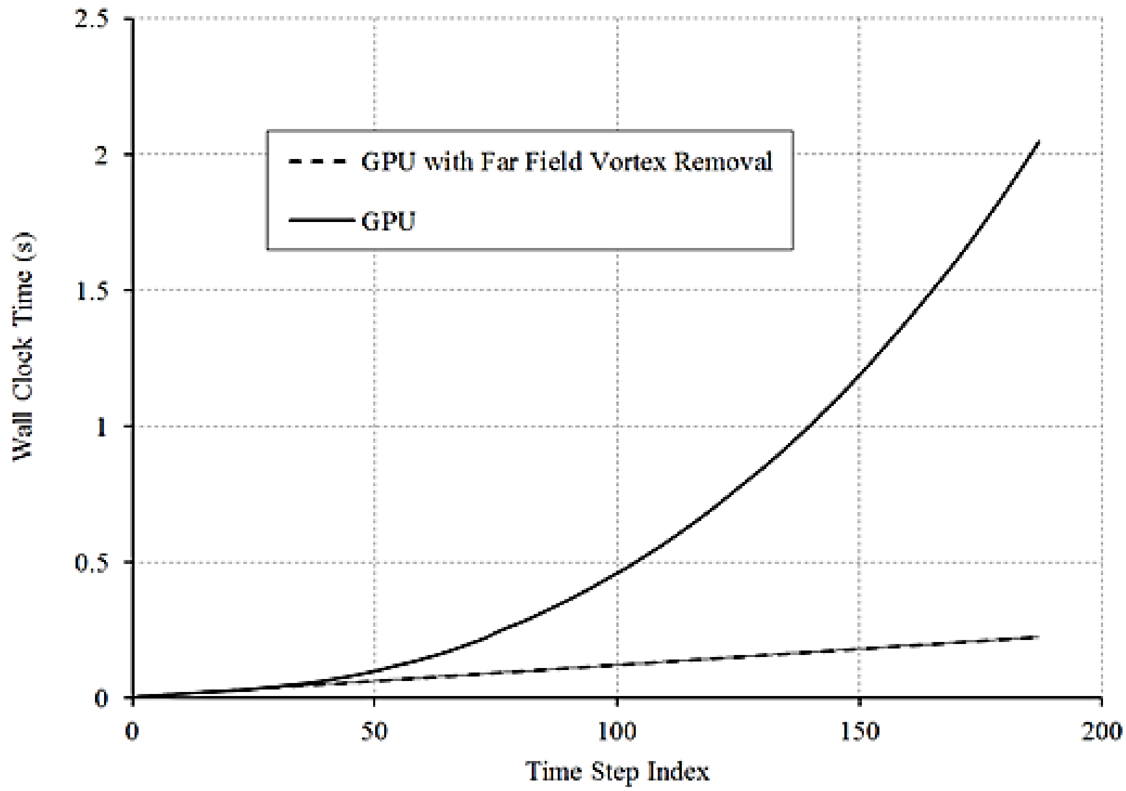


Figure 5.8 - Comparison of computation time with and without vortex removal

far field wake, where the additional computational effort would provide little value. Figure 5.8 shows how much computation time is saved using the lumped wake vortex method over the original parallel implementation without the viscous decay model.

Since the computation time of the original method goes as N^2 (where N is the number of time steps) and the computation time for the lumped wake vortex method goes as N, the difference in computation time is striking, especially for simulations which involve several flapping cycles or high temporal resolution.

5.4.4 Vortex Lumping Threshold

It was found that after point vortices convect past a certain distance, the choice of when to lump them has very little impact on the results. Therefore, the time threshold is determined as the duration required to convect a vortex moving at the free stream velocity to the point where its induced velocity at the center of the wing is less than 0.5% of the free stream velocity.

5.4.5 Varying Incident Flow Angles

In order to verify that the method of maintaining constant moment amplitude was sufficiently general, simulations were carried out for several different incident flow velocity angles. Figure 5.9 shows the wake vortex plots and corresponding moment plots for incident flow velocities of 0°, 45°, and 60°. It can be seen that the moment does in fact maintain a constant amplitude despite the changing incident flow angle. It is interesting to examine the effect changing flow angles, as the designer of a flapping wing device in an outdoor environment would have to plan for sudden gusts of wind which would rapidly change the flow angle.

0°

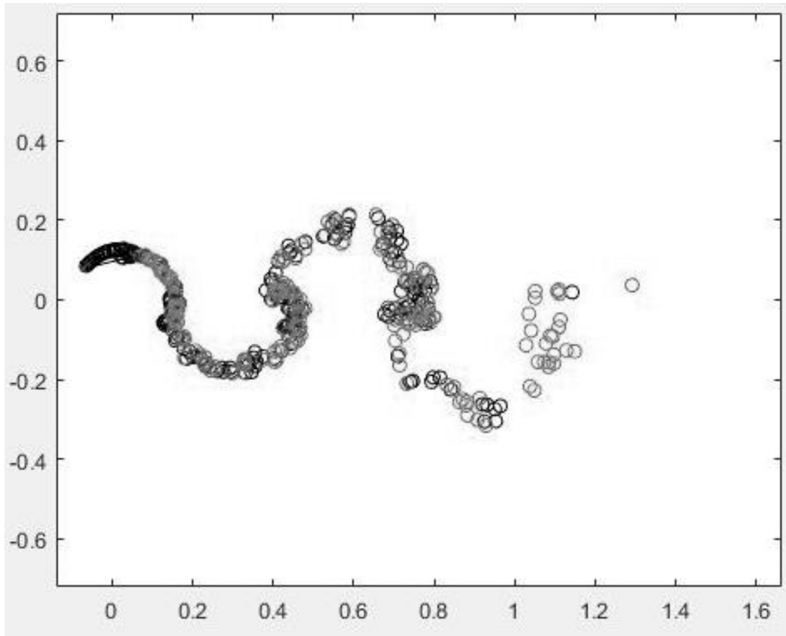


Figure 5.9a

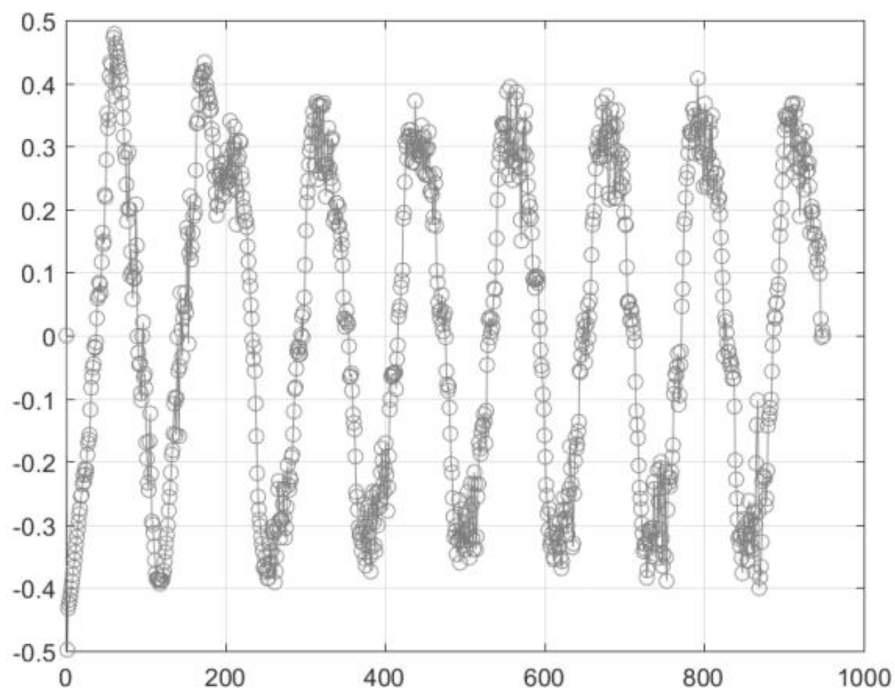


Figure 5.9b

45°

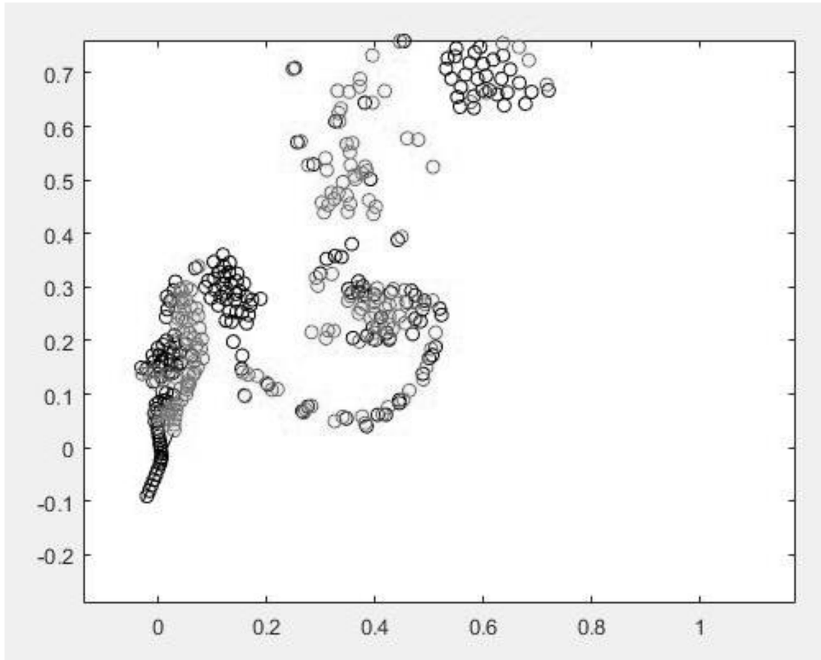


Figure 5.9c

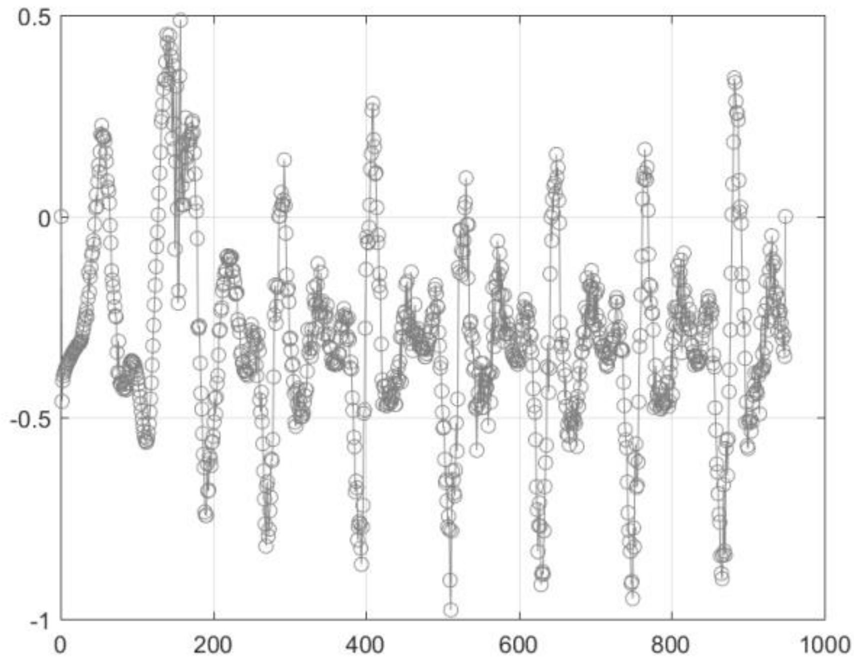


Figure 5.9d

60°

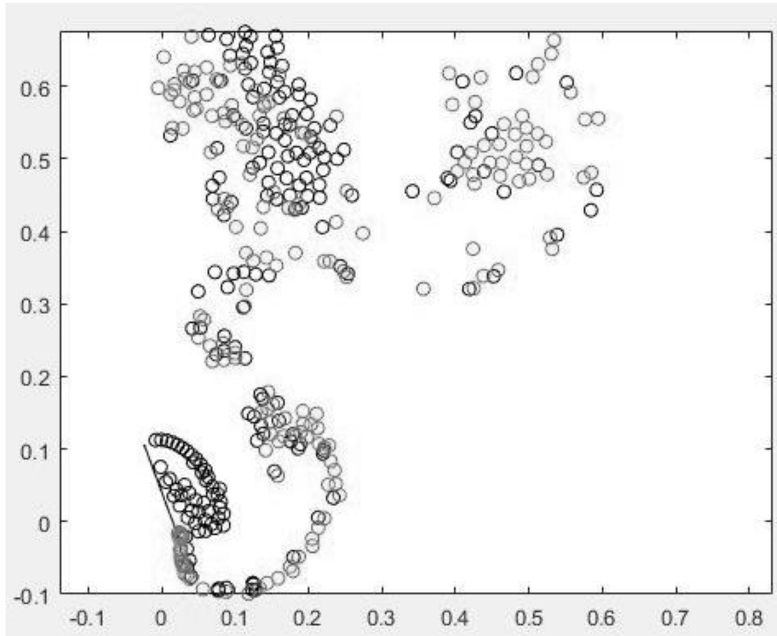


Figure 5.9e

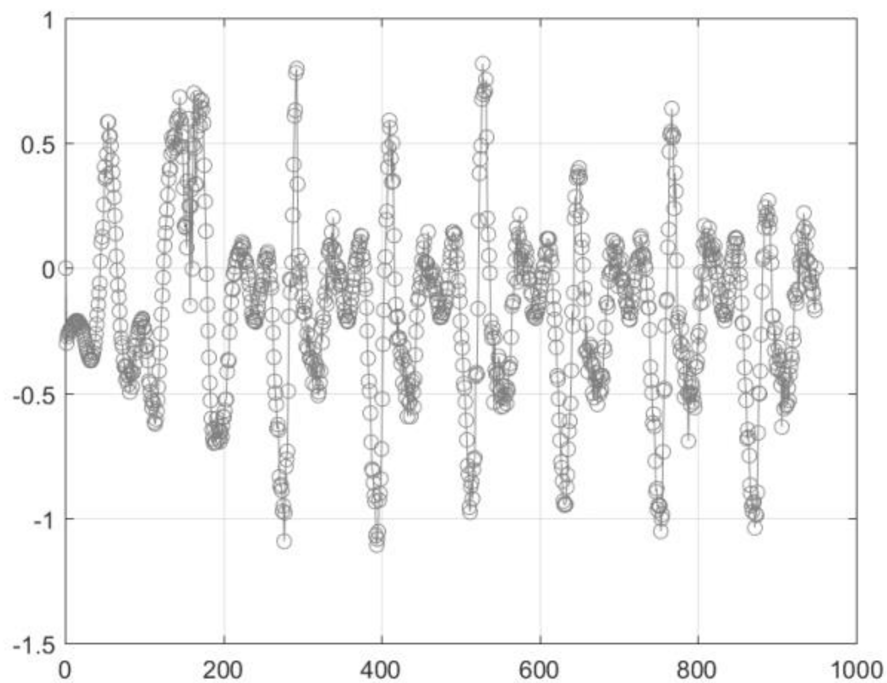


Figure 5.9f

Figure 5.9 – Effect of varying incident flow velocity angle on pitching moment amplitude

5.5 Conclusions

Fast, specialized CFD sovlers will be required to usher in a new age in aerospace technologies. Unsteady flow problems such as that of flapping wings require vast computational resources which make their solutions inacessible to the average designer.

A novel, highly efficeint impulse-momentum vortex method sovler has been developed and optimized using a massively parallel computing approach on an affordable workstation GPU. The solver has proven to be very accurate for short simulations, but the pitching moment tends to increase for simulations involving several flapping cycles.

In order to more accurately model the flow physics and maintain a constant moment amplitude for long simulations, a novel approach involving the lumping of far field point vortices allows for the inviscid solver to simulate viscous wake vortex decay.

In order to maintian solution accuracy, it was found that careful adjusting calculations had to be made to correct the linear and angular momenta. In addition, a lumped far field vortex model was adopted to eliminate unphysical high frequencies associated with the lumping process, thereby maintaining the circulation solution’s stability.

The lumped vortex model was shown to successfully maintain a constant moment, and was compared to the previous method as well as published data from a full unsteady Navier Stokes simulation for a flapping wing.

As an added benefit, the new method greatly reduced the computation time by reducing the proportionality of computation time from N^2 to N. Finally, the time threshold for lumping was determined to be when a given vortex had convected so far from the wing that its influence is less than 0.5% of the free stream velocity.

Several flow angles were tested to ensure the new method was sufficiently consistent in its ability to maintain a constant moment amplitude. The new method did in fact maintain the moment amplitude despite changing incident flow angles.

Chapter 6 | Fluid-Rigid Body Interaction

The preceding chapters demonstrate how the lumped wake vortex method produces a constant pitching moment amplitude and that the massively parallel implementation along with this novel method have increased the computational efficiency by several orders of magnitude. In light of these developments, the fluid solver will now be coupled to a rigid body solver in order to attain the ultimate goal of predicting the flight trajectory of a flapping wing MAV given a specific flapping motion. It is important to note that the heaving direction angle (stroke plane angle from [4]) remains fixed with respect to the ground regardless of the orientation of the MAV. In a real MAV, this could be accomplished using an on-board gyroscope.

6.1 Single Unconstrained Rigid Body Equations of Motion

At first, for simplicity, it was assumed that the MAV would comprise of a single unconstrained rigid body with the wings attached at the center of mass. In 2D, the rigid body equations of motion are independent of one another, as there is only one direction of rotation (i.e. the equations are not coupled as in Euler's Equations of Motion).

The 2D equations of motion are stated as follows:

$$F_x(t) = ma_x \, , \tag{6.1}$$

$$F_y(t) = ma_y \, , \tag{6.2}$$

$$M_z(t) = I_{zz}\alpha_z \, , \tag{6.3}$$

where $F_x(t)$ is the time-dependent forcing function in the x-direction found in the fluid solver, m is the mass of the MAV, a_x is the acceleration of the MAV in the x-direction, $F_y(t)$ is the time-dependent forcing function in the y-direction found in the fluid solver, a_y is the acceleration of the MAV in the y-direction, $M_z(t)$ is the time-dependent pitching moment function about the z-axis found in the fluid solver, I_{zz} is the mass moment of inertia of the MAV about the principal axis z, and α_z is the angular acceleration of the MAV about the z-axis.

6.2 Concurrent Solution Dilemma

In order to solve the governing equations of motion for a rigid body and to determine the position of the MAV as a function of time, the forces must be known a priori. However, the lumped vortex method utilizes the central difference method via the impulse-momentum equations for calculating the force and moment produced from flapping at each instant (Eqns. (2.10) and (2.11)). Since the central difference method requires future information to determine the present finite difference, the force and moment calculation used in the lumped vortex method requires the position of the MAV to be known a priori as well. This creates a chicken or the egg problem when one wants to create a time-marching solution which solves for the forces and position concurrently.

At first the solution to this dilemma seemed simple enough. The problem is that at the present time step, the equations of motion are solved using the forces found by the central difference method, but the information about the future value of the fluid momentum is unknown. Therefore, it seemed logical to switch to a backwards difference scheme for the force and moment calculation so that one would only need to know the past positions of the MAV in order to determine the force and moment acting on the MAV at the present time.

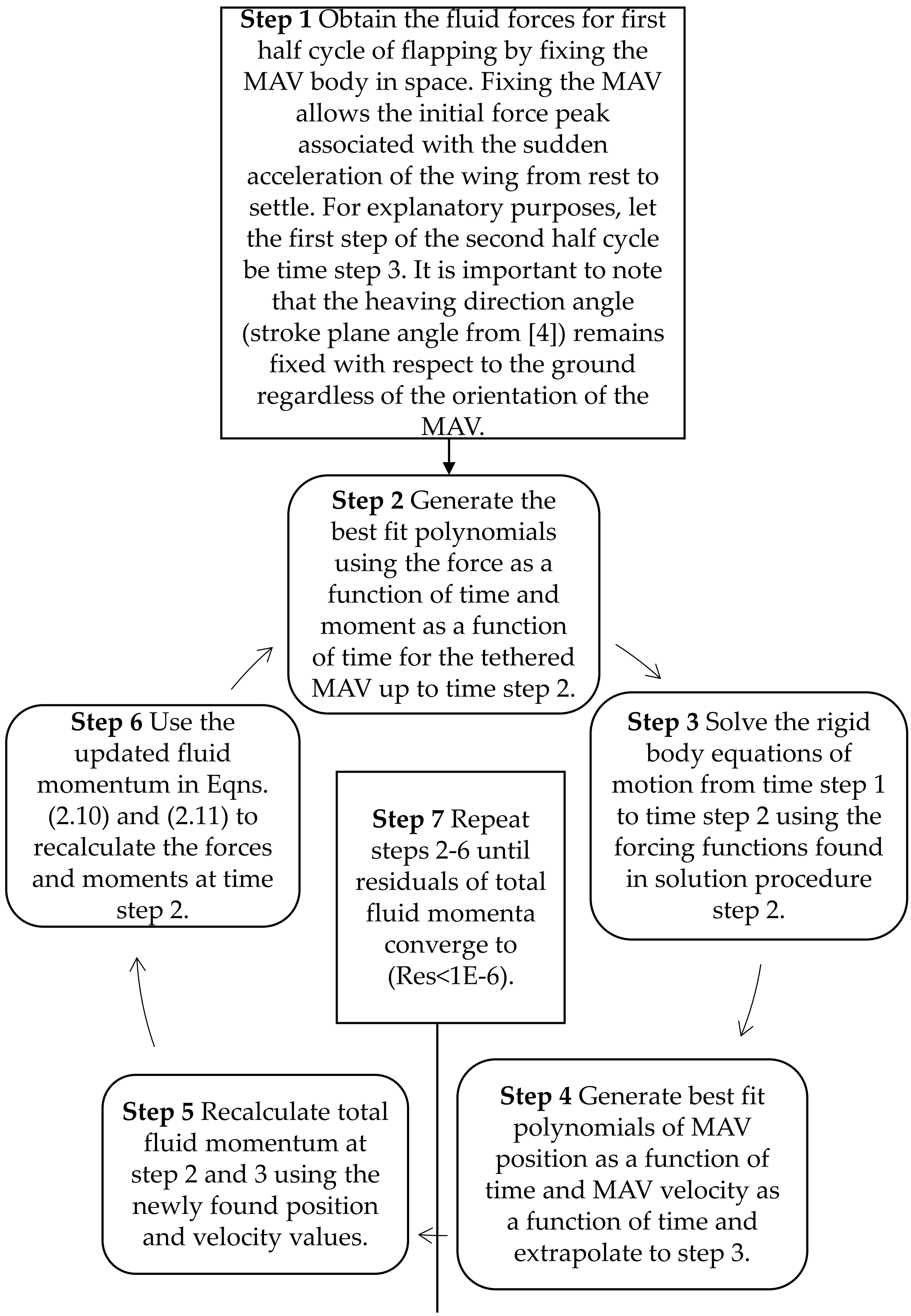
However, no matter which order backwards difference scheme was used (up to 4th order), the solution of the force and moment varied greatly from the central difference solution. To make matters worse, even if the force at the current time step could be determined, one would not know how the force changed as the solid body equations of motion solver progressed from step n to n+1. This would be equivalent to a constant acceleration model, which would require relatively small time steps to produce a sufficiently accurate solution.

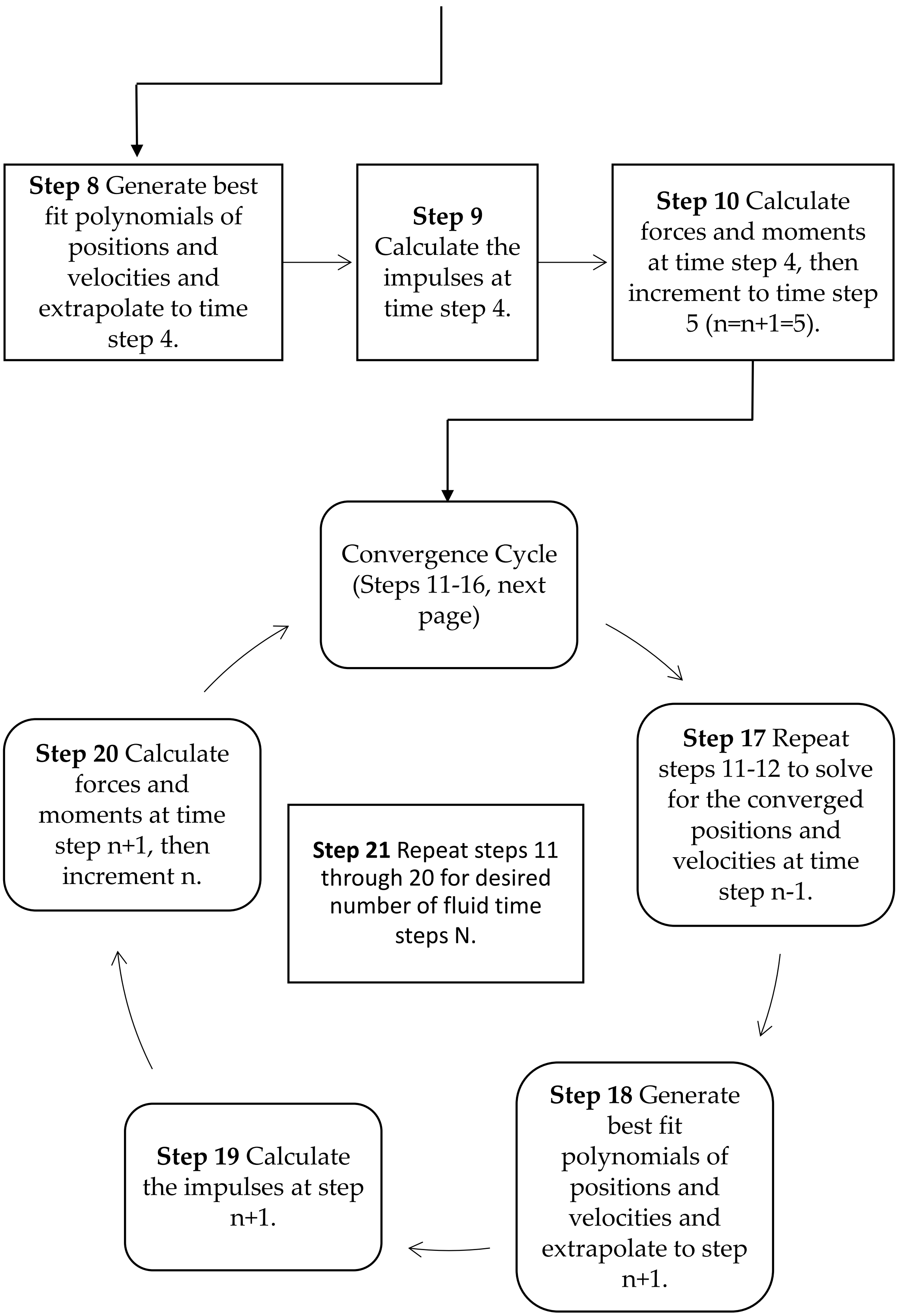
Consequently, the backwards difference method was abandoned in search of a different way to remedy the problem. Several new strategies involving the central difference method and the equations of motion were attempted, albeit unsuccessfully.

6.3 The Kinetic Predictor-Fluid Corrector Algorithm

Finally, it was realized that the solution of the fluid forces had much more substantial high frequency components than the solutions to the position and velocity of the MAV would have. This means that any extrapolation of the fluid forces would likely be very inaccurate while extrapolations of the position and velocity of the MAV would be likely to be much more accurate. This realization led to the Kinetic Predictor-Fluid Corrector Algorithm, where the fluid forces are only solved for up until step n-1, but the position and velocities are extrapolated up until time step n+1. Of all of the methods tested, this one gave the fastest convergence and the largest acceptable fluid time step (i.e. without the solution diverging). When the solution converges, this means that the position, velocity, fluid force, and fluid moment remain sufficiently unchanged from one convergence cycle to the next, and the solution to the rigid body equations of motion are such that the resulting fluid momenta is predicted with less than 0.0001% error. The converge-detecting while loop contains a counter to report any divergent time steps, which are defined as time steps with over 100 convergence cycles. If a divergent time step is detected, it breaks out of the loop and begins solving the next time step, so that one doesn't lose the results of the entire simulation for only one or two divergent time steps. Fully convergent solutions are those for which no divergent time steps have been detected, and these are the only solutions used in the present work.

The solution procedure is shown in the flowchart in Figure 6.1:




Step 8 Generate best fit polynomials of positions and velocities and extrapolate to time step 4.
Step 9 Calculate the impulses at time step 4.
Step 10 Calculate forces and moments at time step 4, then increment to time step 5 (n=n+1=5).
Convergence Cycle (Steps 11-16, next page)
Step 17 Repeat steps 11-12 to solve for the converged positions and velocities at time step n-1.
Step 18 Generate best fit polynomials of positions and velocities and extrapolate to step n+1.
Step 19 Calculate the impulses at step n+1.
Step 20 Calculate forces and moments at time step n+1, then increment n.
Step 21 Repeat steps 11 through 20 for desired number of fluid time steps N.

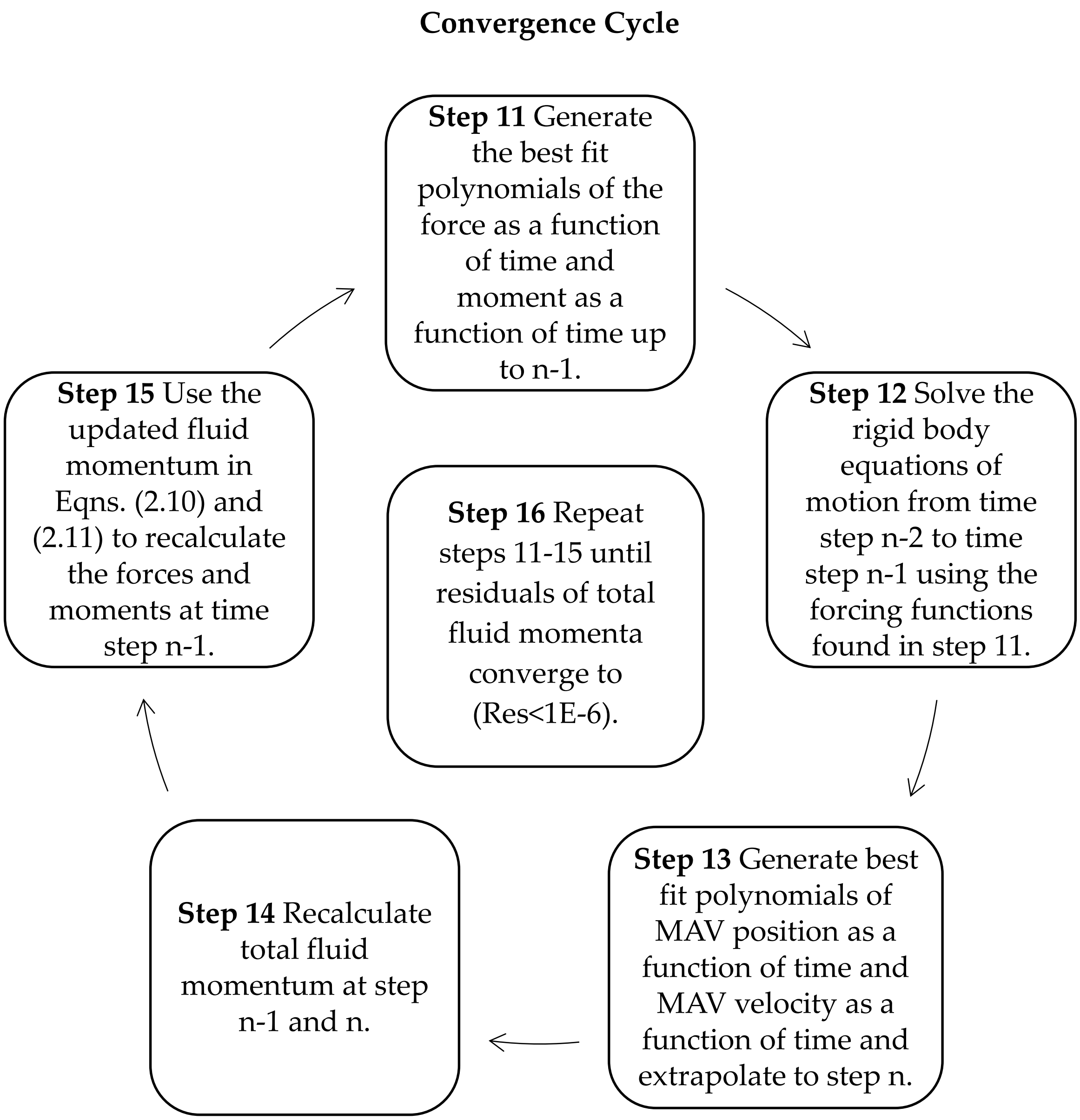


Figure 6.1 - Flow chart of single rigid body solution procedure

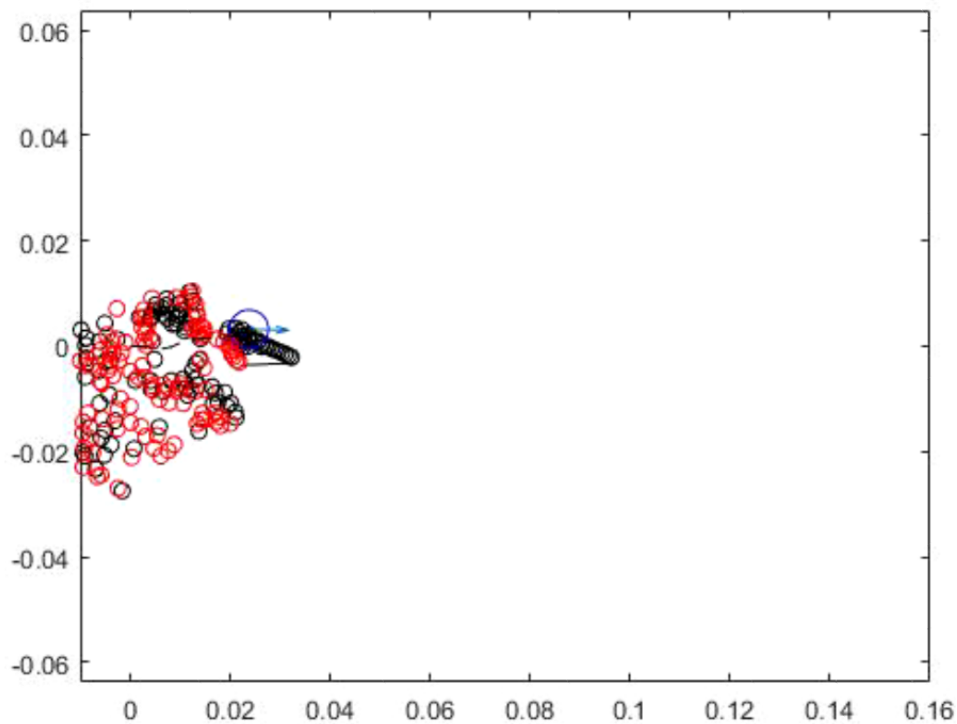


Figure 6.2a – End of second cycle

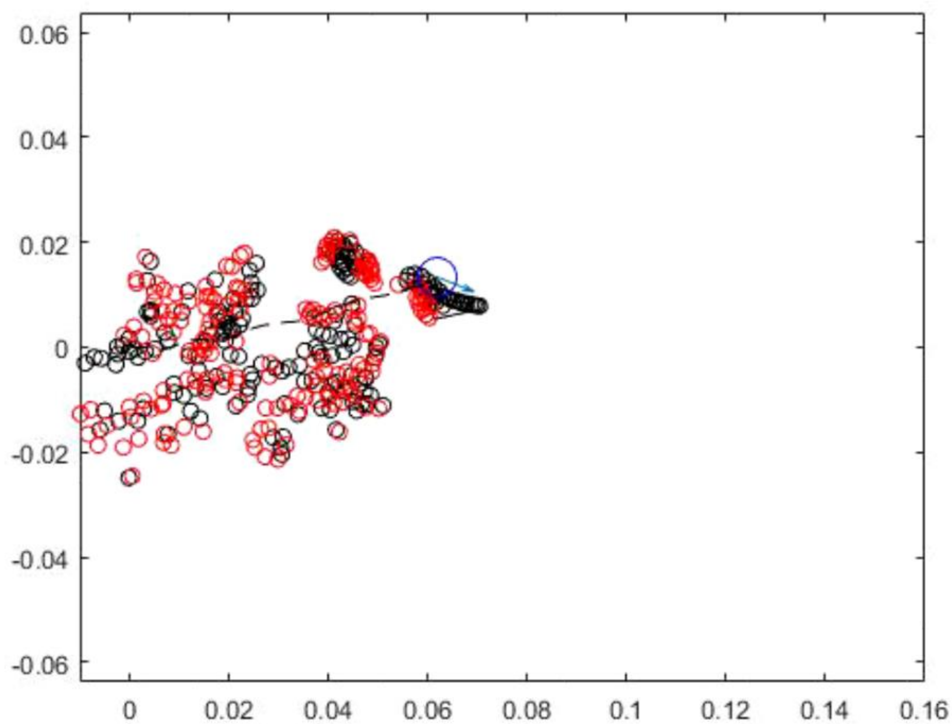


Figure 6.2b – End of fourth stroke

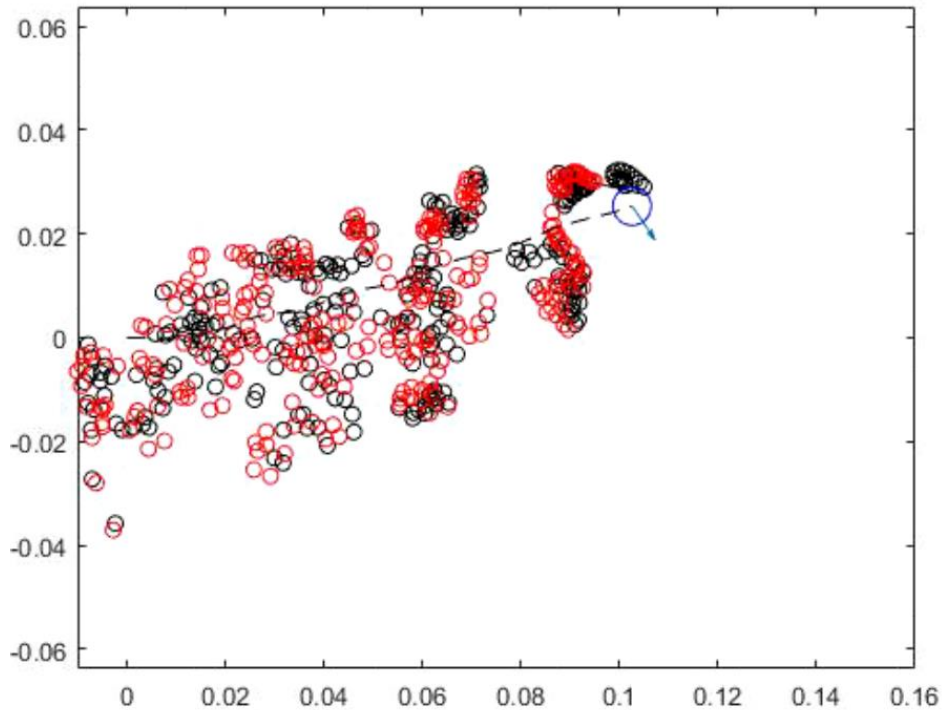


Figure 6.2c – Middle of sixth stroke

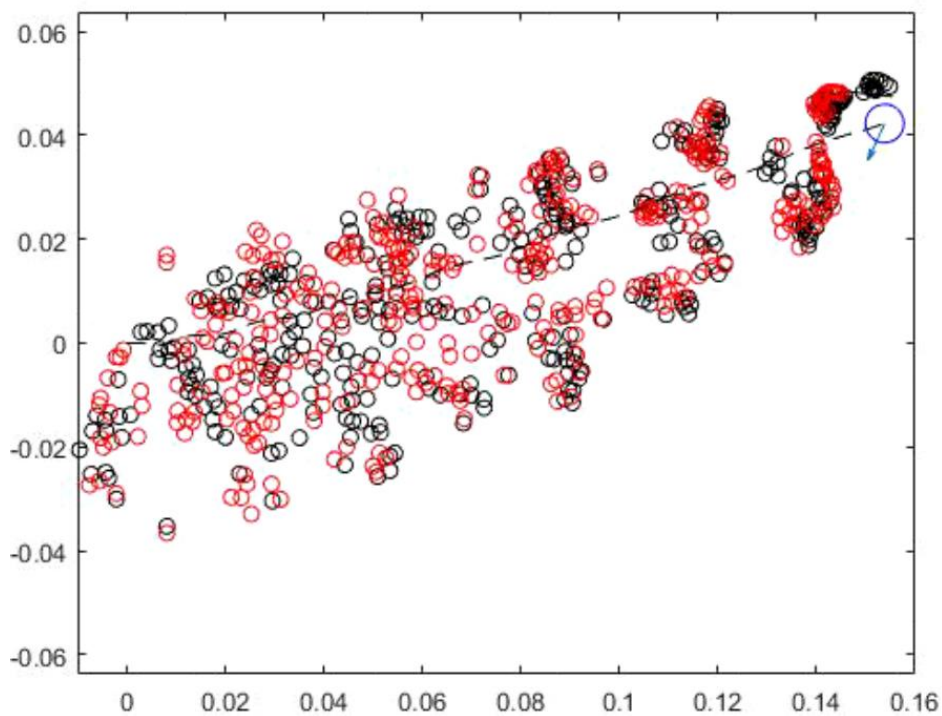


Figure 6.2d – Middle of seventh stroke

Figure 6.2 – Fluid-Single Body Interaction Wake: Black and red circles indicate vortices shed from leading and trailing edges, respectively, dotted line indicates flight path, blue circle indicates MAV position, arrow indicates MAV orientation

6.4 Results

An example of the time evolution of the wake vortex output plot of The Kinetic Predictor-Fluid Corrector Algorithm is shown in Figure 6.2. The black and red circles indicate vortices shed from leading and trailing edges, respectively, the black dotted line indicates the MAV's calculated flight path, the large blue circle indicates the current MAV position, the blue arrow indicates the current calculated MAV orientation. As can be seen from the figure, the MAV body moves in a slight zig-zag path due to the cyclical flapping forces as expected.

The fluid-rigid body solution converges based on total fluid momentum for time steps under a certain threshold which is dependent on the input flapping kinematic parameters. As expected, the solution converges more quickly for smaller time steps than for larger ones. For example, a simulation with a nondimensional time step of 0.0325 periods converges in about 8 convergence cycles and a simulation with a nondimensional time step of 0.0163 periods converges in about 3 convergence cycles. Given that the solution procedure for the equations of motion of a single a 2D planar rigid body is relatively straightforward and well established, the solution can be assumed to be correct. However, in the future and for more complex 3D or multibody simulations, it would be prudent to check the solution for energy and momentum conservation.

The convergence cycle shown in Figure 6.1 is carried out at each time step and consists of a while loop whose exit criteria is defined as when the total fluid momentum remains constant from one cycle to the next and has thus converged. The residuals of the total fluid momenta, which define how much fluid momenta have changed from the previous iteration of the convergence cycle to the present one, are calculated as follows:

$$Res_L = \frac{|L_{total}^{m-1} - L_{total}^{m}|}{|L_{total}^{m-1}|} \ , \tag{6.4}$$

$$Res_H = \frac{|H_{total}^{m-1} - H_{total}^{m}|}{|H_{total}^{m-1}|} \ , \tag{6.5}$$

where Res_L and Res_H are the residuals of the total linear and angular fluid momenta, respectively, L_{total}^{m} and H_{total}^{m} are the total linear and angular fluid momenta calculated for iteration m of the convergence loop, respectively.

Depending on the input parameters, the residuals of the solution of the total fluid momenta at a given time step will usually converge after less than eight predictor-corrector iterations down to levels as low as 1E-6. The time increment was determined by conducting a time step size convergence study which examined the accuracy of the solution as the time step was reduced. The nondimensional time step which solved sufficiently fast and with sufficient accuracy was determined to be 0.0325 periods (which corresponds to m=25 from [4]).

By observing the orientation of the MAV (represented by the arrow) from Figure 6.2a to Figure 6.2d, one may note that the body of the MAV tends to rotate clockwise, which must be due to an unbalanced nose-down moment. Upon observation of the plot of the moment vs. time (Figure 6.3), one may indeed confirm that on average the moment acting on the body of the MAV is clockwise. Since the MAV will likely have sensing equipment (e.g. a camera), this nose down rotation is undesirable, as it would make achieving simple tasks such as locating and avoiding an obstacle very difficult.

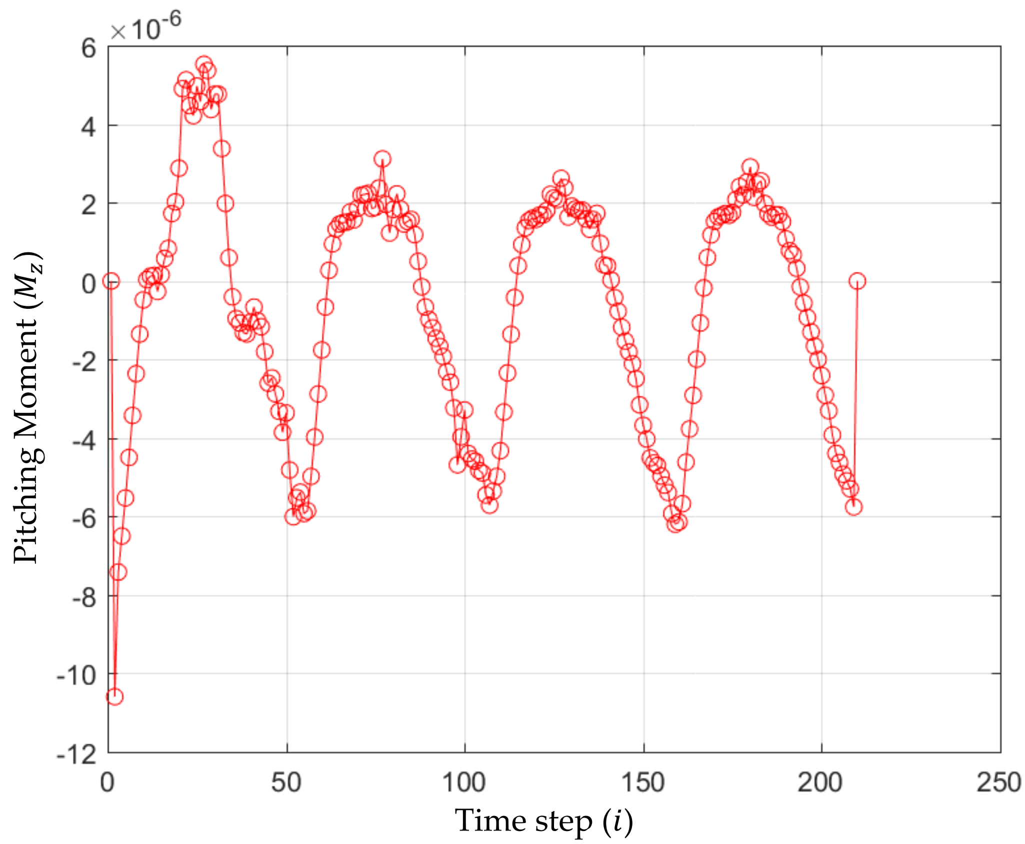


Figure 6.3 - Pitching moment vs. time

In order to balance this nose down moment, one may simply move the center of mass towards the rear of the MAV with respect to the wings, such that the lift force produced by the wing in combination with gravity acting at this center of ass would produce a force couple in forward flight which is exactly equal and opposite to the unbalanced moment generated from flapping, thereby balancing the MAV body.

However, this would not be sufficient for all flight conditions as it is further purported the average value of this unbalanced moment would change based on any variation in fight speed, flapping kinematics, or environmental conditions. Referring to Figure 5.9, one may note that changes in wind direction (or equivalently, the MAV's flight direction) would indeed change the average pitching moment.

For a solution to this control issue, one may turn to natural fliers for inspiration. It is well known that most, if not all, natural flapping wing animals and insects do not simply have a single rigid body, but a number of bodies which can be oriented with respect to one another using muscles. A simple example is a two-body flapping wing insect, whose body consists of a thorax, which contains the head and is where the wing attaches, and an abdomen. In natural flight, one may observe the insect changing the angle between the thorax and abdomen. It is thought that by adjusting the angle between the thorax and abdomen, the insect is carefully balancing the pitching moment created from flapping in order to keep the thorax (and consequently the head and eyes) steady during flight.

Therefore, in the next chapter the solver will be expanded to allow for the solution of two rigid bodies connected by a pin joint and actuated with respect to one another by a rotary actuator. Subsequently, the control of this actuator and the effect of the additional body will be studied. The results are used to test the hypothesis that the abdomen of a two-body flapping wing insect is used to actively stabilize the thorax in order to allow the insect to more easily identify obstacles and targets (e.g. food).

Chapter 7 | The Two-Body Problem

In the previous chapter the Kinetic Predictor-Fluid Corrector Algorithm was introduced and applied to a single 2D rigid body MAV model. During upward-forward flight, the MAV model tended to pitch nose down due to an unbalanced clockwise moment. Taking inspiration from nature, it was suggested that this unbalanced moment could be counteracted and the first rigid body be stabilized by introducing a second movable body attached to the first.

Now that the fluid-rigid body solution procedure has been successfully implemented for the basic single-body case and the problematic and unpredictable pitching moment is causing an undesirable rotation, the focus of the present study is shifted toward solving the equations of motion for a two-body system whose bodies are connected by a pin joint and adjusted with respect to one another using a rotary actuator.

7.1 Coordinate Systems

The coordinate systems and fluid force (F) and moment (M) are illustrated in Figure 7.1.

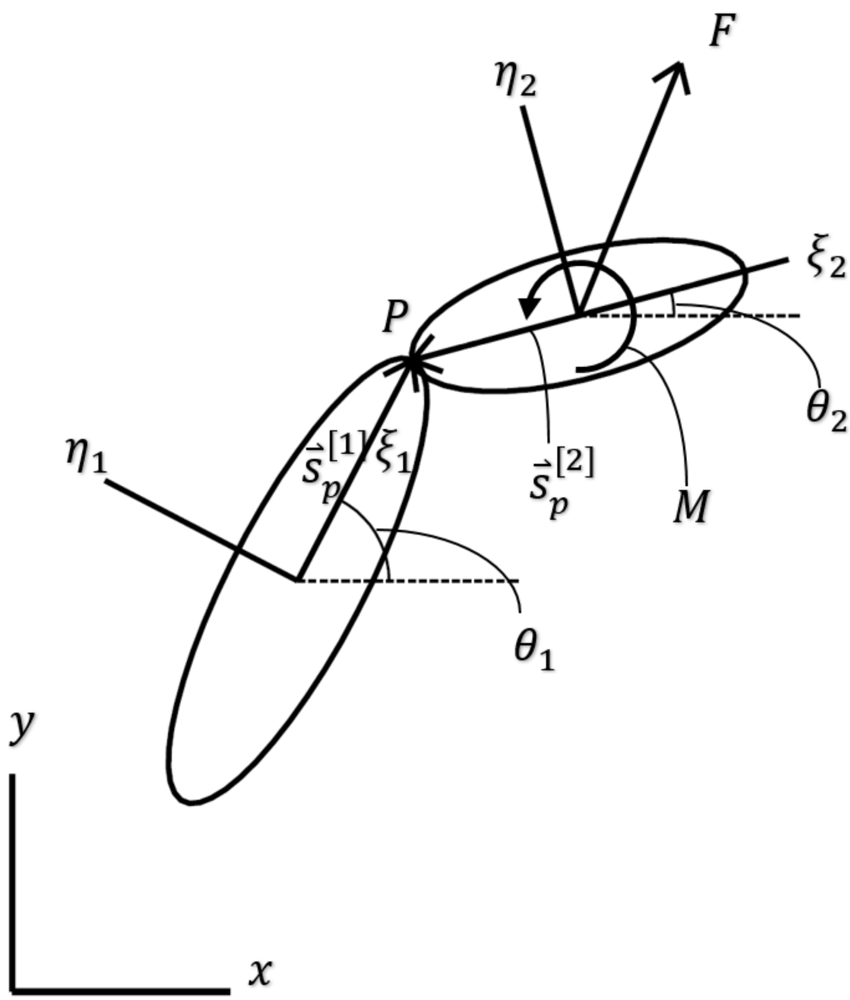


Figure 7.1 - Coordinate systems and free body diagram of two-body MAV

7.2 The Lagrange Multiplier Formulation

There are several well established solution methods for solving a multi-body dynamics problems [16]. One method would be to eliminate the reaction forces from the equations of motion which in the case of two bodies connected by a pin joint would produce four variables and four equations. However, these equations would be nonlinear and thus the solution method would become fairly complex and convergence would be slow. Therefore, the first approach to be demonstrated is the Lagrange Multiplier formulation.

Referring to Figure 7.1, the following constraint equations are defined:

$$\vec{\Phi} = \vec{r}_1 + {}_0\boldsymbol{A}_1\, \vec{s}_p^{\,[1]} - \vec{r}_2 - {}_0\boldsymbol{A}_2\, \vec{s}_p^{\,[2]} = \vec{0}\ , \tag{7.1}$$

where $\vec{r}_1$ and $\vec{r}_2$ are the position vectors of bodies 1 and 2 in the space fixed frame, respectively, ${}_0\boldsymbol{A}_1$ and ${}_0\boldsymbol{A}_2$ are the transformation matrices from the space fixed system to systems 1 and 2, respectively, and $\vec{s}_p^{\,[1]}$ and $\vec{s}_p^{\,[2]}$ are the positions of the pin joint in systems 1 and 2, respectively. These equations are the x and y loop equations which are to be satisfied at all instants in order to ensure that the bodies remain attached at the pin joint.

Next, the coordinates of the two bodies are defined by the following vector:

$$\vec{q} = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \\ x_2 \\ y_2 \\ \theta_2 \end{bmatrix}. \tag{7.2}$$

The derivate of the constraint equations with respect to the coordinates can then be found to be

$$\vec{\Phi}_q = \begin{bmatrix} 1 & 0 & -y_{p/1}^{[O]} & -1 & 0 & y_{p/2}^{[O]} \\ 0 & 1 & x_{p/2}^{[O]} & 0 & -1 & -x_{p/2}^{[O]} \end{bmatrix}, \tag{7.3}$$

where $x_{p/2}^{[O]}$ is read as "the x-value of the vector from the origin of system 2 to point P in the expressed in the space fixed coordinate frame".

Next, the mass matrix is defined by the following diagonal matrix:

$$\boldsymbol{M} = \begin{bmatrix} m_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_2 \end{bmatrix}, \tag{7.4}$$

where m_1 and m_2 are the masses of the first and second body, respectively, and I_1 and I_2 are the mass moments of inertia of the first and second body, respectively. The first and second bodies are defined as the abdomen and thorax, respectively.

The corresponding acceleration vector is as follows:

$$\ddot{\vec{q}} = \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{\theta}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \\ \ddot{\theta}_2 \end{bmatrix} \tag{7.5}$$

where $\ddot{x}_i$, $\ddot{y}_i$, and $\ddot{\theta}_i$ are the linear accelerations in the x and y directions and the angular acceleration in the θ-direction, respectively, for the i-th body (i = *1,2).*

The force vector is defined as follows:

$$\vec{g} = \begin{bmatrix} F_{x1} \\ F_{y1} \\ M_{z1} \\ F_{x2} \\ F_{y2} \\ M_{z2} \end{bmatrix}, \tag{7.6}$$

where F_{xi}, F_{yi}, and M_{zi} are the sum of the forces in the x and y-directions and the sum of the moments about z, respectively, for the i-th body (i = *1,2).*

Next, the pseudo-forces derived from the quadratic angular velocity terms of the equations of motion are defined as follows:

$$\vec{\gamma} = \begin{bmatrix} x_{p/1}^{[O]}\,\dot{\theta}_1^2 - x_{p/2}^{[O]}\,\dot{\theta}_2^2 \\ y_{p/1}^{[O]}\,\dot{\theta}_1^2 - y_{p/2}^{[O]}\,\dot{\theta}_2^2 \end{bmatrix} \;, \tag{7.7}$$

and the Lagrange multipliers which represent the coefficients used to express the constraint reaction forces as a linear combination of the derivative of the constraint equations with respect to the coordinate vector are defined as follows:

$$\vec{g}_C = \overrightarrow{\Phi}_q^T\,\vec{\lambda} \;, \tag{7.8}$$

where $\vec{g}_C$ is the reaction force vector and superscript T indicates the transpose of a vector.

Finally, putting it all together the equations of motion for the two-body pin-connected system are expressed as follows:

$$\begin{bmatrix} \boldsymbol{M} & \overrightarrow{\Phi}_q^T \\ \overrightarrow{\Phi}_q & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\vec{q}} \\ -\vec{\lambda} \end{bmatrix} = \begin{bmatrix} \vec{g} \\ \vec{\gamma} \end{bmatrix} \tag{7.9}$$

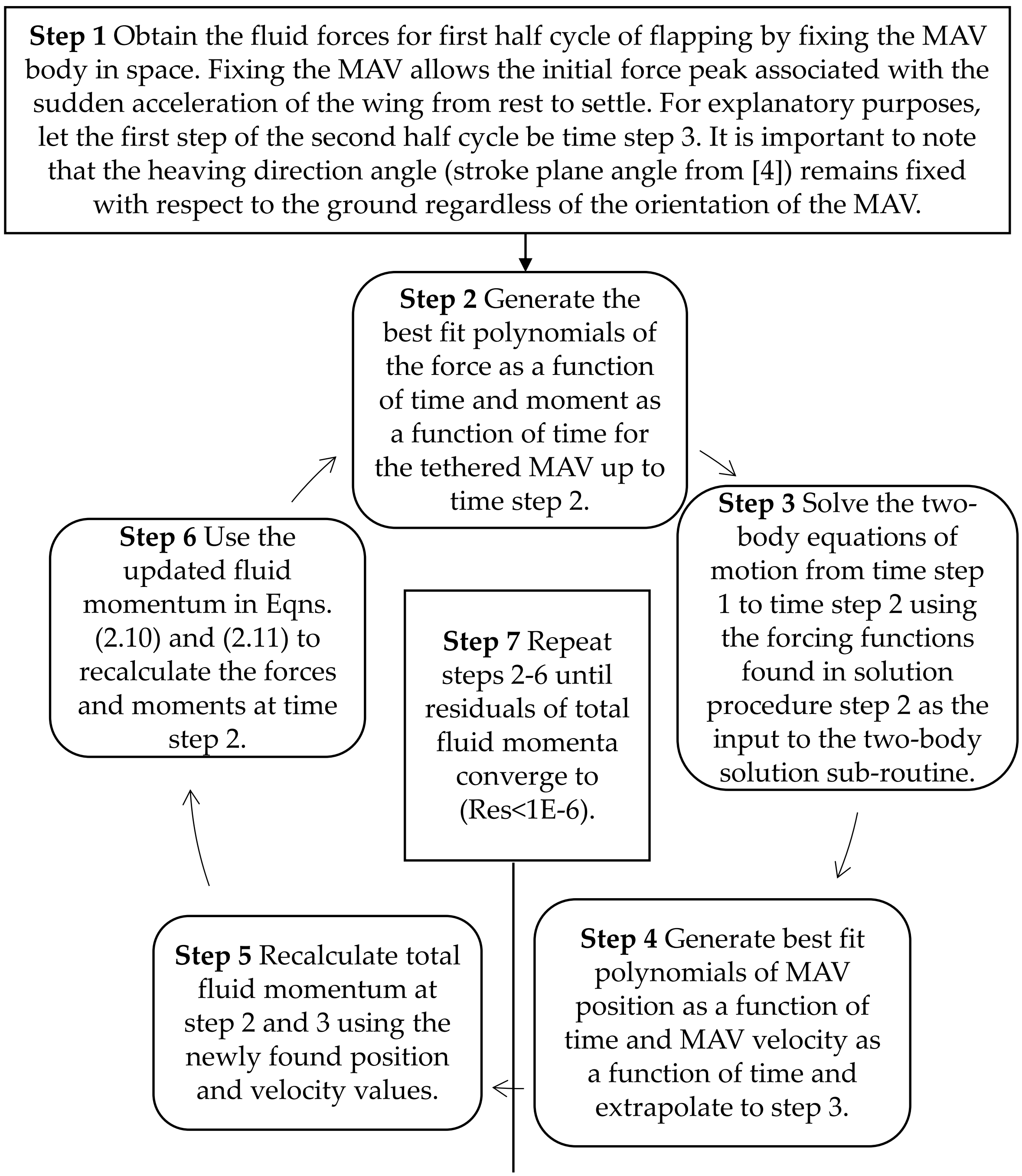
Each time the ODE subroutine is called, this linear system is solved for the accelerations and Lagrange multipliers and the accelerations are used as part of the system of 12 first order ODEs defined as follows:

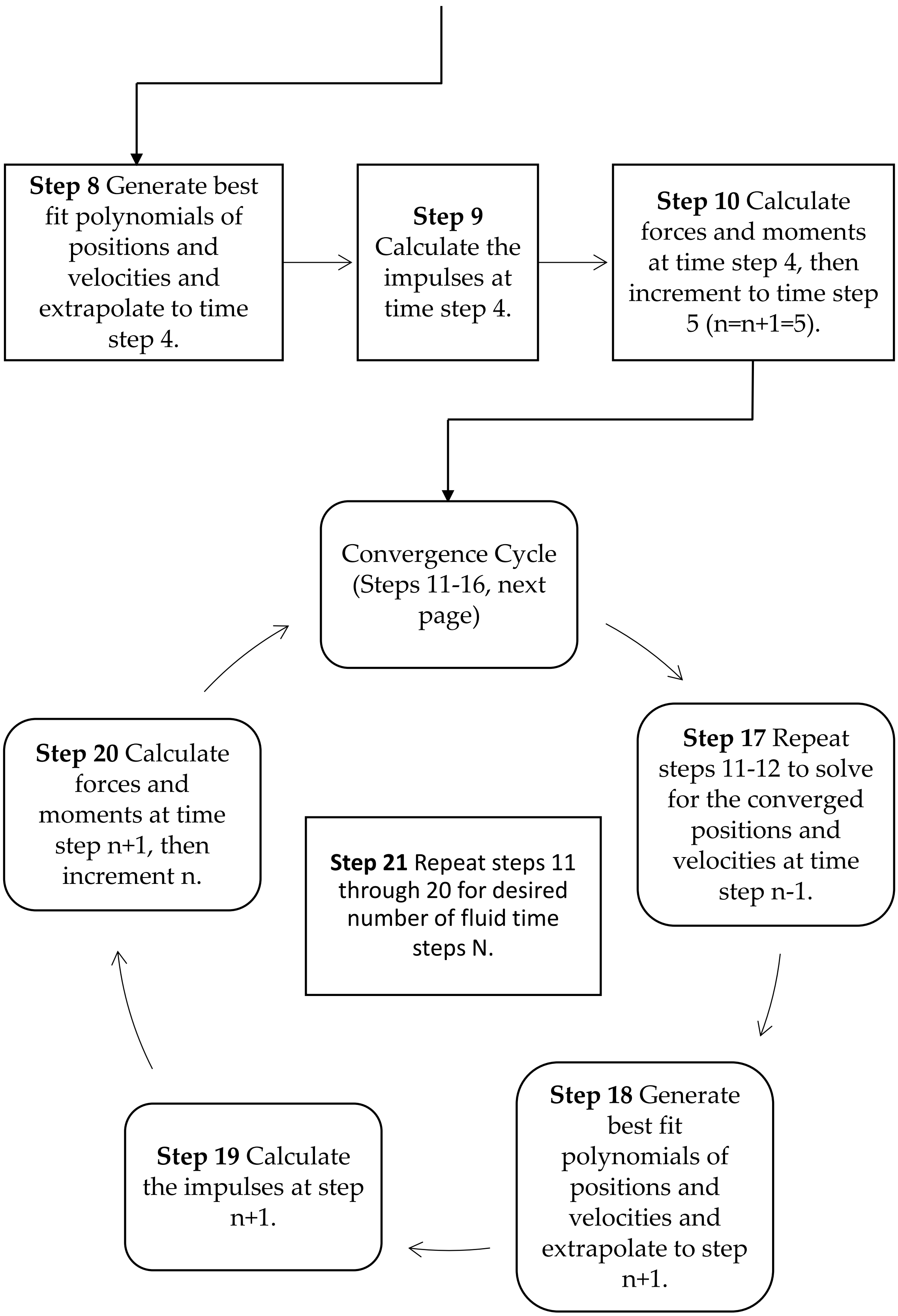
$$\frac{d}{dt}\begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \\ x_2 \\ y_2 \\ \theta_2 \\ u_1 \\ v_1 \\ \omega_1 \\ u_2 \\ v_2 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ \omega_1 \\ u_2 \\ v_2 \\ \omega_2 \\ \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{\theta}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \\ \ddot{\theta}_2 \end{bmatrix}, \tag{7.10}$$

where u_i, v_i, and ω_i are the x and y-direction linear accelerations and θ-direction angular acceleration, respectively, for the i-th body (*i = 1,2)*.

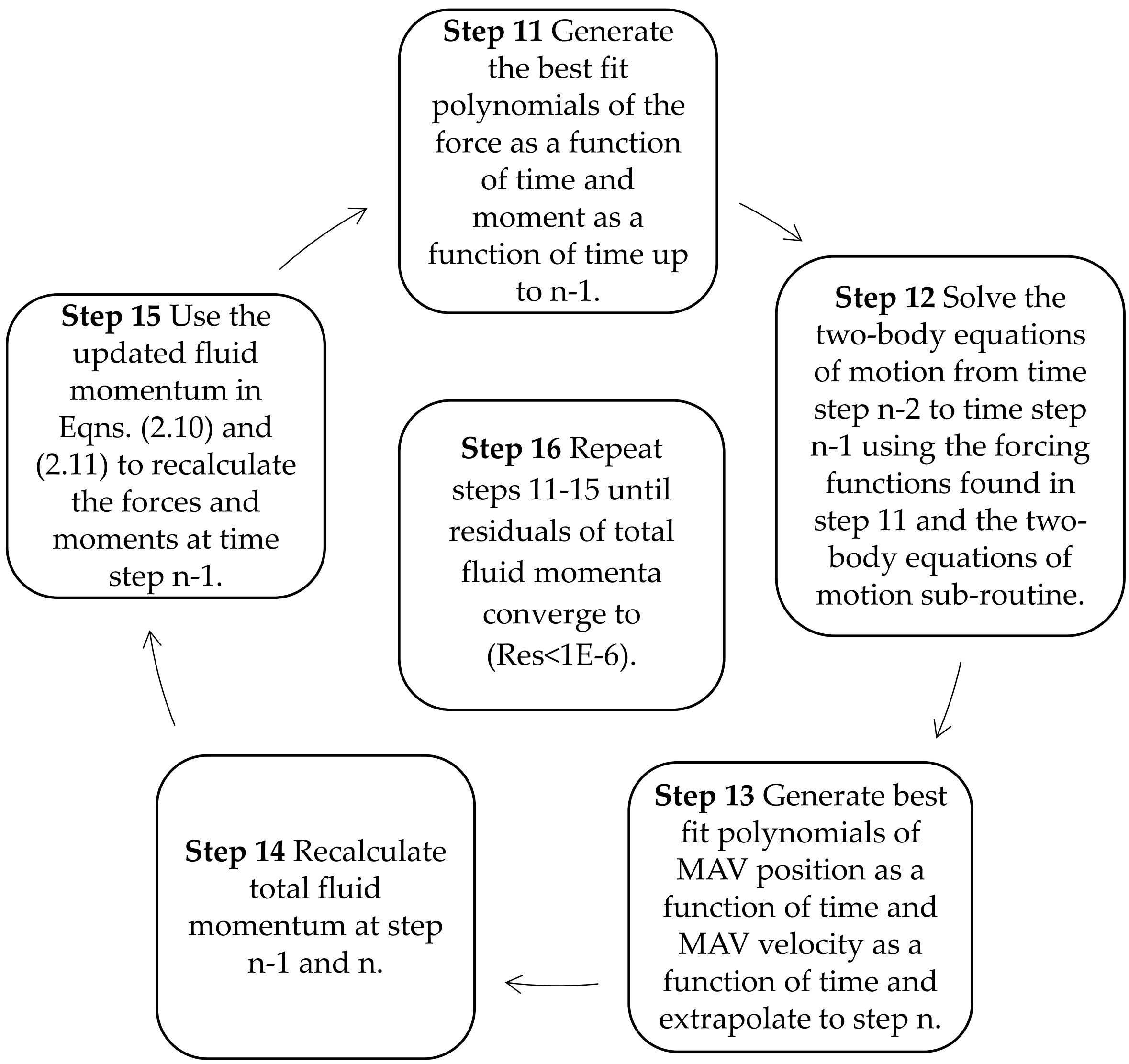
7.3 The Two-Body Solution Procedure

The solution procedure is shown in the flowchart in Figure 7.2:




Step 8 Generate best fit polynomials of positions and velocities and extrapolate to time step 4.
Step 9 Calculate the impulses at time step 4.
Step 10 Calculate forces and moments at time step 4, then increment to time step 5 (n=n+1=5).
Convergence Cycle (Steps 11-16, next page)
Step 17 Repeat steps 11-12 to solve for the converged positions and velocities at time step n-1.
Step 18 Generate best fit polynomials of positions and velocities and extrapolate to step n+1.
Step 19 Calculate the impulses at step n+1.
Step 20 Calculate forces and moments at time step n+1, then increment n.
Step 21 Repeat steps 11 through 20 for desired number of fluid time steps N.

Convergence Cycle



Two-Body Equations of Motion Sub-Routine

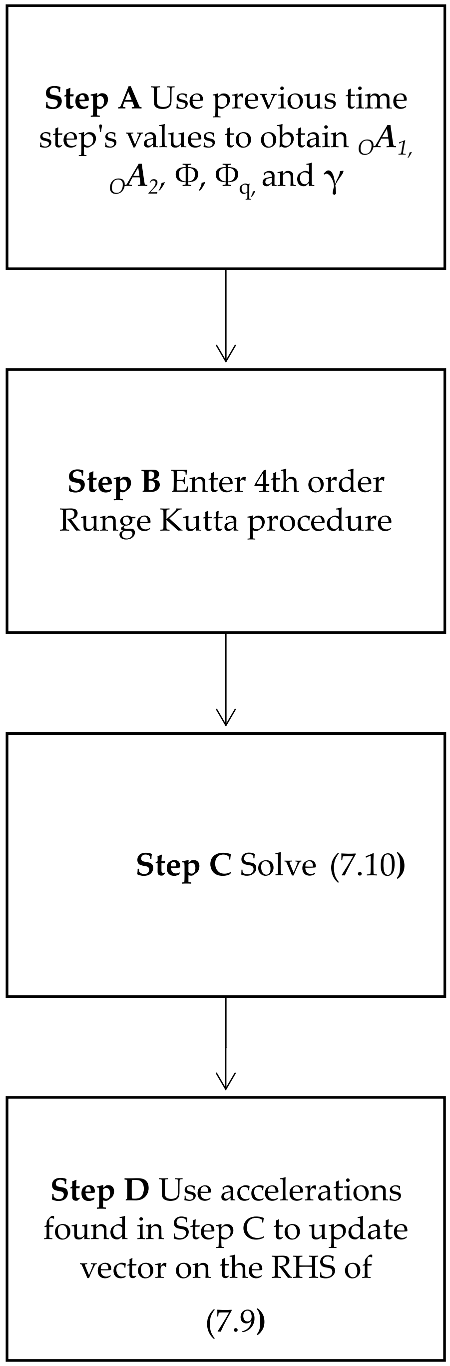


Figure 7.2 -- Flow chart of solution procedure for two-body problem

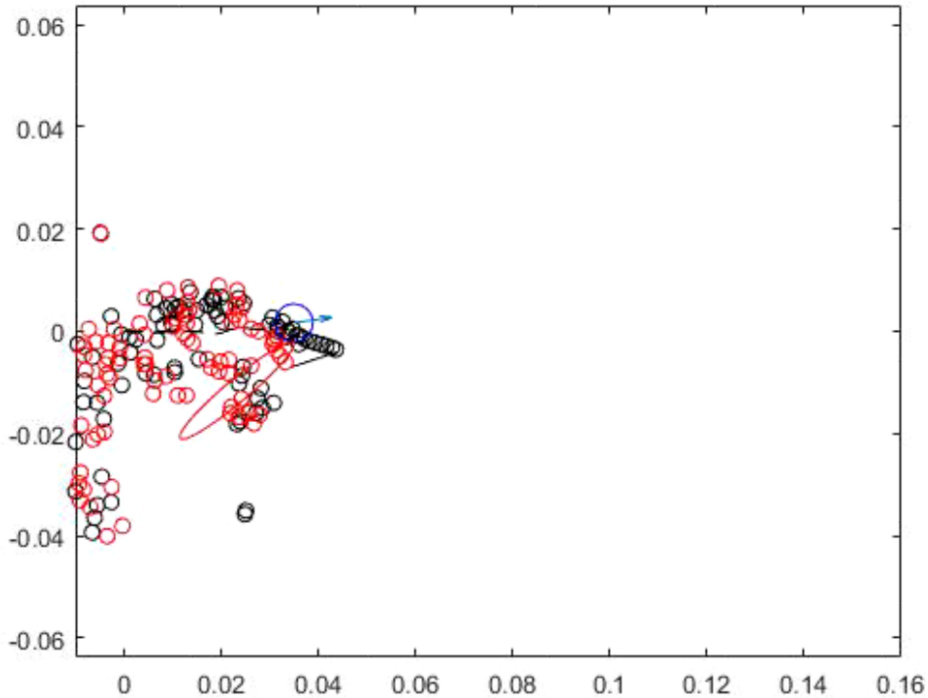


Figure 7.3a

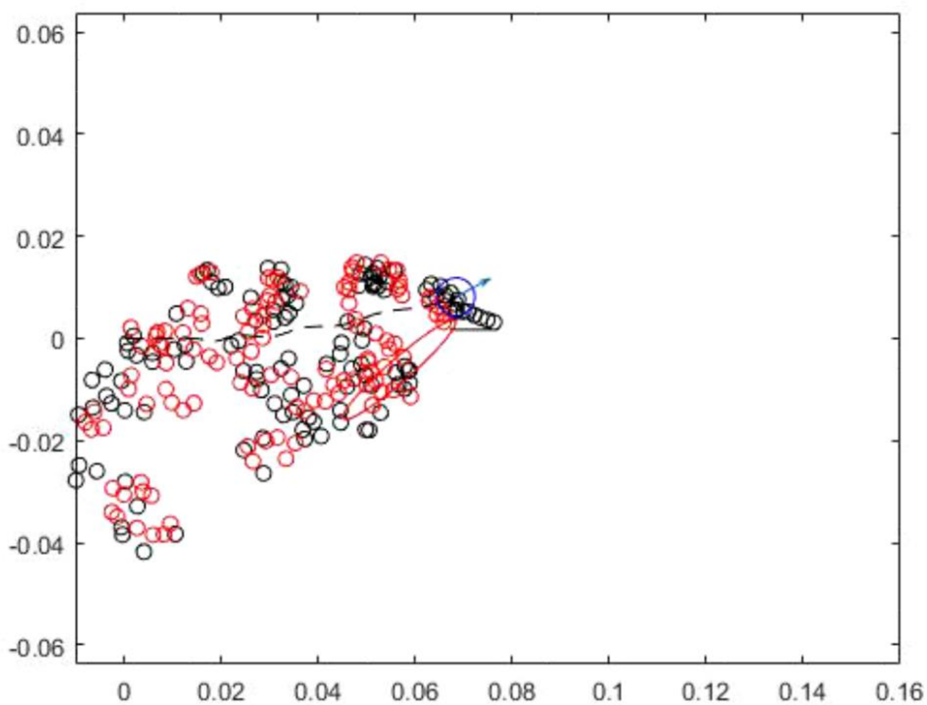


Figure 7.3b

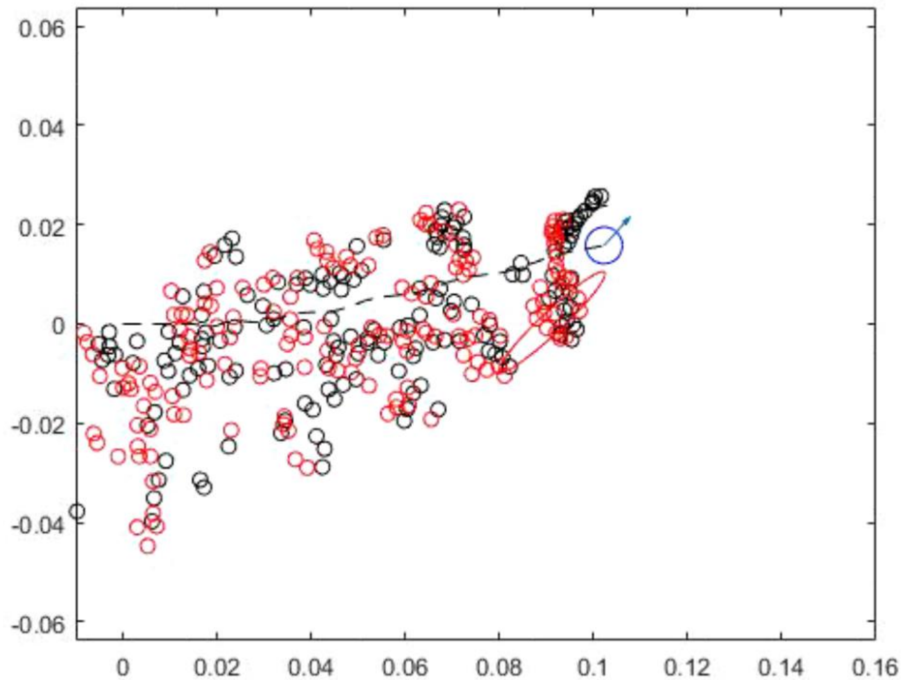


Figure 7.3c

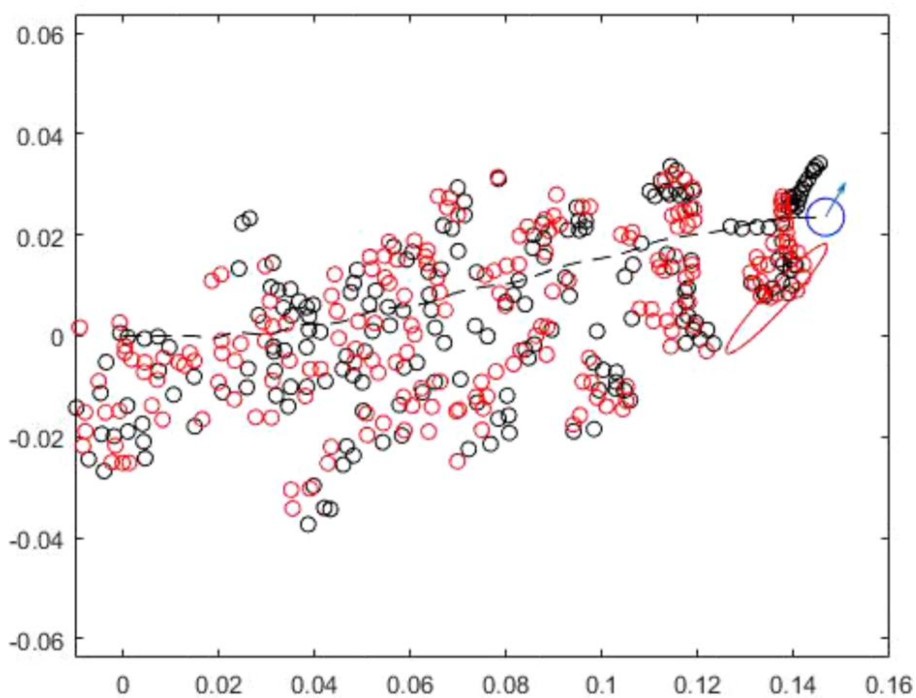


Figure 7.3d

Figure 7.3 – Two-body Lagrange multiplier formulation wake

7.3 Lagrange Multiplier Formulation Results

Figure 7.3 illustrates an example of the flight path and wake evolution output of the fluid-rigid body interaction simulation for two pin-connected bodies solved using the Lagrange multiplier formulation. As can be seen from the figure, the abdomen causes the thorax to rotate counterclockwise.

This result is due to the approximate angle and supporting torque applied at the pin joint to maintain the angle. However, this result is encouraging because the nose-down pitching has been eliminated and thus a method for stabilizing the MAV using the torque applied between the two bodies should be possible.

Upon further examination, however, one may note that the two bodies in Figure 7.3 separate as the solution progresses, thereby indicating that the constraint loop equations (7.1) have been violated. This is a common problem with the Lagrange multiplier formulation of multi-body dynamics. Therefore, despite the advantage of solving the system of equations of motion with a constant mass matrix, the violation of the constraint equations is unacceptable, since this cannot be an accurate physical representation of the problem.

There are several commonly used methods which can alleviate this issue, several of which are described in [16]. The first attempt was to implement the "Constraint Violation Stabilization Method". This method uses tuning variables α and β to attempt to adjust $\boldsymbol{\gamma}$ using the following system of equations:

$$\begin{bmatrix} \boldsymbol{M} & \vec{\Phi}_q^T \\ \vec{\Phi}_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\vec{q}} \\ -\vec{\lambda} \end{bmatrix} = \begin{bmatrix} \vec{g} \\ \vec{\gamma} - 2\alpha\dot{\vec{\Phi}} - \beta^2\vec{\Phi} \end{bmatrix} . \tag{7.11}$$

While this method did improve the solution considerably, the two bodies still tended to separate given certain input parameters. Several combinations of values for α and β were used but none of them adequately remedied the problem. Therefore, it was decided that the full nonlinear set of the equations of motion would be solved using MATLAB's built-in nonlinear system solver "fsolve", which is based on the Levenberg-Marquardt trust-region method from [17].

7.4 Nonlinear Solution Method

In order to improve the accuracy of the solution method, the equations of motion were re-derived such that the reaction forces at the pin joint were eliminated, thereby creating a set of four variables and four equations to be solved. The four dependent variables chosen were x_1, y_1, θ_1, and θ_2. The free-body diagrams of body 1 and 2 are shown in Figure 7.4.

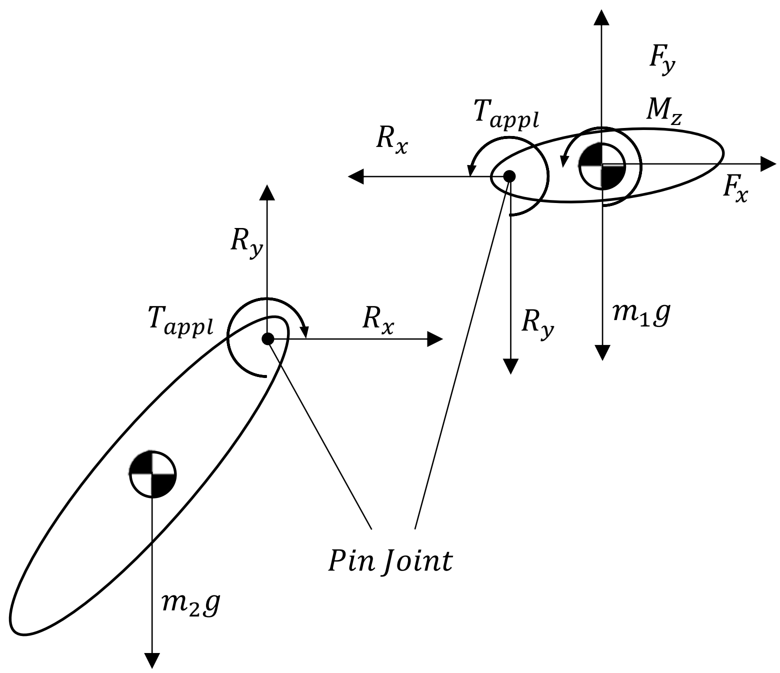


Figure 7.4 - Free body diagram of two-body system

In Figure 7.4, the reaction forces are indicated by R_x and R_y, the fluid forces and moment acting on the wing by F_x, F_y, and M_z, and the torque applied between the bodies to keep balance the MAV by T_{appl}.

The six equations of motion were found to be as follows:

$$Body\ 1:$$

$$x\colon R_x = m_1\ddot{x}_1\ , \tag{7.12a}$$

$$y\colon R_y - m_1 g = m_1\ddot{y}_1\ , \tag{7.12b}$$

$$\theta\colon T_{appl} = R_y L_1 cos\theta_1 - R_x L_1 sin\theta_1 - I_1\ddot{\theta}_1\ , \tag{7.12c}$$

$$Body\ 2:$$

$$x\colon -R_x + F_x = m_2\ddot{x}_2\ , \tag{7.12d}$$

$$y\colon -R_y + F_y - m_2 g = m_2\ddot{y}_2\ , \tag{7.12e}$$

$$\theta\colon T_{appl} + M - R_x L_2 sin\theta_2 + R_y L_2 cos\theta_2 = I_2\ddot{\theta}_2\ , \tag{7.12f}$$

where L_1 and L_2 are the distances from the center of mass of body one to the pin joint and from the center of mass of body 2 to the pin joint, respectively.

By eliminating the reaction forces, one may arrive at the following four equations:

$$(m_1 + m_2)\ddot{x}_1 - F_x + m_2 A = 0\,, \qquad ((7.13a)$$

$$\ddot{y}_1(m_1 + m_2) - F_y + (m_1 + m_2)g + m_2 B = 0\,, \qquad ((7.13b)$$

$$I_1\ddot{\theta}_1 - m_1(g + \ddot{y}_1)L_1\cos\theta_1 + m_1\ddot{x}_1 L_1 sin\theta_1 + T_{appl} = 0\,, \qquad ((7.13c)$$

$$I_2\ddot{\theta}_2 - T_{appl} - M + m_1\ddot{x}_1 L_2 sin\theta_2 - m_1(g + \ddot{y}_1)L_2 cos\theta_2 = 0\,, \qquad (7.13d)$$

where,

$$A = -L_1\dot{\theta}_1^{\,2} cos\theta_1 - L_1\ddot{\theta}_1 \sin\theta_1 - L_2\dot{\theta}_2^{\,2} cos\theta_2 - L_2\ddot{\theta}_2 sin\theta_2\ \ ,$$

$$B = -L_1\dot{\theta}_1^{\,2} sin\theta_1 + L_1\ddot{\theta}_1 \cos\theta_1 - L_2\dot{\theta}_2^{\,2} sin\theta_2 + L_2\ddot{\theta}_2 cos\theta_2\ \ .$$

In order to use the Runge-Kutta method, this system of second order ODEs was then converted into a system of eight first order ODEs as follows:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \\ \theta_2 \\ u_1 \\ v_1 \\ \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ \omega_1 \\ \omega_2 \\ \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}, \tag{7.14}$$

where the nonlinear system solver from [17] was used to find the accelerations within the Runge-Kutta subroutine.

7.5 Selection of Initial Body Angles and Balancing Torque

In order to create a stable two-body system, one must first find the correct angle of body 2 in order to begin near the balancing point for the MAV. In addition, the frictionless pin joint would allow the two bodies to swing wildly without an applied torque to hold the angle between the bodies constant. Therefore, it will also be the goal of this calculation to find the applied torque (T_{appl}). First, the following assumptions will be made:

$$\ddot{x}_1 = \ddot{x}_2 \ , \tag{7.15a}$$

$$\ddot{y}_1 = \ddot{y}_2 \ , \tag{7.15b}$$

$$\dot{\theta}_1 = \ddot{\theta}_1 = \dot{\theta}_2 = \ddot{\theta}_2 = 0 \ . \tag{7.15c}$$

These assumptions effectively describe the state where the bodies move together without rotation. While this may not be possible at every instant, this is generally the goal on average in order to stabilize the MAV and prevent unbalanced rotations.

These six assumptions simplify Eqns. (7.12a)-(7.12f) into the following six equations:

$$R_x - m_1\ddot{x}_1 = 0 \ , \tag{7.16a}$$

$$R_y - m_1(g + \ddot{y}_1) = 0 \ , \tag{7.16b}$$

$$R_y L_1 \cos\theta_1 - R_x L_1 sin\theta_1 - T_{appl} = 0 \ , \tag{7.16c}$$

$$\bar{F}_x - R_x - m_2\ddot{x}_1 = 0 \ , \tag{7.16d}$$

$$\bar{F}_y - R_y - m_2 g - m_2\ddot{y}_1 = 0 \ , \tag{7.16e}$$

$$T_{appl} + \bar{M} - R_x L_2 sin\theta_2 + R_y L_2 cos\theta_2 = 0 \ , \tag{7.16f}$$

where the overbar indicates the time averaged value over the second flapping cycle. The second cycle was chosen because the solution of the forces and moment is very close to the solution after many cycles.

These six equations can then be solved using the nonlinear solution method from [17] for the six unknowns within them. These unknowns are R_x, R_y, T_{appl}, $\ddot{x}_1$, $\ddot{y}_1$, and θ_1. Note that we are not interested in the solution of the linear positions or velocities and that the assumptions from Eqns. 7.15a-7.15c allow for the elimination of the time derivatives of the angular positions. Therefore, the system is solved algebraically without the need for an ODE solver.

As stated earlier, the goal of this calculation is to find θ_1 and T_{appl}. The values found are used as the initial values. After the first time step, an active control system feedback loop will be used to adjust the torque as necessary in order to stabilize the rotation of the MAV.

7.6 Active Stabilization Control

The selected control system is a proportional control closed loop system from [18] which uses the angle of body 2 (the controlled variable) to manipulate the torque applied at the pin joint (equal and opposite to each body). The block diagram for this control system is shown in Figure 7.5.

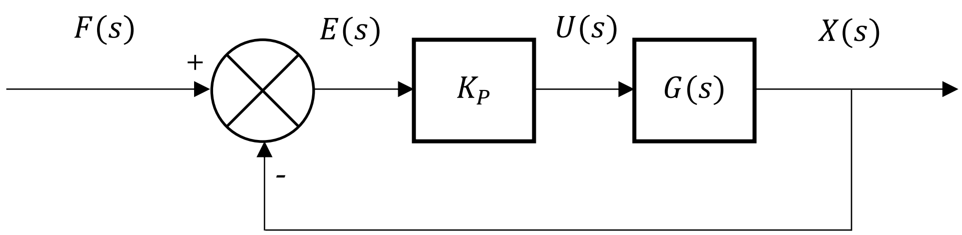


Figure 7.5 - Block diagram of proportional controller

In Figure 7.5, $F(s)$ is the set point, $E(s)$ is the signal error, K_P is the proportional gain, $U(s)$ is the controller input, $G(s)$ is the controller transfer function, and $X(s)$ is the controller output.

The resulting discrete controller output is as follows:

$$T_{appl}^{n} = T_{appl}^{0} - K_P T_{appl}^{0}(\theta_2^{n-1} - \theta_2^{0}) \,, \tag{7.17}$$

where $T_{appl}^{n} = x(t)$ is the updated applied torque used for time step n, T_{appl}^{0} is the approximate applied torque value found in Section 7.5, θ_2^{n-1} is the measured value of the angle of body 2 at the previous time step, and θ_2^{0} is the desired value of the angle of body 2. In a real MAV, the measured value θ_2^{n-1} could be obtained using a gyroscope.

The gain of the proportional control feedback loop was tuned in order to achieve the desired result. It was found that the ideal gain to stabilize the MAV was $400\,\frac{Newton-meters}{radian}$.

7.7 Nonlinear Two-Body Solution Procedure with Control

The full two-body solution procedure, including the nonlinear differential equation solver and active stabilization control is shown in the flowchart in Figure 7.6:

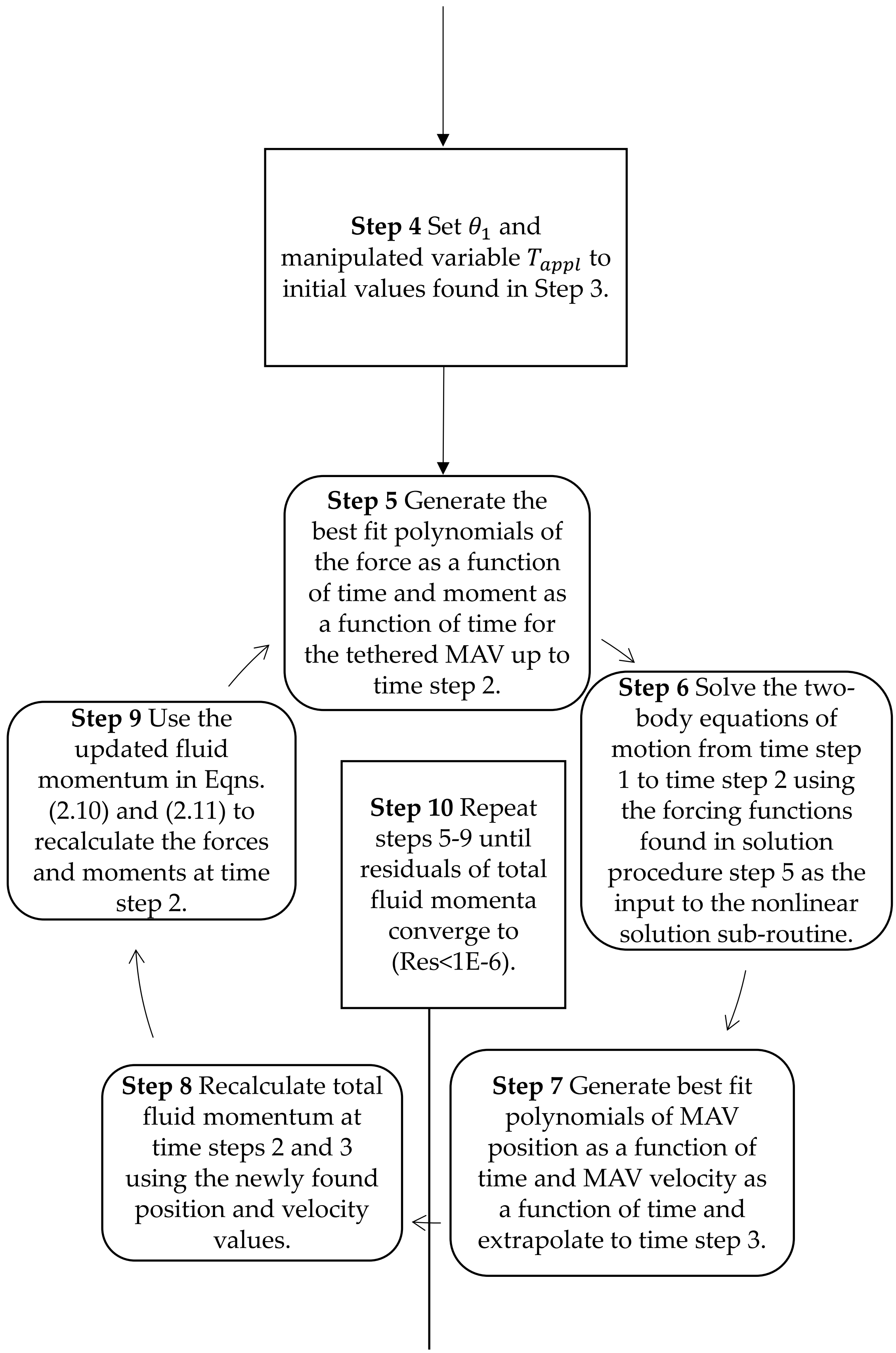
Step 1 Obtain the fluid forces for first two cycles of flapping by fixing the MAV body in space. Fixing the MAV allows the initial force peak associated with the sudden acceleration of the wing from rest to settle. For explanatory purposes, let the first step of the third cycle be time step 3. It is important to note that the heaving direction angle (stroke plane angle from [4]) remains fixed with respect to the ground regardless of the orientation of the MAV.

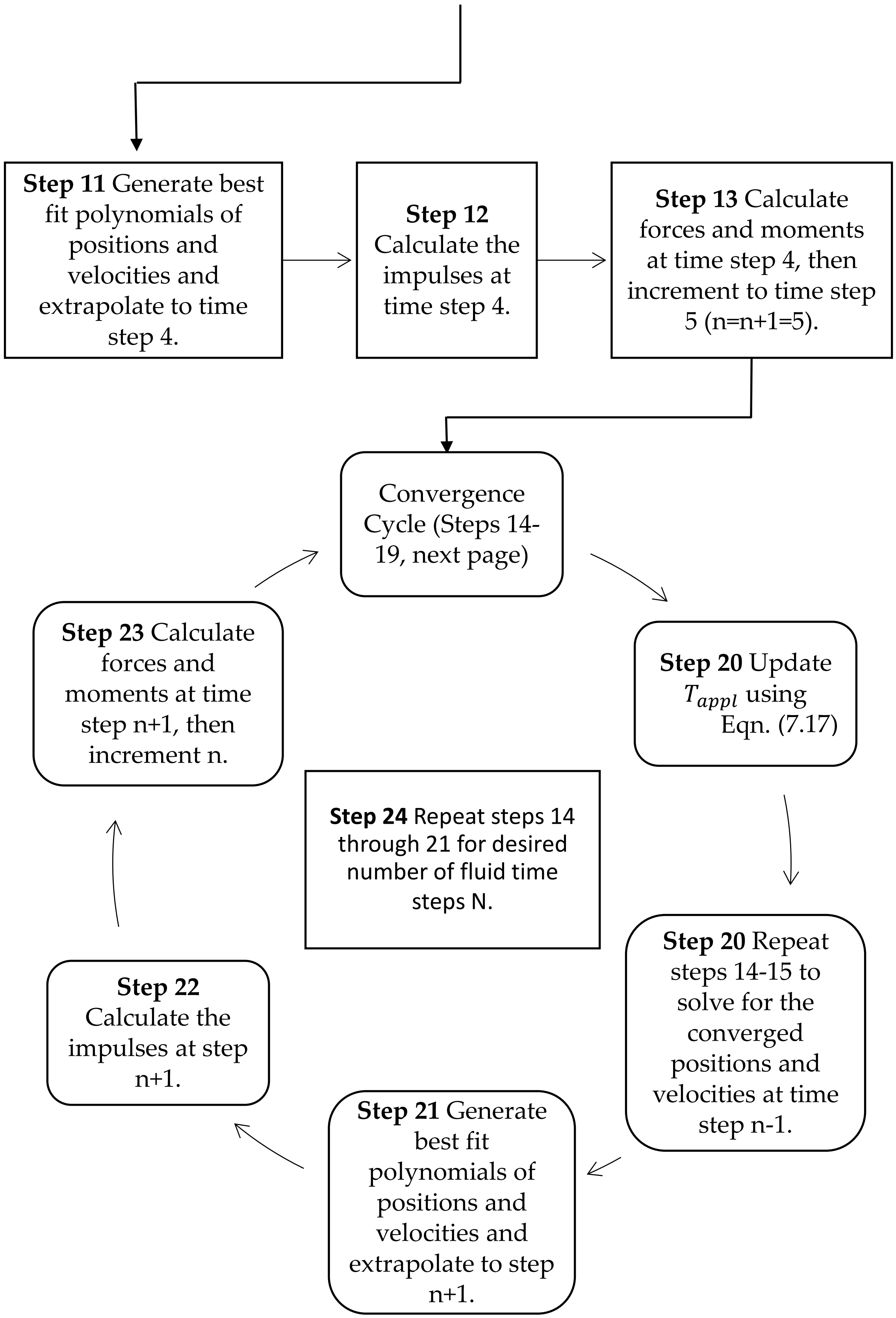
↓

Step 2 Find $\overline{F_x}, \overline{F_y}$, and $\overline{M_z}$ for the second flapping cycle.

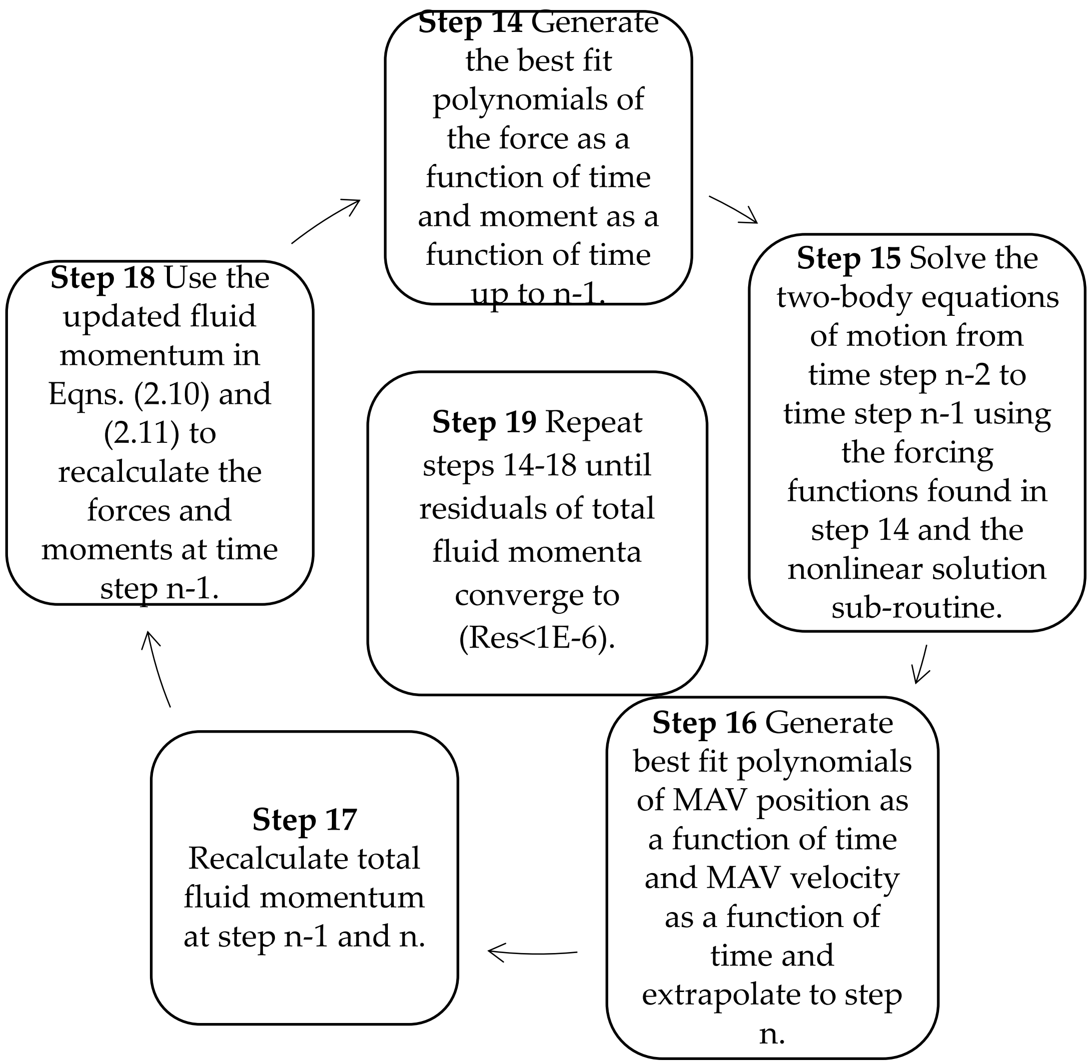
↓

Step 3 Solve for approximate balancing T_{appl} and θ_1 using Eqns. 7.16


Step 4 Set θ_1 and manipulated variable T_{appl} to initial values found in Step 3.
Step 5 Generate the best fit polynomials of the force as a function of time and moment as a function of time for the tethered MAV up to time step 2.
Step 6 Solve the two-body equations of motion from time step 1 to time step 2 using the forcing functions found in solution procedure step 5 as the input to the nonlinear solution sub-routine.
Step 7 Generate best fit polynomials of MAV position as a function of time and MAV velocity as a function of time and extrapolate to time step 3.
Step 8 Recalculate total fluid momentum at time steps 2 and 3 using the newly found position and velocity values.
Step 9 Use the updated fluid momentum in Eqns. (2.10) and (2.11) to recalculate the forces and moments at time step 2.
Step 10 Repeat steps 5-9 until residuals of total fluid momenta converge to (Res<1E-6).


Step 11 Generate best fit polynomials of positions and velocities and extrapolate to time step 4.
Step 12 Calculate the impulses at time step 4.
Step 13 Calculate forces and moments at time step 4, then increment to time step 5 (n=n+1=5).
Convergence Cycle (Steps 14-19, next page)
Step 20 Update T_{appl} using Eqn. (7.17)
Step 20 Repeat steps 14-15 to solve for the converged positions and velocities at time step n-1.
Step 21 Generate best fit polynomials of positions and velocities and extrapolate to step n+1.
Step 22 Calculate the impulses at step n+1.
Step 23 Calculate forces and moments at time step n+1, then increment n.
Step 24 Repeat steps 14 through 21 for desired number of fluid time steps N.

Convergence Cycle



Two-Body Nonlinear Equations of Motion Sub-Routine

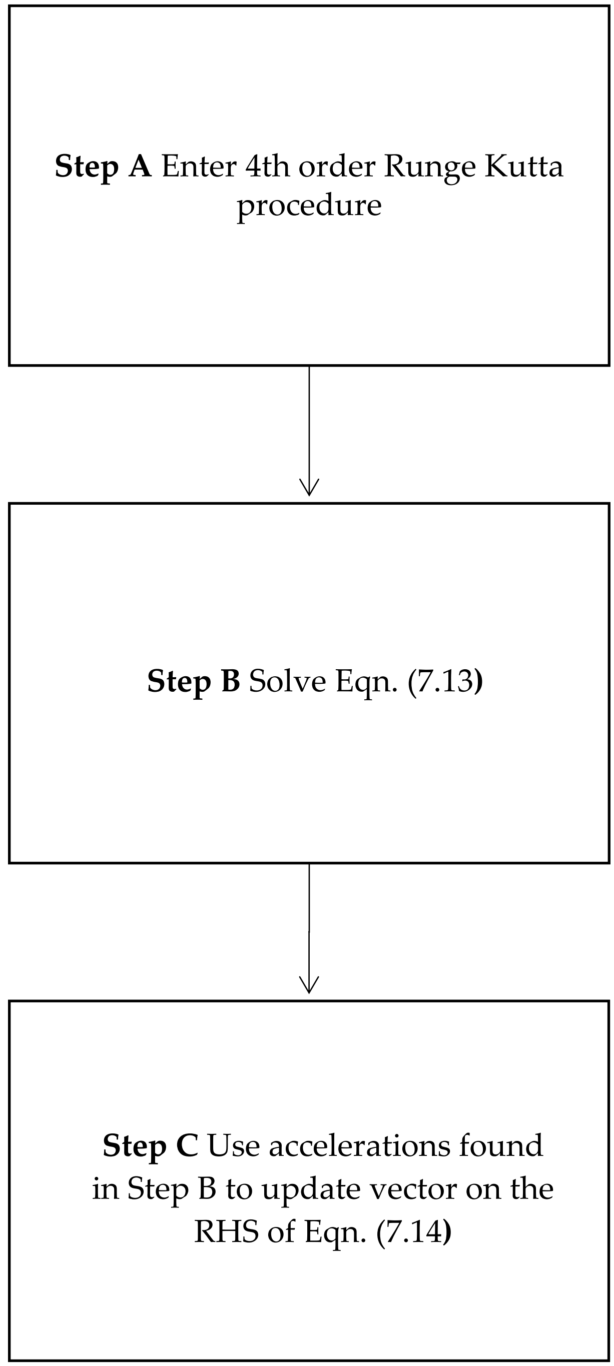


Figure 7.6 - Updated Solution Procedure Flowchart

7.8 Nonlinear Solution Method Results

Figure 7.7 illustrates the resulting wake plot of the nonlinear equation solver with the active control system. Note that the MAV is now balancing such that the rotation of bodies 1 and 2 has ceased. In Figure 7.8, one can note that the control system has effectively reversed the increasing angle of body 1, indicating that the system is beginning to respond to the control and effectively reversing the undesirable rotation.

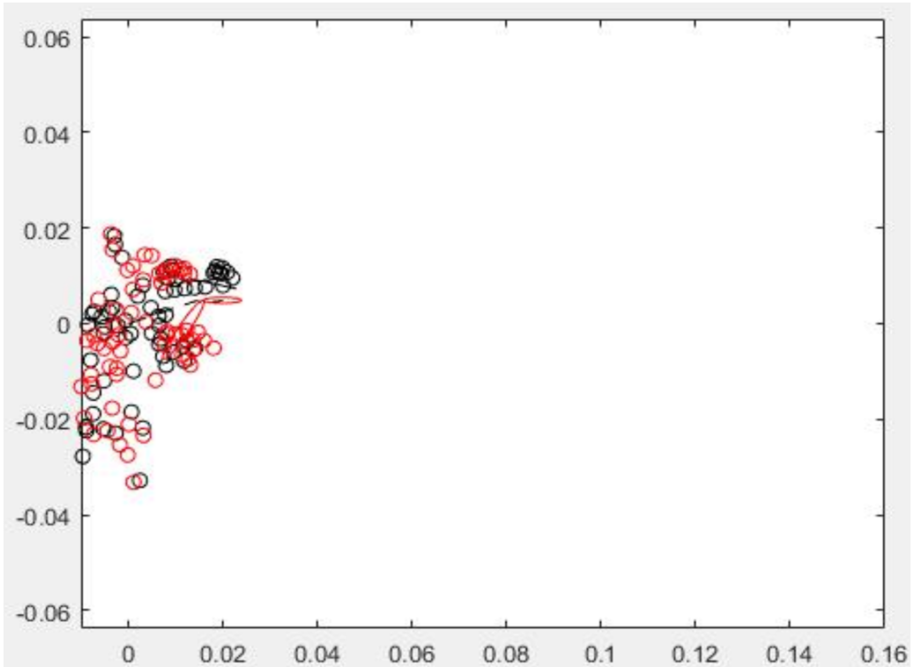


Figure 7.7a

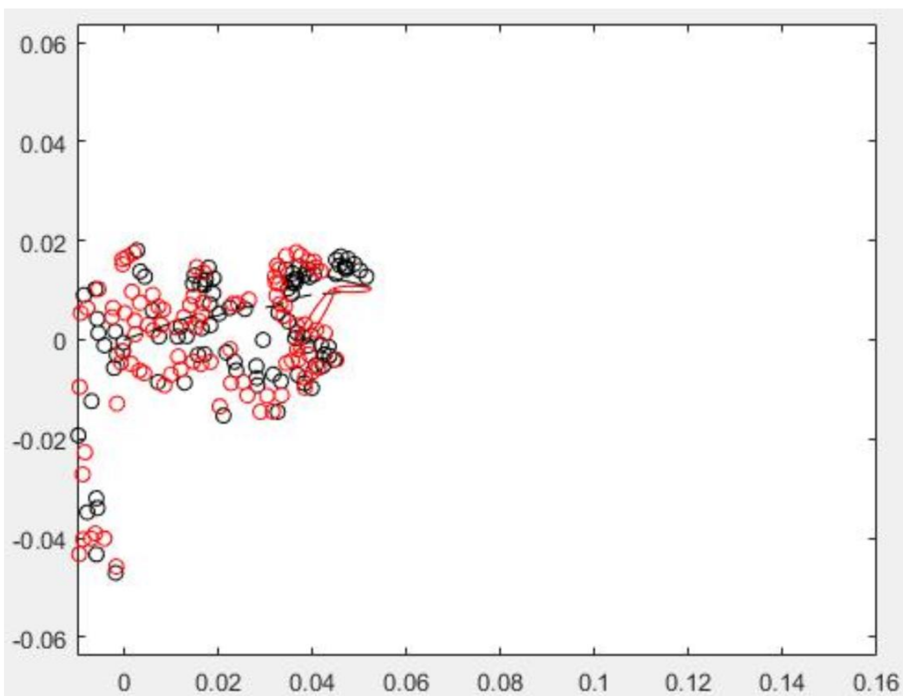


Figure 7.7b

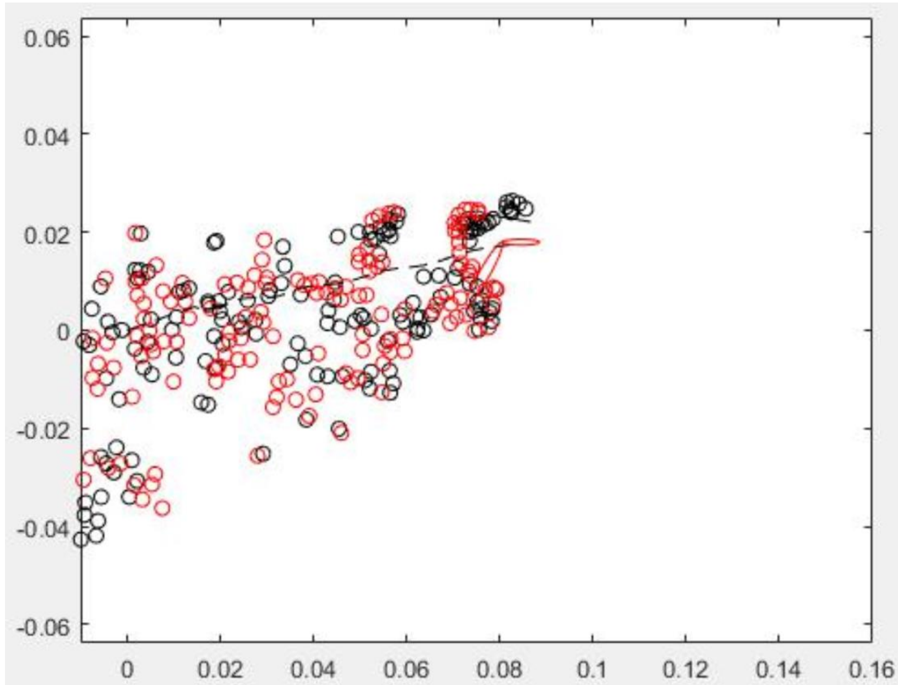


Figure 7.7c

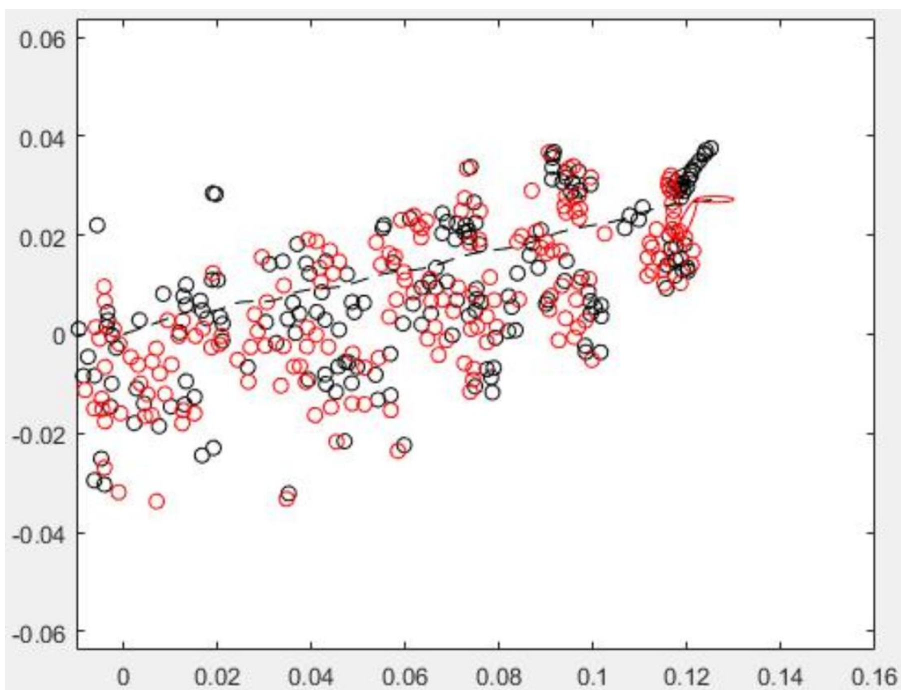


Figure 7.7d

Figure 7.7 - Two-body wake plot with control

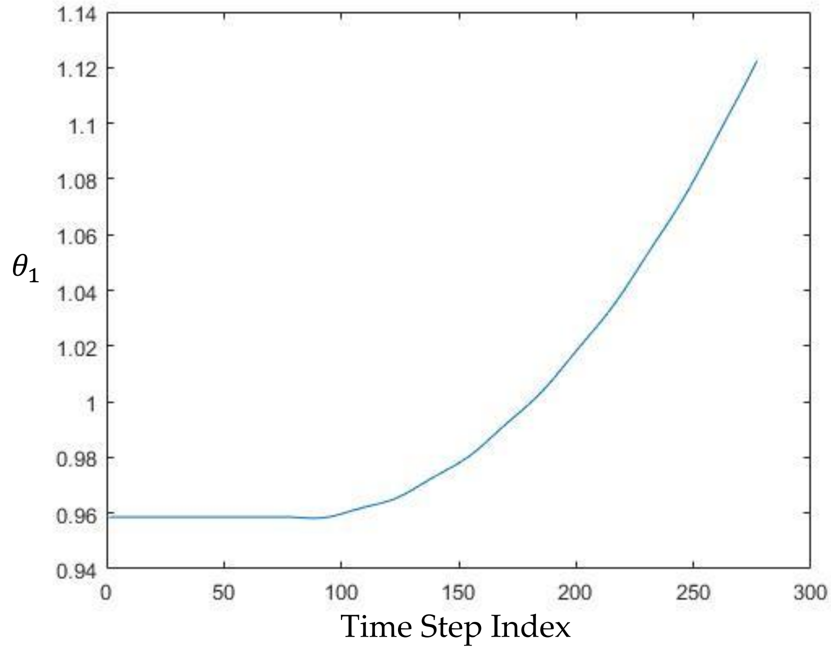


Figure 7.8a

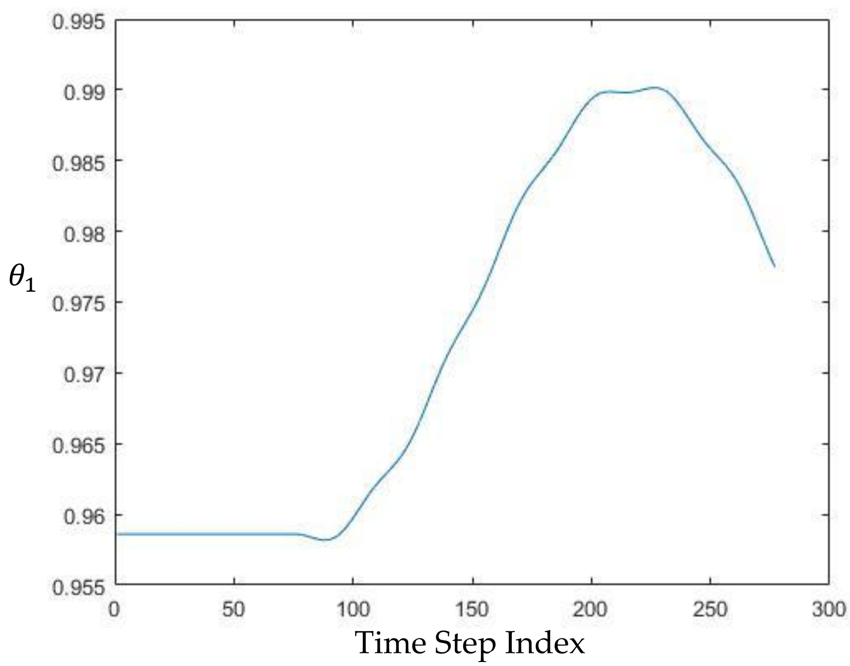


Figure 7.8b

Figure 7.8 - Body 2 angle with (b) and without (a) active control system

Chapter 8 | Conclusions

Fast, specialized CFD solvers will be required to usher in a new age in aerospace technologies. Unsteady flow problems such as that of flapping wings require vast computational resources which make their solutions inaccessible to the average designer.

In addition, the growing energy demand and concerns about its production warrants the investigation of innovative renewable energy technologies. Wind energy is the fastest growing renewable energy market, and the industry standard HAWT has many drawbacks. Flapping wing systems may eliminate many of these drawbacks while maintaining similar aerodynamic efficiencies.

Flapping systems are generally characterized by a weak dependence on Reynolds number and a strong dependence on reduced frequency. The forces are primarily generated by a vortex at the leading edge, which derives its strength from the momentum of the fluid on both the top and bottom surfaces of the wing. The duration which this vortex remains attached defines the optimal flapping frequency and is dependent on the chord length of the wing as well as the free stream velocity of the fluid.

Although flapping wing flight has been studied much more thoroughly than flapping wing energy harvesting, it already appears that many of the same principles are involved in optimizing efficiency apply in both cases. The leading edge vortex is one mechanism that seems to be vital to the viability of both systems, as both operate almost constantly in a highly separated dynamic stall regime.

Rapid pitching is similar to the Magnus effect in that the rotating surface in a cross flow generates lift. It has been shown to enhance lift in both systems a considerable amount.

Wake capture, although carried out with a single wing in insect flight and dual wings in energy harvesting, has been shown to enhance performance in both systems as well.

It also appears that the clap and fling mechanism may apply to both systems. The vortex pair in both cases creates an imaginary wall, compressing the oncoming flow and increasing velocity, thereby lowering the pressure between the wings and increasing the lift force.

This leaves one to wonder whether the tip vortex mechanism may also apply to flapping wing energy harvesters. A further study could include a rotary flapping wing energy harvester that more closely matches the kinematics of the flapping wing insect to examine the plausibility of such a claim.

A novel, highly efficient impulse-momentum vortex method solver has been developed and optimized using a massively parallel computing approach on an affordable workstation GPU. The solver has proven to be very accurate for short simulations, but the pitching moment tends to increase for simulations involving several flapping cycles.

In order to more accurately model the flow physics and maintain a constant moment amplitude for long simulations, a novel approach involving the lumping of far field point vortices allows for the inviscid solver to simulate viscous wake vortex decay.

In order to maintain solution accuracy, it was found that careful adjusting calculations had to be made to correct the linear and angular momenta. In addition, a lumped far field vortex model was adopted to eliminate unphysical high frequencies associated with the removal process, thereby maintaining the circulation solution's stability.

The lumped vortex model was shown to successfully maintain a constant moment, and was compared to the previous method as well as published data from a full unsteady Navier Stokes simulation for a flapping wing.

As an added benefit, the new method greatly reduced the computation time by reducing the proportionality of computation time from N^2 to N. Finally, the time threshold for lumping was determined to be when a given vortex had convected so far from the wing that its influence is less than 0.5% of the free stream velocity.

Several flow angles were tested to ensure the new method was sufficiently consistent in its ability to maintain a constant moment amplitude. The new method did in fact maintain the moment amplitude despite changing incident flow angles.

The Lumped Wake Vortex Method was then coupled successfully to a rigid body solver by the use of a an iterative Kinetic-Predictor Fluid-Corrector algorithm which had been shown to converge to residuals based on the total fluid momenta down to 1E-6.

The fluid-rigid body simulation was then expanded to a two-body pin-connected system in order to test the hypothesis that the abdomen of a flapping wing insect is used to stabilize the thorax. The results of this simulation seem to verify this hypothesis by predicting a reversal of the undesired nose down pitching observed in the single body case.

The initial rigid body solution method was the Lagrange multiplier method. While this method produced reasonably accurate results, the constraint equation was violated, and a better solution method was therefore pursued.

The equations of motion were reduced to a set of four equations and four variables by eliminating the reaction forces between the two bodies. This forced the constraint to be satisfied at the expense of increased solution procedure complexity. A nonlinear equation solver was used within the 4th order Runge Kutta solver to successfully integrate the nonlinear system of ODEs.

An active control system was then implemented successfully, giving promise to the suggested control strategy. The results with and without the active control system indeed provide evidence that the increasing angle observed in the case without control has been successfully reversed.

Future work involving the fluid-rigid body method described herein would include the adaptation of the solver to handle a three-dimensional MAV simulation. Also, the designer would most likely be interested in the motor torque required to drive the flapping mechanism. Therefore, future work could include modifying the simulation to take the motor constants as inputs to determine the flapping motion based on the motor torque rather than prescribing the wing motion. In addition, further work needs to be done to examine the various strategies for control and methods of tuning the active control system. Finally, further work to produce a working prototype of a biomimetic flapping wing MAV could allow for further verification of the simulation as well as the control strategy.

References

[1] T. Kinsey, G. Dumas, G. Lalande, J. Ruel, A. Mehut, P. Viarouge, J. Lemay, Y. Jean, Prototype testing of a hydrokinetic turbine based on oscillating hydrofoils, Renewable energy, 36 (2011) 1710-1718.

[2] W. Shyy, Y. Lian, J. Tang, D. Viieru, H. Liu, Aerodynamics of low Reynolds number flyers, Cambridge University Press2007.

[3] T. Kinsey, G. Dumas, Parametric study of an oscillating airfoil in a power-extraction regime, AIAA journal, 46 (2008) 1318-1330.

[4] M. Denda, P.K. Jujjavarapu, B.C. Jones, A vortex approach for unsteady insect flight analysis in 2D, European Journal of Computational Mechanics, 25 (2016) 218-247.

[5] W. Liu, Q. Xiao, F. Cheng, A bio-inspired study on tidal energy extraction with flexible flapping wings, Bioinspiration & biomimetics, 8 (2013) 036011.

[6] M. Dickinson, Solving the mystery of insect flight, Scientific American, 284 (2001) 48-57.

[7] L. Liu, H. Bian, Z. Du, C. Xiao, Y. Guo, W. Jin, Reliability analysis of blade of the offshore wind turbine supported by the floating foundation, Composite Structures, 211 (2019) 287-300.

[8] L. Yazhou, Studies on wind farm integration into power system [J], Automation of Electric Power Systems, 8 (2003) 17.

[9] J. Young, J.C. Lai, M.F. Platzer, A review of progress and challenges in flapping foil power generation, Progress in aerospace sciences, 67 (2014) 2-28.

[10] C.P. Ellington, C. Van Den Berg, A.P. Willmott, A.L. Thomas, Leading-edge vortices in insect flight, Nature, 384 (1996) 626.

[11] A. Choudhry, Dynamic stall on pitching S809 airfoil, youtube.com, 2012, pp. https://www.youtube.com/watch?v=spOpNm_IA-4.

[12] Q. Zhu, Optimal frequency for flow energy harvesting of a flapping foil, Journal of fluid mechanics, 675 (2011) 495-517.

[13] W. Shyy, P. Trizila, C.-k. Kang, H. Aono, Can tip vortices enhance lift of a flapping wing?, AIAA journal, 47 (2009) 289-293.

[14] M.H. Dickinson, F.-O. Lehmann, S.P. Sane, Wing rotation and the aerodynamic basis of insect flight, Science, 284 (1999) 1954-1960.

[15] A. Leonard, Vortex methods for flow simulation, Journal of Computational Physics, 37 (1980) 289-335.

[16] P.E. Nikravesh, Computer-aided analysis of mechanical systems, Prentice-hall Englewood Cliffs, NJ1988.

[17] J.J. Moré, The Levenberg-Marquardt algorithm: implementation and theory, Numerical analysis, Springer1978, pp. 105-116.

[18] N.S. Nise, CONTROL SYSTEMS ENGINEERING, (With CD), John Wiley & Sons2007.