Numerical and Experimental Analysis of Initial Water Impact of an Air-Dropped REMUS AUV

By

Stephen Michael Roe

B.S., Mechanical Engineering, United States Naval Academy (1996)

Submitted in partial fulfillment of the requirements of the degree of

Master of Science
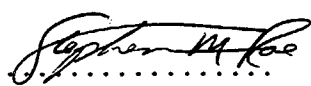
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

and the

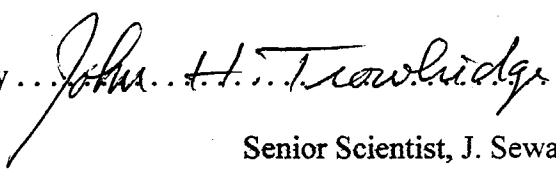WOODS HOLE OCEANOGRAPHIC INSTITUTION

February 2005
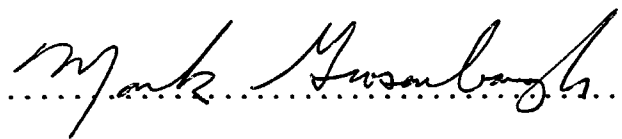
Author . . . .
Stephen M. Roe
Joint Program in Oceanography/Applied Ocean Science and Engineering
Massachusetts Institute of Technology
and Woods Hole Oceanographic Institution
February 2005

Certified by . . . .
John H. Trowbridge
Senior Scientist, J. Seward Johnson Chair as Education Coordinator
Applied Ocean Physics and Engineering, WHOI
Thesis Supervisor

Accepted by . . . .
Mark Grosenbaugh
Chair, Joint Committee for Applied Ocean Science and Engineering
Massachusetts Institute of Technology/
Woods Hole Oceanographic Institution

Numerical and Experimental Analysis of Initial Water Impact of an Air-Dropped REMUS AUV

by

Stephen Michael Roe

Submitted on November 12, 2004, in partial fulfillment
of the requirements for the degree of
Master of Science in Ocean Engineering
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
and the
WOODS HOLE OCEANOGRAPHIC INSTITUTION

Abstract

The initial water impact of a free-falling object is primarily related to the fluid forces on the wetted surface of the object. The shape-dependent added-mass coefficients express the fluid forces integrated over the body, and thus physically represent the additional inertia of water accelerated with the body. The field of hydrodynamic impact has been primarily concerned with estimating the added-mass coefficients of various types of bodies for different water impact types, such as seaplane landings, torpedo drops, and ship slamming.

In this study, a numerical model has been constructed to estimate the hydrodynamic impact loads of a REMUS dropped in free-fall from a helicopter in a low hover. Developed by von Alt and associates at Woods Hole Oceanographic Institution, the REMUS (Remote Environmental Monitoring UnitS) is a small, man-portable, torpedo shaped Autonomous Underwater Vehicle (AUV) that is normally operated from small boats for a variety of scientific, industrial, and military applications. Finite-element method software and computer aided drafting tools were used to create a simplified model of REMUS without fins, propeller, or transducers. This axisymmetric REMUS model was cut by a flat free surface at various pitch angles and submergence values, and a panel mesh of the wetted surface of the vehicle was created using an automatic mesh generator. Surface boundary conditions are enforced for the free surface by reflecting the body panels using the method of images. Each panel mesh was evaluated for its added-mass characteristics using a source collocation panel method developed by Dr. Yonghwan Kim, formerly of the Vortical Flow Research Laboratory (VFRL) at the Massachusetts Institute of Technology. Experimental impact tests were conducted with a specially-instrumented test vehicle to verify the initial impact accelerations.

Thesis Supervisor: John H. Trowbridge

Title: Senior Scientist, WHOI

Co-Thesis Supervisor: Christopher J. von Alt

Title: Principal Engineer, WHOI

Acknowledgements

My most gracious thanks are extended to my multiple advisors for this thesis. Chris von Alt inspired me to take up this project and provided the resources to make it happen. Yonghwan Kim provided the panel method code used extensively in this work and provided much needed theoretical background. Thanks also to John Trowbridge for providing necessary direction and getting me through these last anxious days.

My opportunity to participate in graduate studies at MIT and WHOI was made possible through the Secretary of the Navy's Program in Oceanography and Oceanographic Engineering and Rear Admiral Steven Tomaszeski, USN, Oceanographer of the Navy.

My special thanks to Marga McElroy for looking after me, and her aid in getting things done in and around the Village. Her encouragement and support have been a tremendous aid.

Thanks go out to all of my classmates who have made this an enjoyable shore duty in the Joint Program, but especially to Dave Stuebe, who has always been there whether to move furniture, launch a REMUS, or go for a run.

Finally, my everlasting thanks to Lynne Elkins for her limitless patience and continuing support. You have made it worth coming here, but you make it so hard to go.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Statement of Problem

This study aims to calculate the accelerations associated with the initial water entry of the REMUS (Remote Environmental Monitoring UnitS) Autonomous Underwater Vehicle (AUV) following a free-fall drop from a helicopter in a low hover. The initial stages of water impact in this regime are dominated by the effect of momentum transfer from the impacting body to the water. Added-mass coefficients describe the effect of entrained water accelerated in response to body acceleration. This is a potential flow effect due to the body obstructing the flow. For a hydrodynamic impact problem, the wetted body changes with submergence, so the geometry and the added-mass are not constant, as in fully submerged motion problems. The effective impact force is thus a result of body acceleration, body velocity, and the added-mass coefficients and their derivatives.

1.1.1 Background

The Autonomous Underwater Vehicle is a valuable platform for ocean exploration. AUVs can be engineered in almost endless configurations to perform many diverse tasks. Free-swimming vehicles can perform many useful tasks, such as performing precise surveys of bottom bathymetry or hovering to take photographs of interesting features. AUVs perform repetitive tasks with precision and speed, maneuvering in ways that towed or tethered platforms cannot. AUVs are particularly useful in unfriendly environments that are unsuitable for human divers.

Modular AUVs use a common platform which can be configured to meet a variety of different mission parameters. One such platform is the REMUS AUV, developed by von Alt and

associates at the Oceanographic Systems Laboratory at the Woods Hole Oceanographic Institution (WHOI). REMUS was designed to be lightweight, low-cost, and man-portable. These attributes make REMUS attractive to a broad spectrum of workers in the scientific, commercial, and military communities. More than 50 REMUS systems have been operated in 20 different configurations by nine universities, three U.S. Navy laboratories, one British defense laboratory, and three branches of the U.S. Navy [3].

The U.S. Navy uses the REMUS primarily for mine countermeasures (MCM), where its 100-meter-depth capability is well-suited to the Very Shallow Water (VSW) missions that are common in littoral operations. The vehicles are operated by specialized units, such as Naval Special Clearance Team ONE (NSCT-1), which consists of Navy SEALs, EOD divers, Marine Corps Force Reconnaissance divers, and marine mammals. While NSCT-1 and its predecessor, the Very Shallow Water Detachment, have been evaluating REMUS since 2001, the first wartime operational deployment of the instrument took place in 2003, when it proved to be an essential tool for performing surveys of the waters around Umm Qasr during the early stages of Operation Iraqi Freedom [3].

1.1.2 Motivation

Current U.S. Navy users deploy REMUS from small boats such as Combat Rubber Raiding Crafts. This mode of operation limits REMUS use to potential mine danger areas that are readily accessible by small boats. REMUS's torpedo shape and its small size suggest that it could be successfully deployed by dropping it from a helicopter in a low hover in a procedure similar to that used dropping a rescue swimmer. This requires the helicopter to hover at an altitude of 15 feet during the drop. Rapid deployment of REMUS vehicles from helicopters is a capability that will increase the range of operations from the base unit and reduce transit times to the objective areas. A single helicopter could support multiple AUVs to reduce search time for large objective areas.

REMUS was not originally designed to meet the criteria for this type of deployment, so a study of the impact dynamics is an appropriate start for an evaluation of this use. For this

method to be viable, predicted impact forces must be small enough to permit routine use without compromising vehicle integrity. This requires the maintenance of watertight integrity and that no control surfaces or internal components are damaged. The vehicle should be able to start up and execute its mission file as in any other deployment. REMUS was designed for criteria such as precision navigation, instrument payload, and mission range. These considerations create a vehicle less rugged than a torpedo weapon, which is deployed at higher speed and altitudes. Further tests will be required to determine whether the construction, control surfaces, and instrumentation are suitable for routine helicopter deployment.

1.2 Previous Work

The fundamental paper on the subject of hydrodynamic impact is von Karman's 1929 study of pressure on seaplane floats during landing [13]. This analysis considered the effect of a two-dimensional wedge impacting a horizontal free surface of water. During the impact, the initial momentum of the descending wedge is distributed between the wedge and the water, where the added-mass models the effects of the water. Von Karman approximated the added-mass of the wedge as a function of depth to be one-half the flat-plate added-mass for the width of the wedge at the waterline. He was able to calculate accelerations for wedges of various dead rise angles, and from these, forces and pressures. His maximum pressures agreed well with experimental results obtained from pressure gauges in landing floats.

Subsequent work suggested additional corrections to von Karman's technique. Wagner's 1932 research [14] added the effect of additional wetted area due to local increase in water elevation. Excellent literature reviews of the subsequent work have been written by many authors, including Szebehely [12] and May [4]. Payne's 1981 paper [11] suggests that von Karman's original formulation agreed more closely with later, more reliable experimental data than that of Wagner and subsequent authors who based their work on his corrections. Boef's papers on the impact of free-fall lifeboats [1, 2] reviewed theory, provided detail using von Karman's method for horizontal impact of circular cylinders, and developed a method for computing the two-dimensional impact forces and trajectories for a free-fall lifeboat with three

12

degrees of freedom. Finally, there exist numerous analytical studies of simple geometries, such as Miloh's work on the added-mass characteristics of a sphere impacting a free surface [6, 7, and 8]. Such studies are useful as a validation for numerical techniques which can handle more complicated geometries.

1.3 Overview of Thesis/Scope of Work

This thesis describes the development and validation of a numerical model to predict the accelerations associated with the initial water entry of a REMUS AUV vertically dropped from a helicopter in a low hover. Chapter 2 will describe in detail the theory behind and development of the numerical model. Chapter 3 will describe the validation of the model against known analytic solutions and present numerical results for the REMUS shape. Chapter 4 will describe the experimental setup, the characteristics of the REMUS test vehicle and a comparison of numerical and experimental results. Chapter 5 will present summary, conclusions, and recommendations for further work on this topic.

Performing live helicopter drops of an actual REMUS vehicle as a proof of concept or to collect data about the impact is beyond the scope of this thesis. This work intends to perform an initial analysis for the purpose of better defining its dynamics prior to any official Navy test program to qualify this deployment method.

Chapter 2

Numerical Modeling

In this chapter I will show the development method for calculating the water impact of a free-falling REMUS. This technique uses the momentum formulation of the generalized von Karman method with the heave added-mass coefficients calculated numerically using a source collocation panel method. A method is demonstrated to automatically generate panel meshes using FEMLAB.

2.1 The Generalized von Karman Method

Von Karman was the first to propose a method for calculating the forces associated with the impact of water entry. His 1929 paper [13] looked at the specific case of a seaplane landing, simplified as a two-dimensional wedge in a flat impact. He then applied conservation of momentum to the body mass and the added-mass to calculate the maximum pressures associated with the impact. The generalized von Karman method applies these principles to other shapes, without the corrections added later by Wagner [14] and subsequent authors. There are several reasons to exclude the successive corrections. First, it has been suggested by Payne [11] that these corrections are incorrect and cause an increase in the error of the predicted forces. Second, regardless of their merit, the methods proposed by Wagner and the subsequent authors add layers of complexity that are difficult to execute for the more complicated geometries that will be addressed here via numerical methods.

Von Karman developed a formula for the added-mass of a wedge shape by using a simple analytically understood shape, the two-dimensional flat plate. The width of the flat plate increases with submergence due to the angle of inclination of the under surface of the float. The

determination of the added-mass of the wetted portion of the shape as it proceeds into the fluid is the most important problem in this technique. The problem of computing added-mass near the free-surface is a much more complicated problem than added-mass fully submerged and far from walls or other interfaces. The free-surface adds an additional boundary condition with the effects of waves and splash. The von Karman method considers large impact effects, so a first order approximation replaces the free-surface with a constant equipotential flat surface. Splash effect, incoming wave diffraction, and wave radiation are considered higher order effects, which are neglected.

2.1.1 Momentum Considerations

The dynamics of the impact of an arbitrarily-shaped body are governed by momentum principles. Unless otherwise specified, the coordinate system for the following discussion is a right-handed Cartesian inertial system (x, y, z) centered on the free-surface with z positive upward as defined in Figure 2.1. In the equations of motion, z refers to the position of the lowest point of the body according to this system. The indicial notation will also be used as needed, where dots denote time derivatives:

Figure 2.1: Vehicle referenced to inertial coordinate system
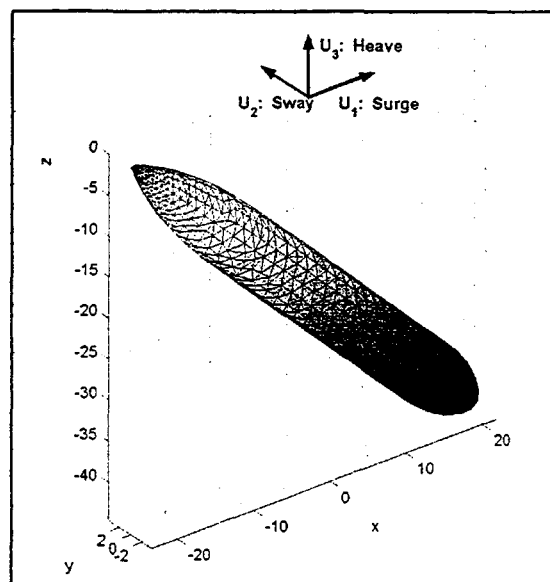
$$(x, y, z) = (x_1, x_2, x_3), \text{ and} \qquad (2.1.1)$$

$$(\dot{x}, \dot{y}, \dot{z}) = (U_1, U_2, U_3). \qquad (2.1.2)$$

The following subsection is after Boef [1], with new notation. He states, "Hence by treating the impact at water entry as an elastic collision between the entering body and its added mass, we can derive its motion without cumbersome fluid-dynamical calculations." The initial momentum of the body just prior to impact, $m\dot{z}_0$, is at subsequent time-steps divided between the body and the fluid:

$$m\dot{z}_0 = (m + a_{33}(z))\dot{z}, \qquad (2.1.3)$$

where m is the body mass, $\dot{z}_0$ is the initial vertical impact velocity, and a_{33} is the heave added-mass coefficient of the body in the presence of the free-surface, derived below; see subsection 2.1.2. The momentum transferred to the water during impact is $a_{33}\dot{z}$, the hydrodynamic inertia felt by the body during immersion. The force associated with this momentum transfer is

$$F_{mom}(t) = \frac{d}{dt}(m\dot{z}(t)) = -\frac{d}{dt}(a_{33}(z)\dot{z}(t)), \qquad (2.1.4)$$

which can also be expressed as

$$F_{mom}(t) = -(a_{33}\ddot{z}) - (\dot{a}_{33}\dot{z}) = -a_{33}\ddot{z} - \frac{da_{33}}{dz}(\dot{z})^2. \qquad (2.1.5)$$

This force is the primary force at work during impact. The effects of gravity and buoyancy are neglected, as by von Karman in his original treatment. The generalized equations of motion for the free-fall hydrodynamic impact can be expressed

$$m\ddot{z} = -mg, \quad z(t) > 0 \qquad (2.1.6)$$

$$m\ddot{z} = F_{mom}. \quad z(t) < 0 \qquad (2.1.7)$$

It follows that the acceleration of the body following impact is

$$\ddot{z} = \frac{-\dfrac{da_{33}}{dz}(\dot{z})^2}{m + a_{33}}, \quad z(t) < 0 \qquad (2.1.8)$$

and the total impact force experienced by the body is

$$F_{impact} = m\ddot{z} = \frac{m}{m + a_{33}}\left(-\frac{da_{33}}{dz}(\dot{z})^2\right). \quad z(t) < 0 \tag{2.1.9}$$

2.1.2 The Definition of the Added-Mass Coefficient

Large impact problems are dominated by inertial effects; therefore, it is appropriate to treat the fluid as ideal (inviscid) and irrotational. The following is a summary of chapters 4 and 6 in Newman's *Marine Hydrodynamics* [10], which deal with potential flow and the definition of added-mass in an unbounded fluid and in the presence of a free-surface. Equations in this subsection are from that text, expressed in my notation.

Newman's Chapter 4 describes potential flows, or ideal and irrotational flows in which the velocity field is expressed as the gradient of the scalar velocity potential ϕ:

$$V = \nabla\phi. \tag{2.1.10}$$

Velocity potentials must satisfy the Laplace Equation,

$$\nabla^2\phi \equiv \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = 0, \tag{2.1.11}$$

which expresses continuity, or the conservation of fluid mass. The simplest potential flow is a uniform stream with velocity components (U, V, W), represented by

$$\phi = Ux + Vy + Wz. \tag{2.1.12}$$

Another simple yet important potential flow is the source, which is called a singularity because it satisfies the Laplace Equation at all points in the domain except at its location. The following is the equation for a source of strength q located at source point (ξ, η, ζ) at radius r from field points (x, y, z) in a three-dimensional fluid domain:

$$\phi = -\frac{q}{4\pi r} = -\frac{q}{4\pi\sqrt{(\xi - x)^2 + (\eta - y)^2 + (\varsigma - z)^2}} \tag{2.1.13}$$

Since both the source and the uniform stream independently satisfy Equation 2.1.11, adding them together also provides a solution, in this case a Rankine half body. The half body has a dividing stream surface which separates the outside fluid domain from the inside domain. The inner

17

domain contains the source singularity where the Laplace Equation breaks down. The external domain represents the flow of fluid of a uniform stream around a body obstructing the flow. The Rankine ovoid is a simple closed body that can be represented by the addition of another source downstream and inside the inner domain, with equal strength but opposite sign (a sink) to the first source. In the limiting case as the distance separating the source/sink pair goes to zero, the resulting ovoid approaches a sphere. The source/sink pair becomes a new singularity, the dipole or double source,

$$\phi = \frac{\mu x}{4\pi\left(x^2 + y^2 + z^2\right)^{\frac{3}{2}}},$$

(2.1.14)

in the limit as the distance a between the source/sink pair goes to zero (in this case along the x axis) while the product of their strength and the distance between is kept constant, so that

$$\mu = \lim_{a\to 0} 2qa.$$

(2.1.15)

Singularities of varying strengths may be combined in a volume to represent flow past more complex body shapes. Details of the external flow can be calculated from this composite velocity potential, along with the resulting forces and pressures acting on the body.

Newman's Chapter 4 presents the derivation of the added-mass coefficients in terms of hydrodynamic pressure integrated over the general body surface and an unbounded ideal fluid domain. Bernoulli's equation is written in terms of ϕ:

$$\frac{\partial \phi}{\partial t} + \frac{1}{2}\frac{\partial \phi}{\partial x_j}\frac{\partial \phi}{\partial x_j} = -\frac{1}{\rho}\left(p + \rho g z\right),$$

(2.1.16)

Where ρ is the density, p is the pressure, and the summation convention for repeated indices applies. Equations for the force and moment on the fully submerged body are written in terms of integrals over the body surface using Bernoulli's equation:

$$\mathbf{F} = \iint_{S_B} p\mathbf{n} \, dS = -\rho \iint_{S_B}\left(\frac{\partial \phi}{\partial t} + \frac{1}{2}\nabla\phi\cdot\nabla\phi\right)\mathbf{n} \, dS,$$

(2.1.17)

$$\mathbf{M} = \iint_{S_B} p\left(\mathbf{r}\times\mathbf{n}\right) \, dS = -\rho\iint_{S_B}\left(\frac{\partial \phi}{\partial t} + \frac{1}{2}\nabla\phi\cdot\nabla\phi\right)\left(\mathbf{r}\times\mathbf{n}\right) \, dS,$$

(2.1.18)

where S_B is the surface of the body, r is the position vector, and n is the outward normal vector on the surface of the body (directed into the fluid domain). For a reference frame in which the body is moving in an unbounded fluid, these equations are cast in the form

$$F_j = -\dot{U}_i m_{ji} - \varepsilon_{jkl} U_i \Omega_k m_{li} \tag{2.1.19}$$

$$M_j = -\dot{U}_i m_{j+3,i} - \varepsilon_{jkl} U_i \Omega_k m_{l+3,i} - \varepsilon_{jkl} U_i U_k m_{li}. \tag{2.1.20}$$

Here ε_{jkl} is the alternating tensor and the indicial notation applies for the velocity vector U and the rotation rate vector Ω

$$U = (U_1, U_2, U_3), \tag{2.1.21}$$

$$\Omega = (\Omega_1, \Omega_2, \Omega_3) \equiv (U_4, U_5, U_6), \tag{2.1.22}$$

and

$$m_{ji} = \rho \iint_{S_B} \phi_i \frac{\partial \phi_j}{\partial n} dS, \quad i, j = 1...6 \tag{2.1.23}$$

is the six-by-six symmetric added-mass matrix for a generalized three-dimensional shape fully submerged in an ideal, irrotational fluid far from boundaries. The importance of this derivation for the present work is not the equations for force and moment, but the concept of added-mass, which represents the weighted integration of fluid particles accelerated with the body in response to body motion [10]. Thus, the added-mass coefficient is a generalization that expresses potential flow effects of interaction between a body and the surrounding water purely as a function of body geometry.

Newman's [10] Chapter 6 derives force and moment equations similar to those above, accounting for the effects of hydrostatics and surface wave diffraction and radiation near a free-surface. The parts of interest to the impact problem are the force and moment which are proportional to body acceleration and velocity.

$$F_i = -\sum_{j=1}^{6} (a_{ij} \dot{U}_j + b_{ij} U_j), \quad i = 1, 2, ..., 6 \tag{2.1.24}$$

19

The added-mass coefficient, a_{ij}, and damping coefficient, b_{ij}, are functions of the wave frequency, ω. The added-mass coefficients a_{ij} and m_{ij} are different but can be related. The impact problem involves impulsive motion, so the free-surface boundary condition becomes

$$\phi_j = 0 \quad \text{on } z = 0, \tag{2.1.25}$$

as inertia forces dominate gravity in the limit as $\omega \to \infty$. In this high frequency limit, the potentials must be odd in z, and the vertical modes, heave, roll, and pitch, will correspond to a rigid double body in which the wetted portion of the body is reflected about a fixed horizontal plane using the method of images. The double body is then evaluated as for the fully submerged added-mass coefficient, m_{ij}, and the desired free-surface coefficients are

$$a_{ij} \to \frac{1}{2} m_{ij}, \quad i, j = 3, 4, 5 \tag{2.1.26}$$

and

$$b_{ij} \to 0. \tag{2.1.27}$$

The remaining a_{ij}, i,j = 1, 2, 6 are not related to the double body, and will not be calculated in the present work. This analysis uses only the heave added-mass coefficient, a_{33}, in its calculation of forces and accelerations due to impact following vertical free-fall. Since the wetted portion of the body changes as a function of submergence and orientation as it enters the water, the double body and the coefficient a_{33} must be calculated at each step of the process.

2.2 The Collocation Source Panel Method Formulation

Added-mass coefficients are analytically known for a small number of body shapes. Analytical solutions are generally limited to simple geometries, although solutions are sought for complex shapes such as the present case. Two-dimensional shapes have proven useful, such as the flat-plate in von Karman's original solution. Strip theory has been a popular tool in naval architecture when the body is slender and the two-dimensional added-mass coefficient is known for the slice of the body represented by each strip. A fully-submerged REMUS vehicle could be

so represented, but the technique fails for the complex double body shapes that result from intersecting the body with the flat free-surface at a range of pitch angles and submergence values. Another method is needed to calculate the hydrodynamic effects of complex body shapes.

Complex double-body shapes representing free surface penetration require a similar but more intricate treatment. The previous section discussed a method of expressing a closed body as a sum of elementary velocity potentials, with the singularities separated from the exterior flow by a dividing stream surface. A better method is to cover the surface of the closed body with a continuous distribution of singularities of varying strengths. This is the approach taken by a collocation panel method, which divides the body geometry into a number of panels in order to discretely solve a system of equations and find the strength for the distribution of singularities on each panel. The panel method used for the present work was written by Dr. Yonghwan Kim of the Seoul National University, formerly of the Vortical Flow Research Lab at the Massachusetts Institute of Technology. The description of the method is courtesy of notes provided by Dr. Kim, with additional insight provided by class notes from Professor Jerome Milgram's course *13.024 Numerical Marine Hydrodynamics* at the Massachusetts Institute of Technology [5].

The problem is set up as follows: a body enclosed by the boundary S_B exists in an unbounded fluid domain with field points defined by the coordinate $\mathbf{x} = (x, y, z)$, and points on S_B are defined by the coordinate $\xi = (\xi, \eta, \zeta)$. The normal vector, $\mathbf{n}$, points into the body (out of the fluid). Flow in the fluid domain is characterized by the total potential Φ, consisting of a uniform flow of velocity U plus a perturbation potential ϕ, caused by the presence of the body.

$$\Phi = -Ux + \phi \tag{2.2.1}$$

Elementary source singularities

$$G(\xi, \mathbf{x}) = \frac{1}{\sqrt{(\xi - x)^2 + (\eta - y)^2 + (\zeta - z)^2}}. \tag{2.2.2}$$

are distributed on S_B with strength $\sigma(\xi)$, such that

$$\phi(\mathbf{x}) = \iint_{S_B} \sigma(\xi) G(\xi, \mathbf{x}) dS, \tag{2.2.3}$$

where $\phi(\mathbf{x})$ is the velocity potential of the fluid as a result of the presence of the body. If the body is then divided into N panels and the source strength is assumed constant over each panel, the discrete form of Equation 2.2.3 is

$$\phi(\mathbf{x}) = \sum_{i=1}^{N} \sigma_i G(\xi, \mathbf{x}) \Delta S_i,$$ (2.2.4)

where σ_i is the constant source strength of the ith panel.

The normal derivative of the fluid velocity, $\dfrac{\partial \phi}{\partial \mathbf{n}}$, is known from the no-flux body boundary condition, which states that no fluid passes through the surface, so the normal velocity must be zero on the surface:

$$\frac{\partial \Phi}{\partial \mathbf{n}} = 0$$ (2.2.5)

$$-U\hat{i} \cdot \mathbf{n} + \frac{\partial \phi}{\partial \mathbf{n}} = 0, \qquad \frac{\partial \phi}{\partial \mathbf{n}} = U\hat{i} \cdot \mathbf{n}$$ (2.2.6)

where $\hat{i}$ is the unit vector in the x direction. The discretized boundary value problem can be written as

$$\frac{\partial \phi}{\partial \mathbf{n}}(x_k) = \sum_{i=1}^{N} \sigma_i \frac{\partial}{\partial \mathbf{n}} G(\xi; \mathbf{x}_k) \Delta S_i = U\hat{i} \cdot \mathbf{n}_k. \qquad k = 1, 2, ..., N$$ (2.2.7)

This condition is enforced at the N collocation points, chosen as the centroids of the panels and resulting in an $N \times N$ system of equations that can be solved for the sources strengths σ_i.

Once the values of σ_i are known, properties of the fluid flow can be calculated at any external location by summing the contribution of each panel. The added-mass coefficients are easily calculated this way, where Equation 2.2.8 replaces Equation 2.1.23 in the discrete case:

$$m_{ji} = \rho \sum_{k=1}^{K} \left(\phi_i \frac{\partial \phi_j}{\partial n} \right)_k \Delta S_k.$$ (2.2.8)

The output of the panel code provided by Dr. Kim is a six-by-six matrix of added-mass coefficients in units of L^3, L^4, and L^5, where L is a length unit. These are converted to units of mass and products and moments of inertia by multiplying by the density of water, ρ_w, resulting in units of M, ML, and ML^2 (where M is a mass unit). Each bracketed expression in Equation 2.2.9 is a three-by-three dimensionally-homogeneous block.

$$m_{ij} = \rho_w \cdot output_{ij} \rightarrow \rho_w \cdot \begin{bmatrix} \begin{bmatrix} L^3 \end{bmatrix} & \begin{bmatrix} L^4 \end{bmatrix} \\ \begin{bmatrix} L^4 \end{bmatrix} & \begin{bmatrix} L^5 \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \begin{bmatrix} M \end{bmatrix} & \begin{bmatrix} ML \end{bmatrix} \\ \begin{bmatrix} ML \end{bmatrix} & \begin{bmatrix} ML^2 \end{bmatrix} \end{bmatrix} \qquad (2.2.9)$$

2.3 Panel Generation for Generalized Body Shapes

The development of the panel method above was justified by the complexity of the double body shapes needed to model hydrodynamic impact. However, before these methods can be used the body shapes must be accurately transformed into panel meshes. The generation of panel representations of various shapes is one of the primary tasks of the present work.

2.3.1 Properties of a good panel mesh

Prior to the discussion of how mesh generation was accomplished, it is necessary to review the properties of a good panel mesh. These properties result from the requirements of the panel method in use. In this case panels

1. may be defined by either three or four nodes,

2. should have a fairly uniform size (large size discrepancies result in numerical errors), and

3. should have an aspect ratio near unity (long, thin panels result in numerical errors).

Additional properties are necessary in order to represent double bodies as appropriate for an impact calculation by the generalized von Karman method; panels

4. should cleanly represent the surface cut by the free surface,

5. should take advantage of body symmetry, especially about the free surface, and

6. should be automatically generated as a function of submergence.

The last property was added because a large number of panel meshes are required to resolve the relationship between a_{33} and submergence and apply the equation of motion derived above.

In order to satisfy these guidelines, the author used a combination of the commercial software packages MATLAB and FEMLAB to generate panel meshes and run the panel code. FEMLAB is a finite element method program that interfaces with MATLAB to extend MATLAB's differential-equation solving capabilities to three-dimensional partial differential equations. FEMLAB was not used in the finite element method sense for which it was designed. It merely provided a useful tool for generating high quality panel meshes in a form compatible with MATLAB, in which all other programming was performed.

2.3.2 Modeling technique using FEMLAB

FEMLAB uses a Graphical User Interface (GUI) that allows the user to define geometry for solving partial differential equations models in one, two, or three dimensions. For three-dimensional geometries this is a boundary modeling technique in which a solid is defined by its boundaries. Models may be created using a variety of Computer Aided Drafting (CAD) tools within the FEMLAB interface. There are a number of simple geometric objects, such as blocks, cones, cylinders, ellipsoids, and spheres, which can be manipulated and combined using Boolean operators to create composite solid objects. These tools were particularly useful for numerical validation of the panel code results for simple geometries like the sphere or spheroid for which analytical results exist (see section 3.1). Additionally, other objects can be created by manipulating curves and line segments on a two-dimensional work plane and extruding or revolving the two-dimensional shape to create a solid object. A combination of these techniques was employed to create the necessary double body geometries.

The construction of the REMUS model will be described to show the general process used to create panel meshes. Other geometries can be handled in a similar way. The REMUS model was created by drawing a curve on a work plane to represent an offset table of the hull radius from the longitudinal centerline, then revolving about the centerline to create the basic torpedo hull shape without any of the protruding fins or transducers. The REMUS shape consists of the following sections: a spheroid nose, a cylindrical parallel midsection, and a tail section of Myring-B shape [9]. Figure 2.2 shows the profiles of these three sections. The elliptical nose

24

and rectangular midsection profiles were simply created, but the tail section (highlighted in gray) presented a greater challenge.
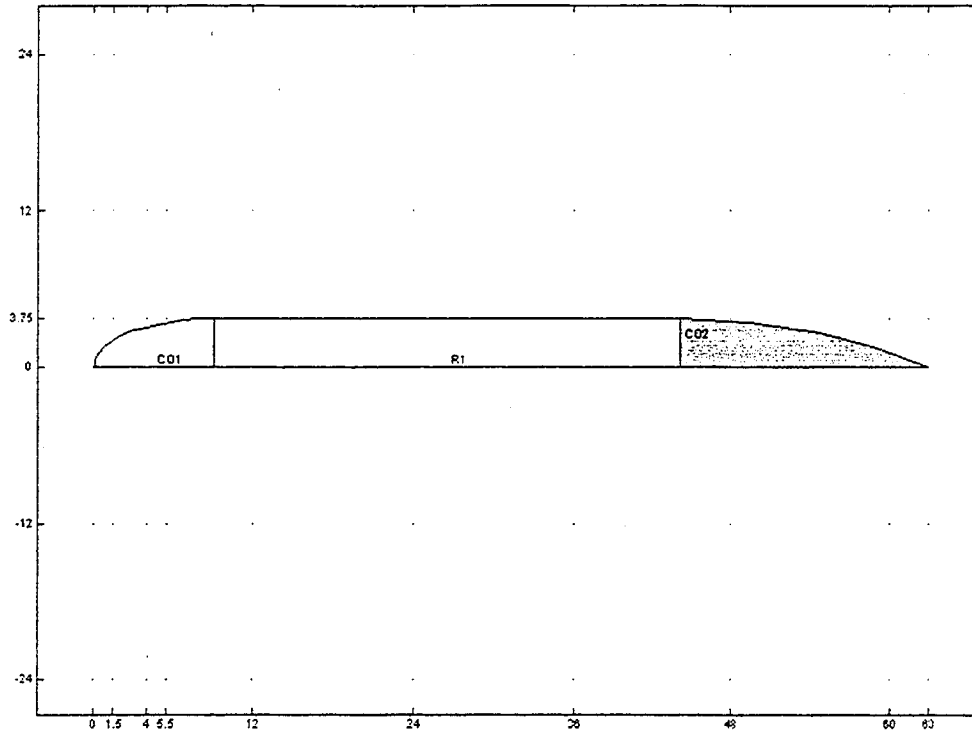
The Myring-B shape is defined by an equation of 2nd- and 3rd- order in the longitudinal variable, with coefficients determined by the maximum hull diameter, length of the tail, and the tail semi-vertex angle.   It is not possible to create a curve parametrically in the CAD environment, so a number of points were plotted and a 3rd- order rational Bezier curve was used to create a continuous curve approximating these points from the midbody section to the end of the tail. The 3rd-order Bezier curve is the highest order curve available in the FEMLAB CAD environment. Once the entire profile is created, the parts are combined using the Boolean union operator. The next step is to revolve the profile about the longitudinal axis to create a solid geometry object, as shown in Figure 2.3. Note that the resulting body is divided into a number of subsections. There is an edge dividing the nose-, mid-, and tail-sections, and each of these

sections is subdivided into four quadrants. These subsections are important both due to their symmetry and because they are related to one of the failure modes, (see subsection 2.3.3).

Figure 2.3: FEMLAB 3-D solid model of REMUS
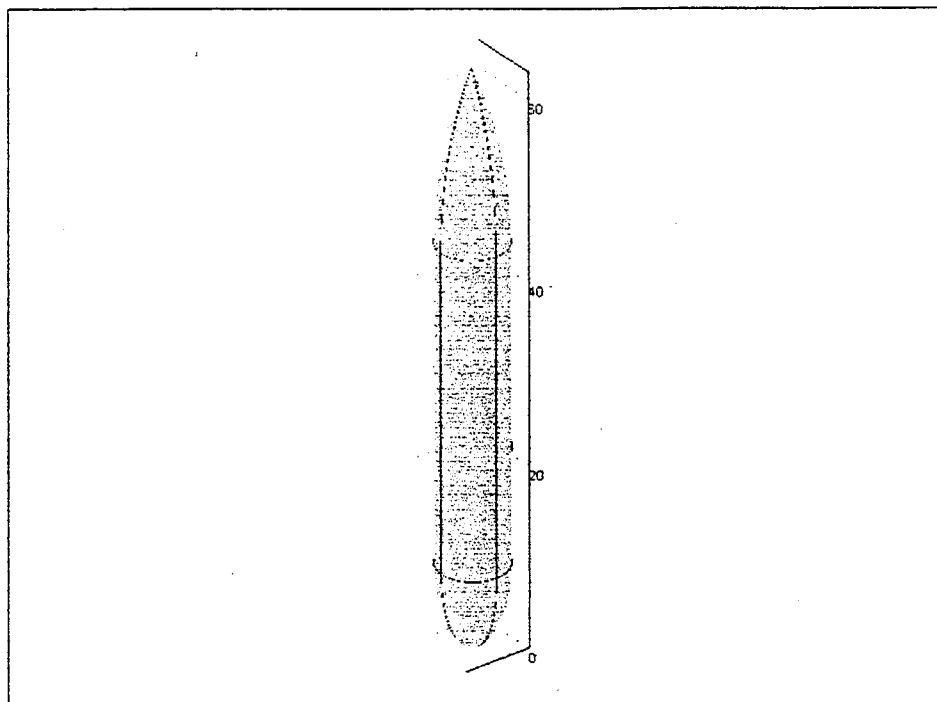
If the model construct is translated such that the origin coincides with the body center of mass, the resulting panel mesh can be input to the panel code to calculate the six-by-six symmetric added-mass matrix for the fully submerged body in the body-centered coordinate system. However, the goal is to automatically generate panel meshes for surface penetration, which is more complex. Free-surface effects can be modeled by performing additional operations prior to mesh generation: the body can be moved or rotated in the three-dimensional environment, and Boolean operations can be performed on the solid models. A block of appropriate size can be defined with a top surface at $z = 0$ to represent the constant flat free-surface assumed by the von Karman method. The body can be rotated to the desired impact angle and translated to the position of desired submergence. The wetted portion is isolated by performing an intersection operation on the body with the block representing the mass of water below the free-surface, as shown in Figure 2.4. It is possible to continue the manipulation in the

FEMLAB environment to produce the required double body. Because the body is axisymmetric, a double body may be created by rotating a copy of the cut portion and merging it with the original. This method tended to cause numerical errors in FEMLAB's handling of the object boundaries and internal mesh routine, so it was abandoned in favor of manipulation in MATLAB.

Figure 2.4: FEMLAB model of REMUS at -45 degrees, partially submerged, before and after intersection
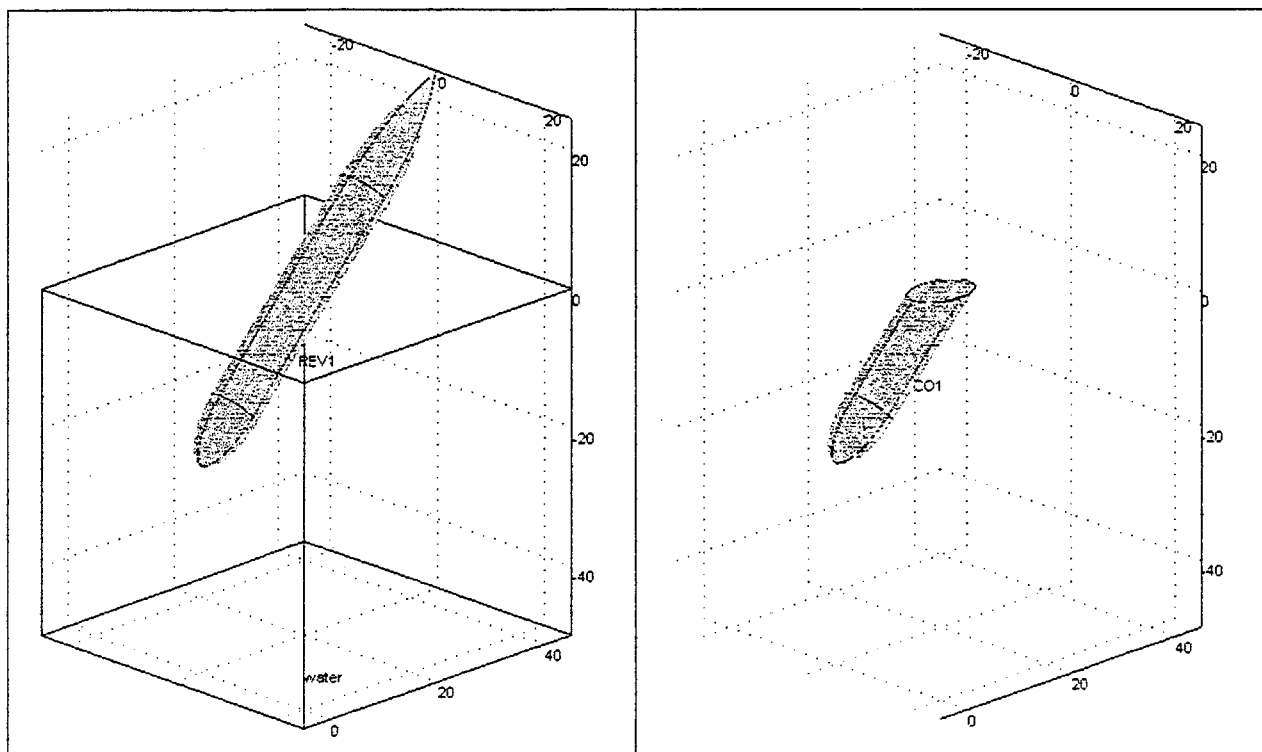
The body is cleanly cut by the flat free-surface, and eight of the original twelve subsections remain, plus one additional subsection for the plane of the free-surface. FEMLAB can create a mesh of the submerged portion of the model representing the wetted surface of the body. The author used the predefined mesh sizes available from the Global tab of the Mesh Parameters menu in the GUI. The predefined mesh sizes control the following parameters: *maximum element size scaling factor*, *element growth rate*, *mesh curvature factor*, and *mesh curvature cut off*. These parameters control how element size changes from one element to adjoining elements in response to local curvature. The reader with access to FEMLAB is

27

referred to the section "Creating Meshes in 3D" in the software documentation. For three-dimensional objects, FEMLAB creates a volume mesh by distributing vertices within and on the surface of the boundary model. The vertices are connected to fill the volume with tetrahedrons. FEMLAB attempts to optimize element quality, a measured of how close the elements are to equilateral. The surface manifestation of the internal volume mesh is a continuous distribution of nearly equilateral triangles, shown in Figure 2.5. This panel mesh meets the first four desired properties.

Figure 2.5: Initial FEMLAB panel representation of partially submerged REMUS
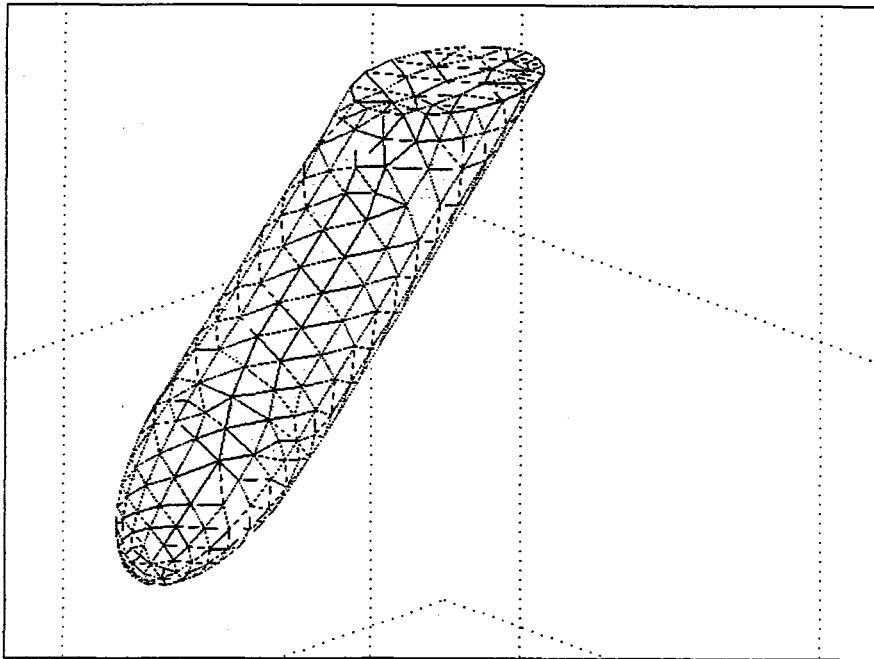
The mesh can be exported to MATLAB as a structure array that contains fields with the essential information about the object. The example shown here was created using the "Coarser" setting, which is the third of nine predefined mesh sizes. This produced a mesh with 455 surface vertices and 648 boundary elements that can be extracted from the structure array. Each vertex is an (x, y, z) coordinate, and each panel is defined by a set of three vertices. This mesh can be manipulated in the MATLAB environment.

The fifth property of interest is the symmetry of the panel mesh. Close inspection of the panels shows that the model divisions (see above) were respected by the panel generation, so that the panels of each subsection can be handled separately. Thus it is possible to take advantage of port-starboard symmetry as well as symmetry at the free surface. Half of the panels were discarded and replaced by a mirror image about the $y = 0$ plane. Next, the panels on the free surface cut are discarded and all the remaining panels are reflected across the $z = 0$ plane. Figure 2.6 shows the resulting double body, consisting of 1088 panels.

Figure 2.6: Symmetric double body
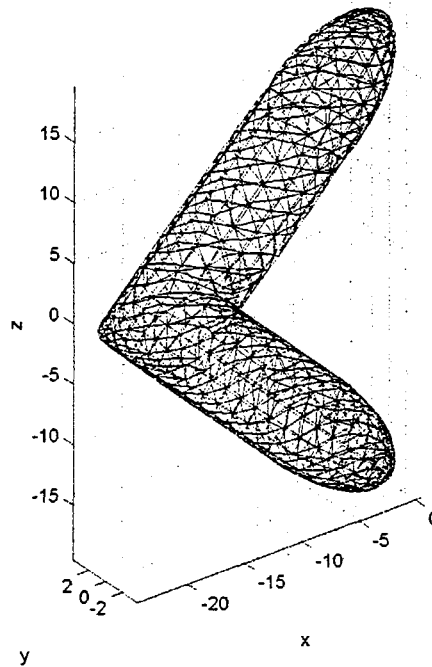
From this point, the panels are translated to a reference frame centered at the center of mass of the body, but otherwise aligned with the previously described axes. The center of mass is a constant point determined by the mass distribution of the actual REMUS vehicle, so the origin of this coordinate system is often external to the volume enclosed by the panels, especially for small submergence values, as shown in Figure 2.7.

Figure 2.7: Small submergence double body with origin at REMUS center of mass
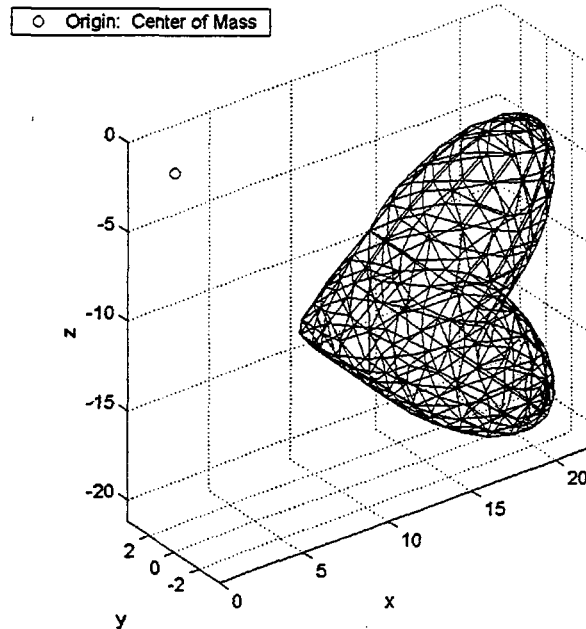
The sixth desired property is automatic panel generation for specified variable, such as pitch angle and submergence. This is possible for the above technique because FEMLAB is designed to interface with MATLAB. FEMLAB 3.0 is the first release that does not require MATLAB to run, but the connection is still important. All of the FEMLAB operations shown above were originally completed using the Graphical User Interface. However, FEMLAB is capable of creating an m-file script to reproduce the steps performed in the GUI. The script can be edited in MATLAB, where specific values of submergence, pitch angle, and parameters like mesh size can be replaced by variables. Another m-file is written to specify these variables in *for* loops and run the mesh generation script and the added-mass panel code for a large number of variations.

2.3.3 Known problems with the meshing technique

Only one m-file script should be necessary for each body shape. However, this is not the case due to internal errors within FEMLAB that may occur when the body is intersected with the block representing the water mass. The script fails, and no solid model is created. Failures occur for very small submergence values if a block is specified large enough to fully enclose the body at full submergence for any pitch value, as is necessary for a general script. Specifying smaller block sizes for specific pitch angle and submergence values solved this problem. For each desired pitch angle, two or three files are created to cover the full spectrum of submergence. An example is listed in Appendix A.2 for a REMUS at a pitch of -45° (nose below the horizon) for submergence between 10 and 46 inches, measured from the lowest point of the body. It is necessary to call different mesh scripts using *if* statements, as shown in Appendix A.1. Appendix A.3 provides the script for manipulation of the initial mesh from FEMLAB and for running the added-mass panel code.

The second failure mode for this technique is related to the previously discussed model subsections (see subsection 2.3.2 above). This mesh generation technique was developed so that a panel mesh could be automatically generated for a specified pitch angle and submergence. For any given angle, if the submergence values are spaced closely enough, at certain steps the solid model will be positioned such that very small portions of one or more body section will be left after the intersection operation with the block representing the water mass, as shown in Figure 2.8. In this instance, one of two errors may occur. First, FEMLAB can have an error processing the intersection command, which causes the mesh generation script to fail in a way similar to that described above, where no new object is created. Alternately, a new shape may be created, but the resulting panel mesh may be unacceptable. Panels may not cross the edges that define the body subsections, but the *element growth rate* defines the rate at which panel size can vary between adjacent panels, even if they touch across a subsection edge. Thus, if a body subsection is too small, the panel mesh will have small panels in that section and grow to the specified maximum element size far from that subsection, as shown in Figure 2.9. This panel mesh has more panels, but because they are neither uniform in size nor uniformly distributed, they do not lead to a better representation of the body. This mesh violates the requirement that panel size be fairly uniform, and therefore causes numerical error when the mesh is input to the added-mass panel code.

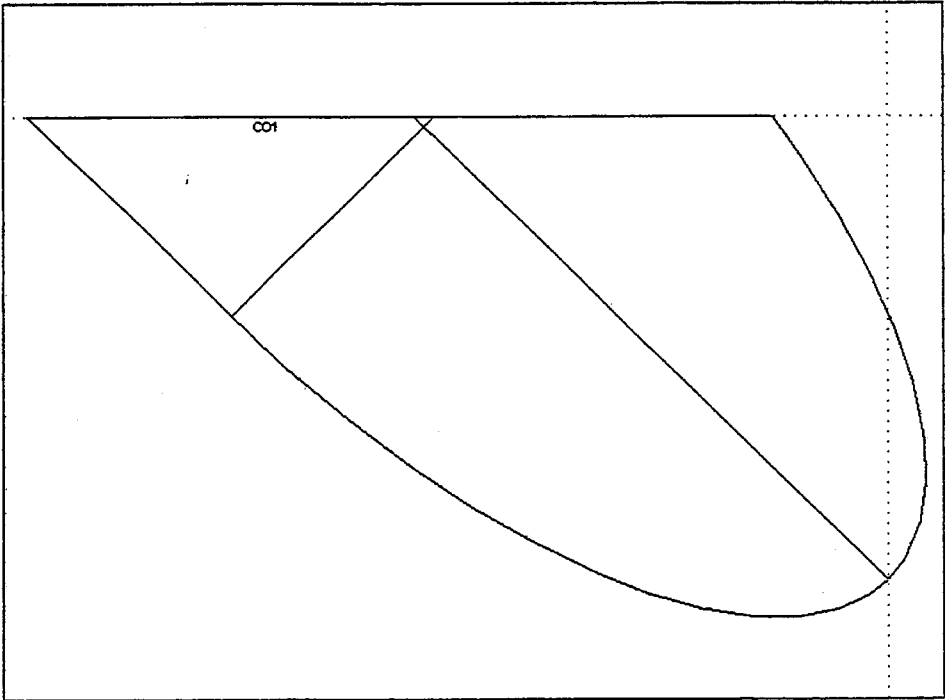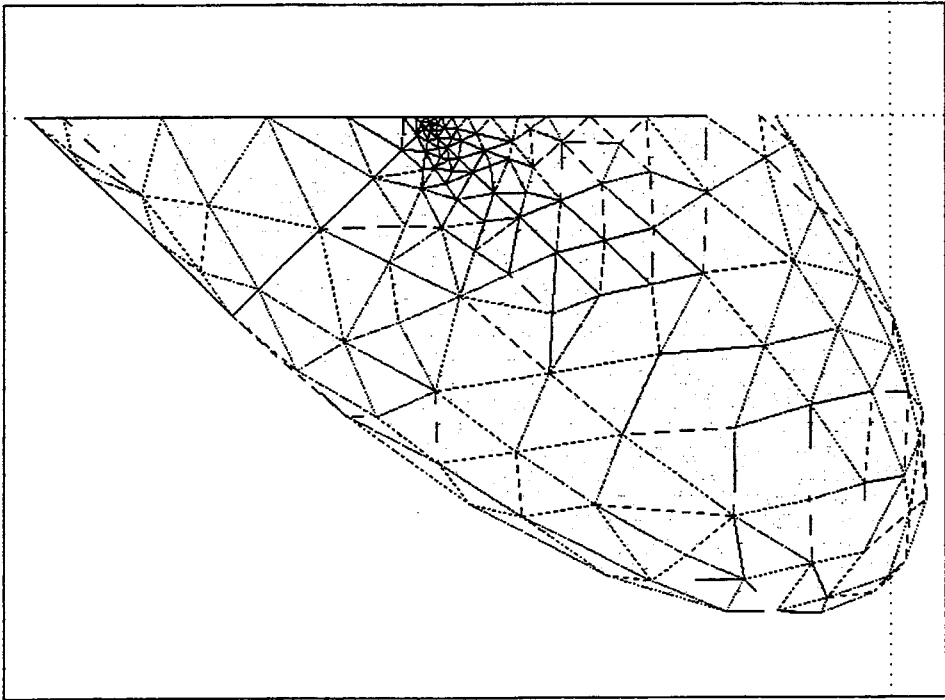Figure 2.8: Body intersected too close to subsection edge

Figure 2.9: Resulting mesh with unevenly-distributed panels

When a subsection boundary failure occurs, no panel mesh is produced, so added-mass data is missing for this submergence value. The second type of error does not cause a script failure, so added-mass data exists for this value, but may be inaccurate. A third type of error occurs when a panel mesh is created that is flawed due to incorrect panel placement in areas of the body with high curvature. Sometimes the panel generation connects the panel vertices through a region of high curvature such that these panels cut into the internal volume of the body (see Figure 2.10). Again, inaccurate added-mass data exists for this point. Visual examination of the panel meshes permits identification of the second and third types of error and the exclusion of outlying data points related to these errors.

Figure 2.10: Flawed panel mesh
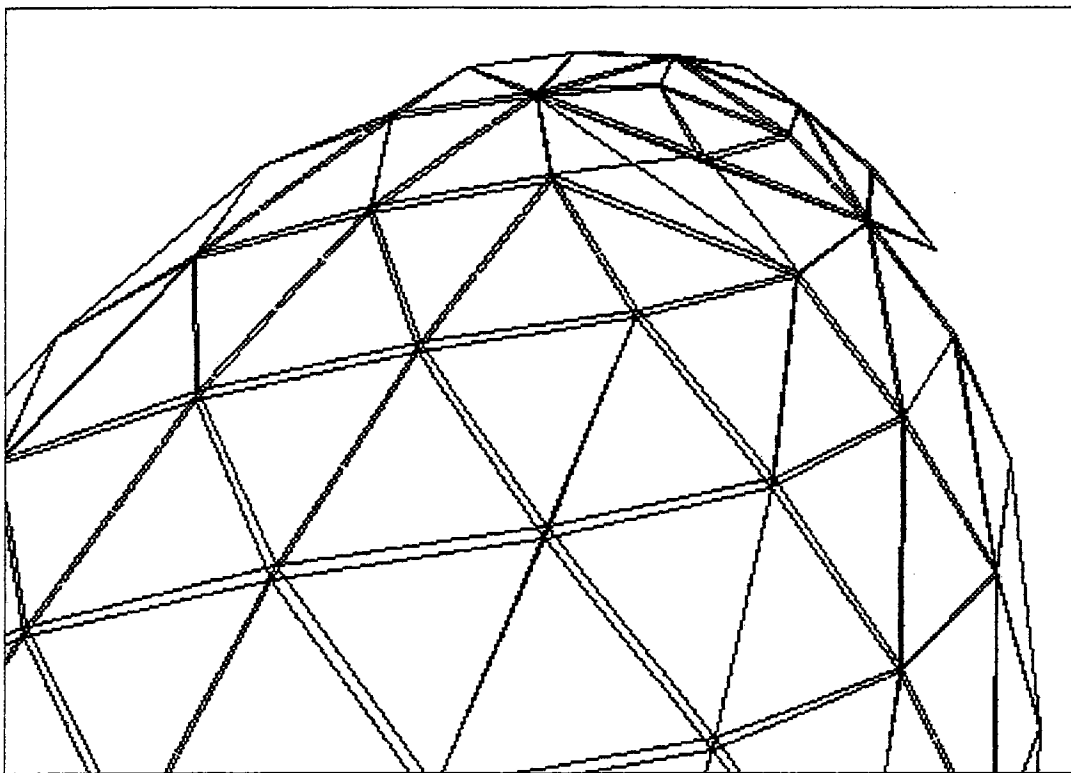
Chapter 3

Numerical Validation and Results

In this chapter I will show the added-mass coefficients calculated by the source collocation panel code with panel meshes created by the FEMLAB mesh generation technique. Numerical results are compared to analytical solutions that exist for simple shapes. Results for REMUS are presented for several pitch angles at the end of the chapter.
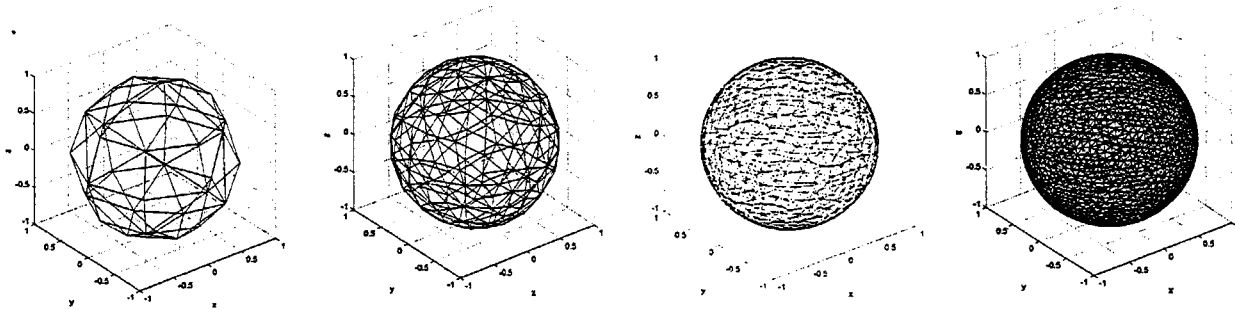
3.1 Validation of Added-mass Coefficients

Chapter 2 showed the development of a numerical method to calculate added-mass coefficients for bodies of arbitrary shape. It is necessary to compare these coefficients against analytical results prior impact acceleration predictions based on the coefficients. Analytical solutions exist for simple three-dimensional solid objects, such as ellipsoids, spheroids, and spheres. These objects are easily modeled using the FEMLAB/MATLAB meshing technique detailed above.

3.1.1 The Fully Submerged Sphere

The simplest three-dimensional geometry is the sphere. Due to symmetry, the non-zero added-mass properties can be expressed by a single coefficient:

$$m_{11} = m_{22} = m_{33} = \frac{2}{3}\pi\rho r^3 = \frac{1}{2}\rho V. \qquad (3.1.1)$$

Figure 3.1: Four panel representations of a sphere, with 64, 320, 1284, and 3784 panels respectively

The author's goal is to demonstrate the convergence of the three numerically-calculated added-mass coefficients with increasing number of uniformly distributed panels. Figure 3.1 shows a progression of sphere panel meshes generated by four of the predefined mesh sizes in FEMLAB. Symmetry was maximized by reflecting the panels from one quadrant of the initial FEMLAB panelization of the sphere across the three equatorial planes. Results are nondimensionalized with respect to the displaced volume of a sphere and the added-mass coefficients converge to a value near one half (see Figure 3.2). The value for the largest number of panels (3784 panels) is 0.509, an error of approximately two percent. This error is assumed to be due the discrete panel solution method, but it is not known how much the error could be reduced by the addition of more panels. Note that in the momentum formulation of the equation of motion, the dominant term is the effect of the spatial derivative of the added-mass. If the discretizing error is steady with submergence, it will have a minimal effect on the calculated impact force and acceleration.

Figure 3.2: Convergence of added-mass coefficients of a sphere vs. number of panels
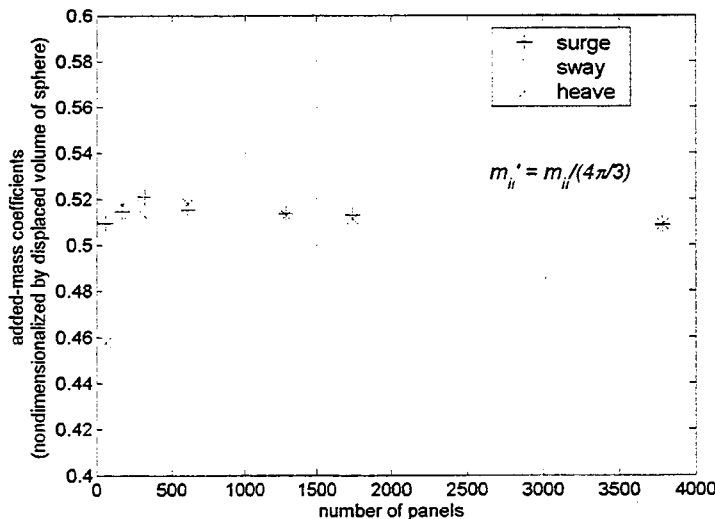
3.1.2 The Fully-Submerged Spheroid

Newman presents results for the added-mass of the fully submerged spheroid, or ellipsoid of revolution. A spheroid can be oblate or prolate depending on the axis of revolution. Figure 3.3 shows a range of oblate and prolate spheroids. Figure 3.4, reproduced from Newman's text, shows the three distinct nonzero added-mass coefficients as a function of the ratio between the semilength of the axes, nondimensionalized respectively by the mass and moment of inertia of the displaced volume. Figure 3.5 shows numerical results for the same range of spheroids, generated by FEMLAB using the "Coarser" setting. Note that for these fifteen objects, the number of panels varies from 232 to 1560 panels. This is because more panels are required to accurately portray the change in curvature near the ends of the long, thin prolate spheroids than for the nearly spherical bodies on either side of $b/a = 1$. More panels are also required at the opposite extreme, where the oblate spheroid tends toward the flat plate.

Figure 3.3: Panelizations of various oblate and prolate spheroids
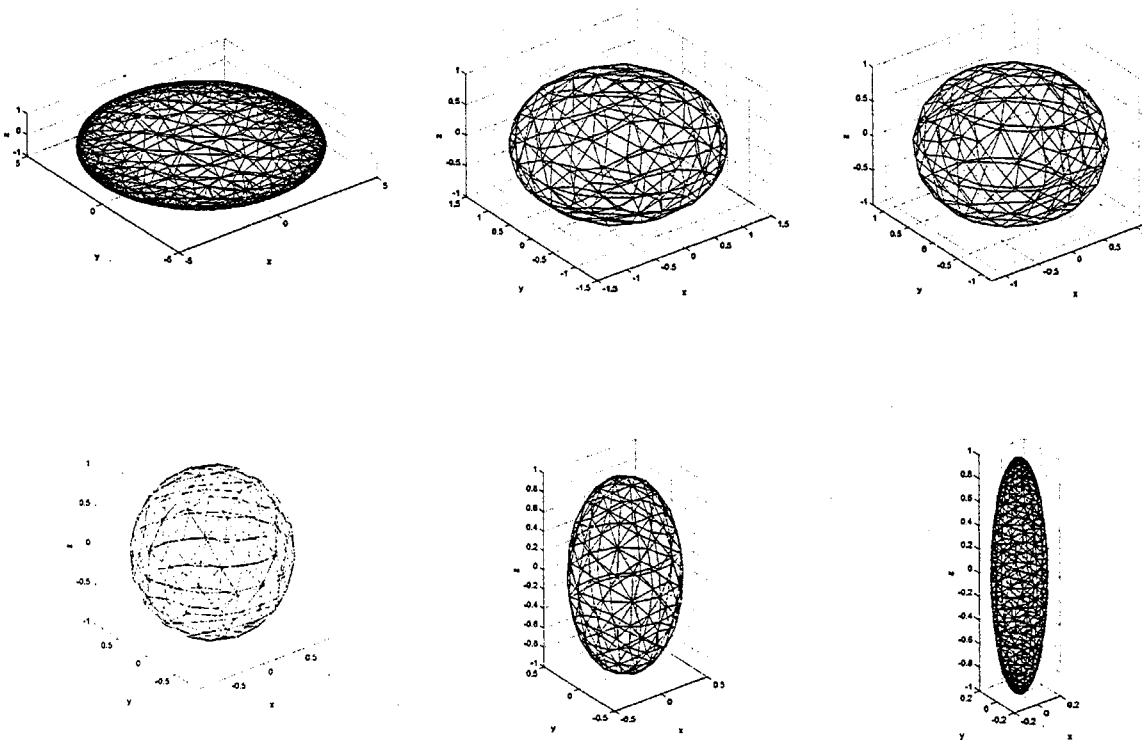
Figure 3.4: Analytical added-mass of spheroid from Newman [10]

$$m_{11}' = \frac{m_{11}}{\frac{4}{3}\pi\rho ab^2}$$

$$m_{22}' = \frac{m_{22}}{\frac{4}{3}\pi\rho ab^2}$$

$$m_{55}' = \frac{m_{55}}{\frac{4}{15}\pi\rho ab^2(a^2+b^2)}$$
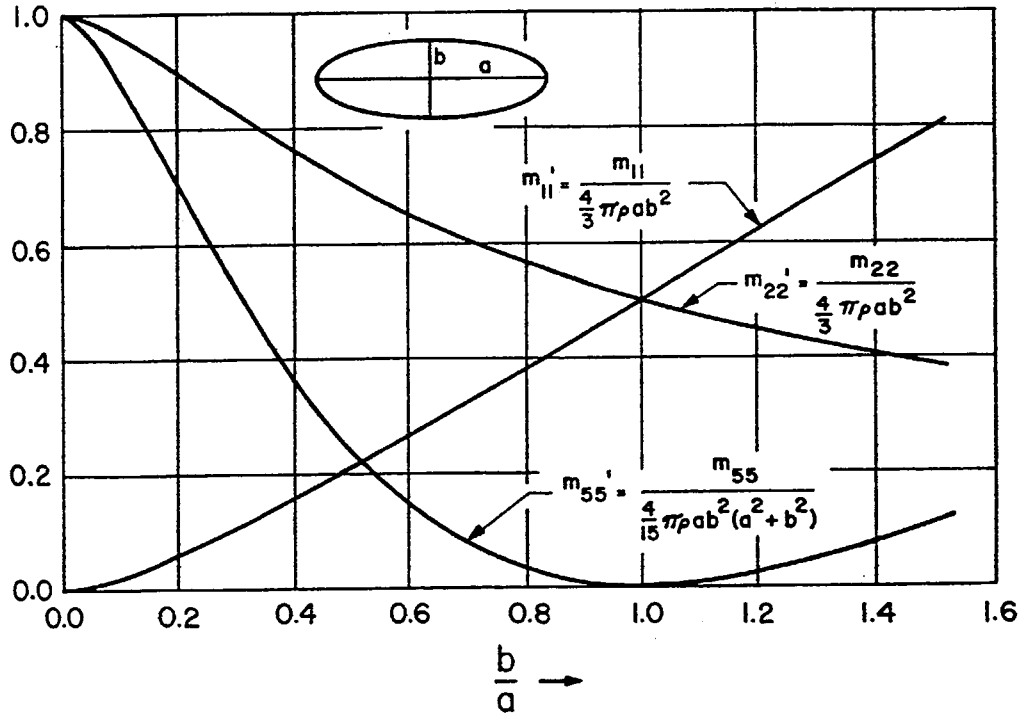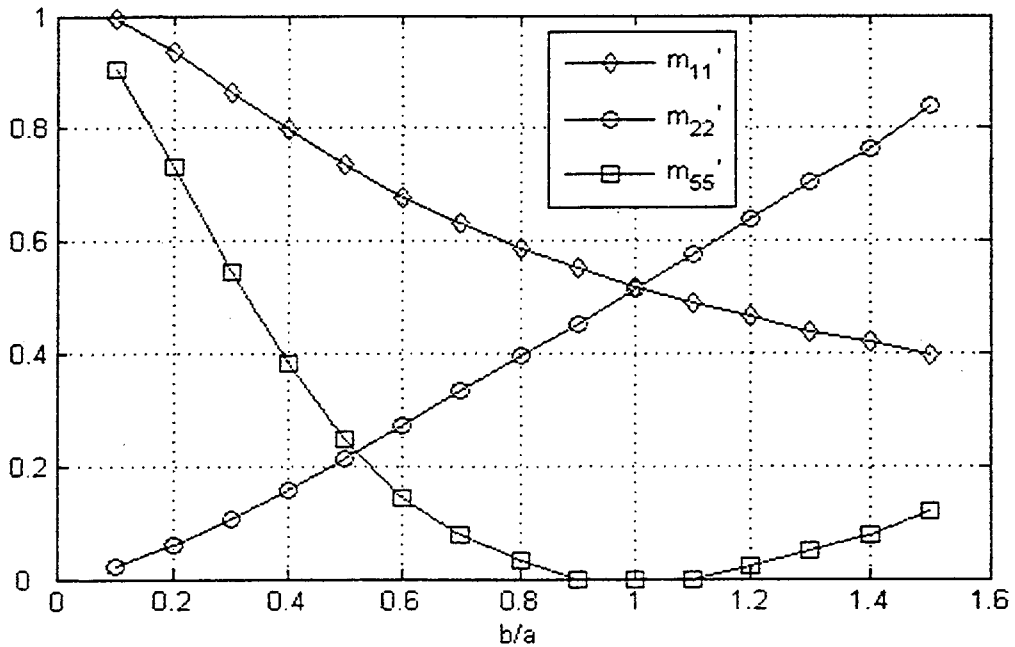
$$\frac{b}{a} \longrightarrow$$

Figure 3.5: Numerical added-mass results for spheroid from FEMLAB tests

b/a

3.1.3 The Sphere as a function of Submergence

The examples above demonstrate that it is possible to accurately calculate the added-mass coefficients of fully submerged bodies. A more important test is whether or not the method can calculate added-mass as a function of submergence. Miloh analytically solved the horizontal added-mass of a sphere penetrating the free-surface by representing the double spherical bowl with toroidal coordinates [6, 7, and 8]. In his formulation, $b(t)$ is the absolute submergence of the lowest point of the sphere, and τ is nondimensionalized by the radius R:

$$\tau = b(t)/R. \tag{3.1.2}$$

Miloh uses two different terms for the nondimensional vertical added-mass coefficient:

$$C_2 = \frac{output_{33}}{\pi R^3} = \frac{a_{33}}{2\pi\rho_w R^3} \tag{3.1.3}$$

$$\lambda_2(\tau) = \frac{3C_2(\tau)}{(3\tau^2 - \tau^3)}. \tag{3.1.4}$$

Note that these equations mix notation from several of Miloh's papers with that defined above. The term $output_{33}$ refers to the heave output of Dr. Kim's added-mass code. In these expressions, λ_2 is defined with respect to the submerged volume of the sphere.

The next two figures show the comparison of the analytical and numerical solutions for the coefficient λ_2. Numerical results were calculated for 100 evenly-spaced submergence steps. Note the two highlighted data points in Figure 3.9. These are the two points for which Miloh gives exact solutions. The point $\tau = 1$ is the sphere half-submerged in the free surface, corresponding to the entire sphere fully-submerged. This result, show earlier, is 3.6 % off the exact value of ½ (a smaller number of panels was used that in subsection 3.1.1, resulting in greater error). The point $\tau = 2$ is the fully-submerged sphere with top surface tangent to the free surface. The numerical result differs from the listed exact value of 0.3552314 by 2.4 %.

Figure 3.7 is displayed to show the slight noise in the coefficient C_2 that was not visible in Figure 3.9. This noise is a non-physical result, and is assumed to be the result of representing the body by discrete panels. While the results may be consistently accurate within about four percent, this fine-scale irregularity from one point to the next can be a problem in the calculation

of the first derivative of the added-mass coefficient. The first derivative determines the dominant term in the momentum-based equation of motion, expressed in Equations 2.1-2.7. The derivative of C_2 with respect to the nondimensional submergence is shown in Figure 3.10 from [8] (with an error of a factor of two from [6]) and again in Figure 3.11 as calculated two ways. The noisier curve is the result of taking the first difference of the C_2 data. The smoothed curve is the result of approximating C_2 with an eleventh-order least squares fit of the data, accomplished using the MATLAB "polyfit" function. The function "polyder" produces the tenth order derivative.

Figure 3.6: Examples of double spherical bowl meshes
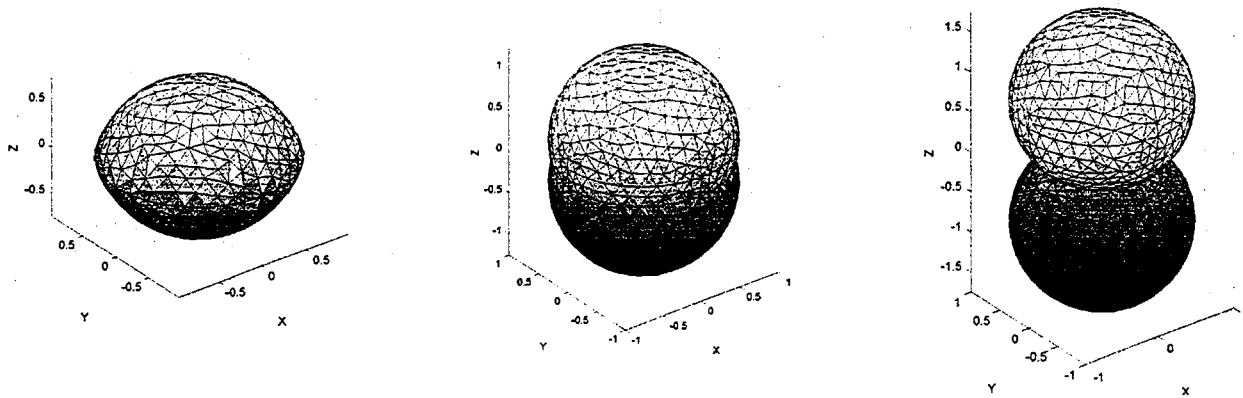
Figure 3.7: Numerical results for coefficient C_2
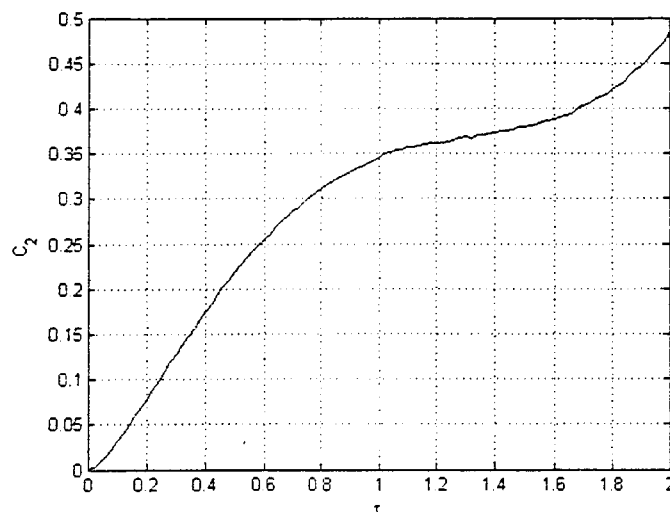
Figure 3.8: Analytical results for coefficient λ_2 from Miloh [8]
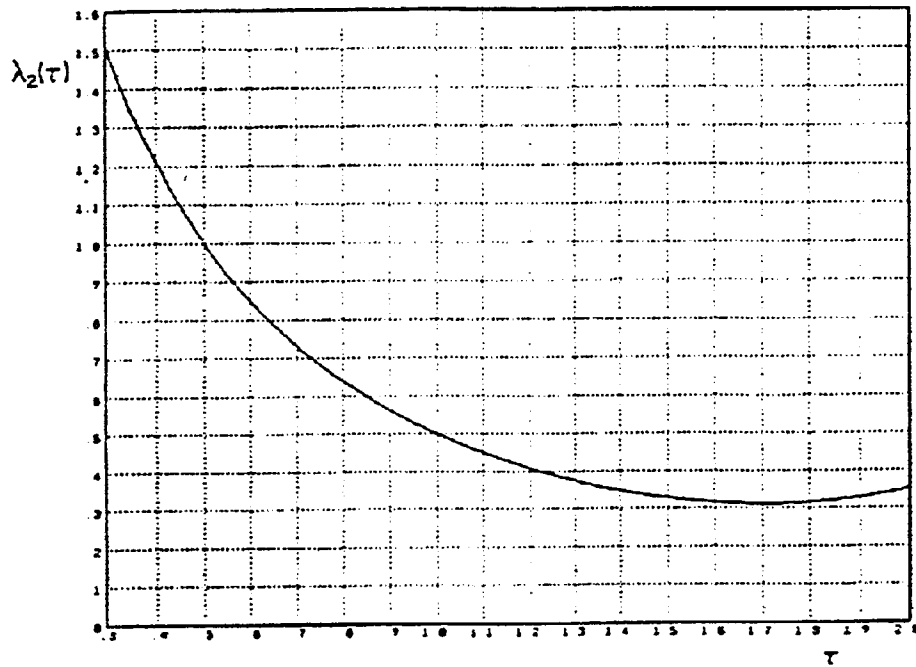

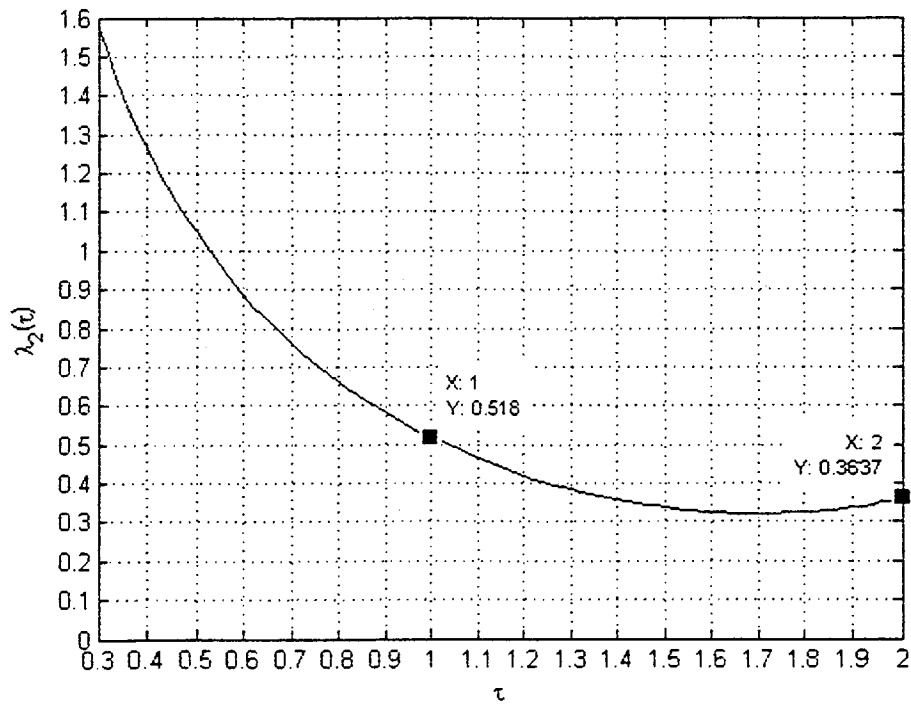Figure 3.9: Numerical results for coefficient λ_2

Figure 3.10: Analytical results for derivative $dC_2/d\tau$ from Miloh [8]

(note that this in this figure $dC_2/d\tau$ is two times the value from a previous publication [6])
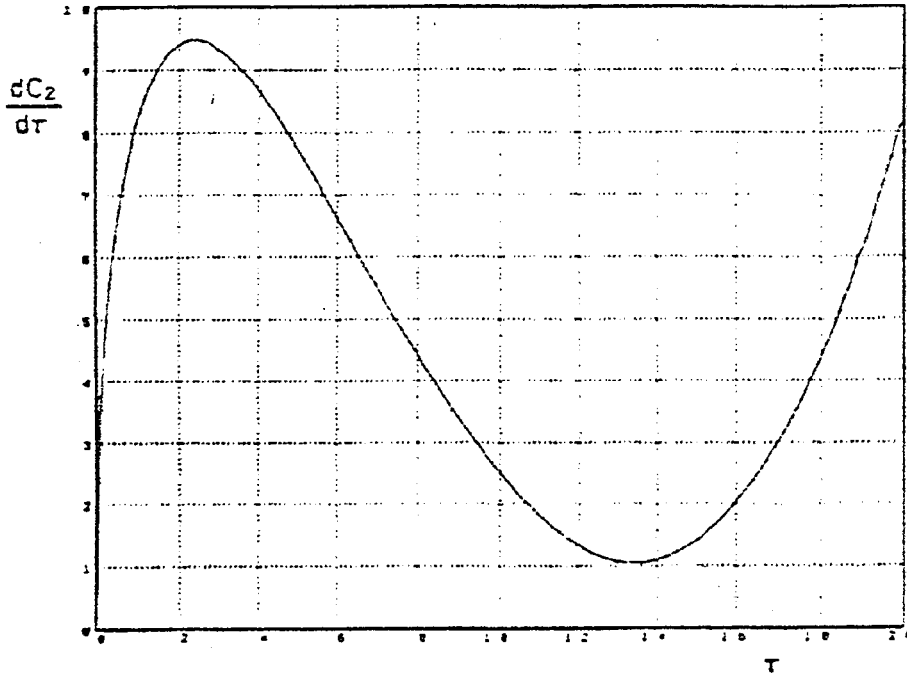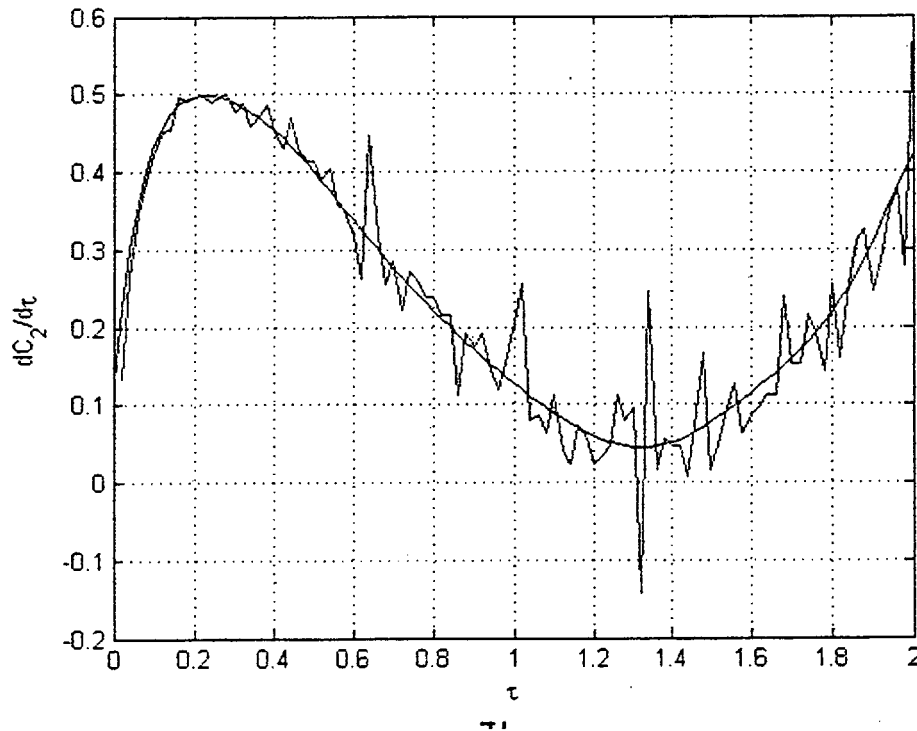

Figure 3.11: Numerical results for derivative $dC_2/d\tau$

This technique performs a smoothing function that does not give undue weight to outlying data points. Second, it expresses the added-mass data as a continuous function (with a smooth derivative) over the interval of interest. This function can be evaluated at any valid submergence value at extremely low computational cost, compared to alternate techniques, such as table look-up (interpolation) methods. This is important when the data are incorporated into the equation of motion model, then integrated using a standard Ordinary Differential Equation solver like MATLAB's ODE45 function. Third, added-mass data with uneven submergence intervals can be used, such as when the mesh generation fails for certain submergence values, as described above. The major weakness of this technique is that the user must make a judgment about which order polynomial best fits the data. There is no theory which predicts the form of these added-mass curves for complex bodies lacking analytical solutions. An attempt was made to use the lowest order that accurately reflected the entire data set, but fit particularly well at small submergence values, where the majority of the impact effects are felt.

3.2 REMUS Numerical Results

This section presents numerical results for REMUS at various pitch angles in terms of heave added-mass and its spatial derivative, and also in terms of integrated acceleration.

3.2.1 REMUS Heave Added-Mass and the Derivative da_{33}/dz

The author calculated the REMUS heave added-mass for several values of the pitch angle, which is assumed to remain constant throughout the impact. Figure 3.12 shows the added mass for a nose-down vertical entry plotted against the nondimensional submergence τ;

$$\tau = \frac{z}{z_{max}}, \qquad (3.2.1)$$

where z is the displacement of the lowest point of the body below the free surface and z_{max} is this displacement at full submergence (when the highest point of the body is tangent to the free surface). Calculations were made at 100 evenly spaced points, but 14 of these points were rejected due to poor panel mesh properties, causing the gaps at several locations in the curve.

The REMUS profile is shown as a reference for the different regions of the graph. The largest slope occurs during the nose impact at the left of the figure at small values of the nondimensional submergence. The added-mass curve levels while the parallel midsection is entering, then increases in slope once more as the tail section narrows at around $\tau = 0.75$. This phenomenon is caused by the increasing energy required to for flow to follow the body contour as it contracts and then expands again on the double body shape. This is a non-physical result caused by the breakdown of the assumptions of the von Karman method. The free surface was replaced by a flat equipotential surface represented by the reflected image body. As more of the body penetrates the free-surface, the assumption of the equipotential plane with zero normal velocity fails, resulting in predictions of non-physical behaviors. Further results are displayed for small submergence values only.

Figure 3.12: REMUS added-mass at pitch -90 degrees, with reference profile
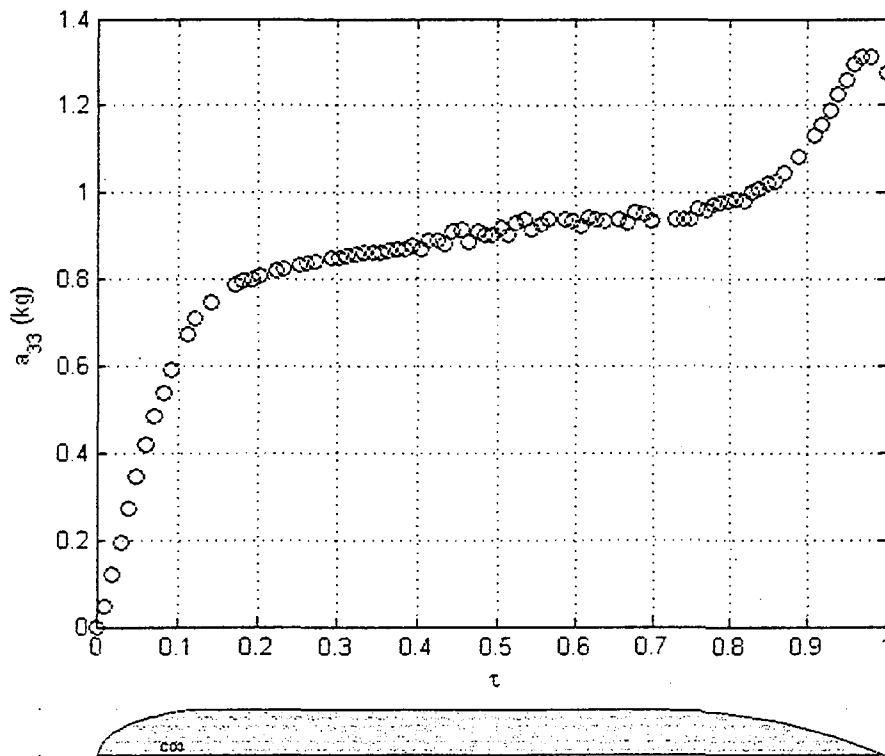
These data are intended for use in an equation of motion model to predict heave acceleration (see subsection 3.2.2), so it is useful to define the position of the REMUS model by its center of mass relative to the previously defined coordinate system (see Figure 2.1). The

variable z_{cg} refers to the displacement of the center of mass from the free surface. Initial contact occurs at $z = 0$ at a positive value of z_{cg} and proceeds to full submergence at a negative value of z_{cg}. An 11th-order polynomial is fit to the data to provide a smooth curve for a_{33} and its derivative for positive values of z_{cg} (see Figures 3.13 and 3.14).

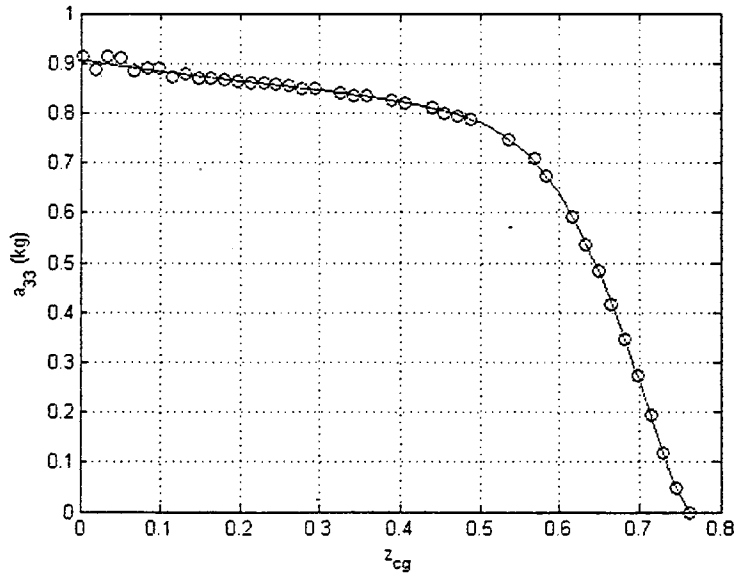Figure 3.13: REMUS added-mass for small submergence at pitch -90° with 11th-order polynomial

Figure 3.14: 10th-order polynomial derivative for pitch -90°

44
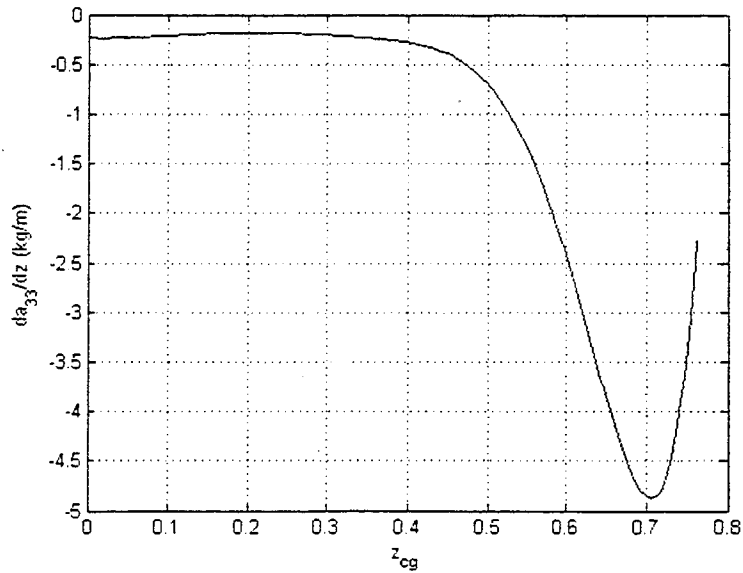
The results from the nose first (-90° pitch) impact are the limiting case in which the body orientation is most streamlined to the flow for vertical impact. The author expects impact effects to be the smallest for this case. Figures 3.15 and 3.16 present results for the other limiting case, the flat impact (0° pitch). Two factors make this case the maximum impact. First, the added-mass is greatest in the direction of the most flow obstruction. At $z_{cg} = 0$, compare a value of a_{33} (-90°) = 0.9063 to a_{33} (0°) = 16.783. Second, for flat impact, this higher value is reached in a smaller vertical distance. These two factors combine to cause the higher values of the spatial derivative $\dfrac{da_{33}}{dz}$, and thus the maximum acceleration (see subsection 3.2.2).

Figure 3.15: REMUS added-mass for small submergence at pitch 0° with 5th-order polynomial
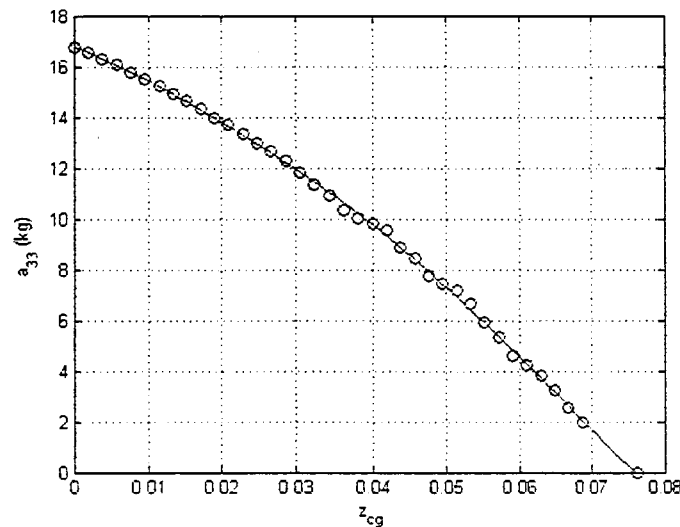
Figure 3.16: 4th-order polynomial derivative for pitch 0°
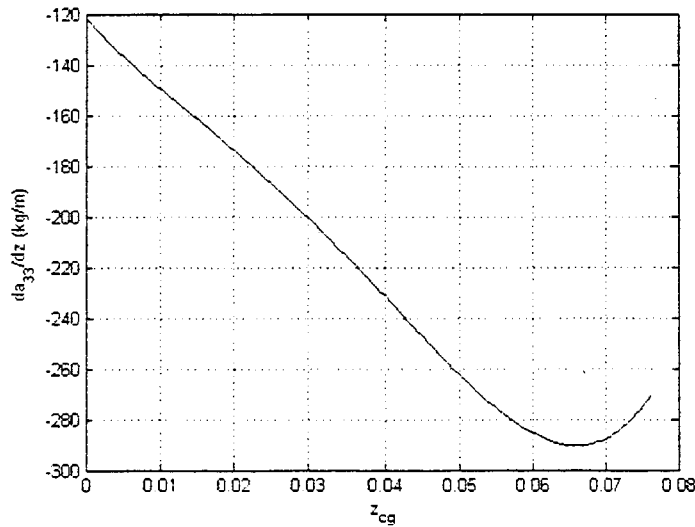
The results vary between the two extremes for intermediate pitch values. Figures 3.17 to 3.20 show results for two other angles. Note that there are more data points missing in these figures. This is due to mesh generation failures discussed in subsection 2.3.3, which are more common at angles which cause the body subdivisions to intersect the free surface plane at other than a right angle.

Figure 3.17: REMUS added-mass for small submergence at pitch -45° with 6th-order polynomial

Figure 3.18: 5th-order polynomial derivative for pitch -45°

Figure 3.19: REMUS added-mass for small submergence at pitch -75° with 7th-order polynomial

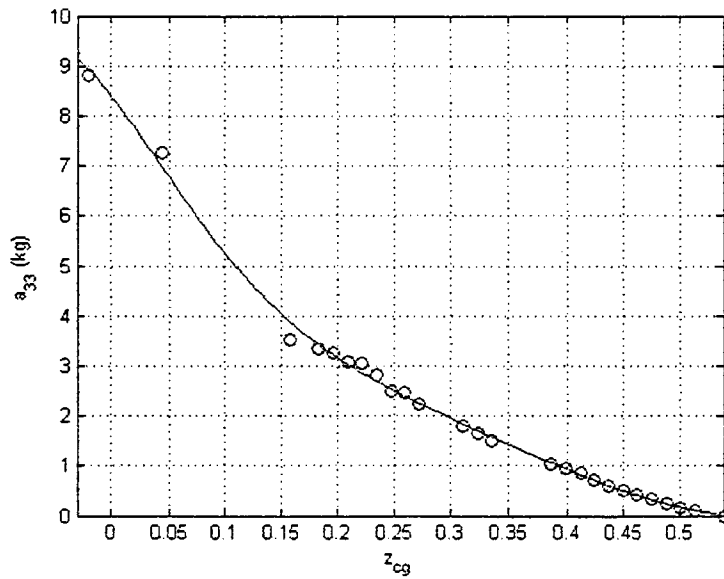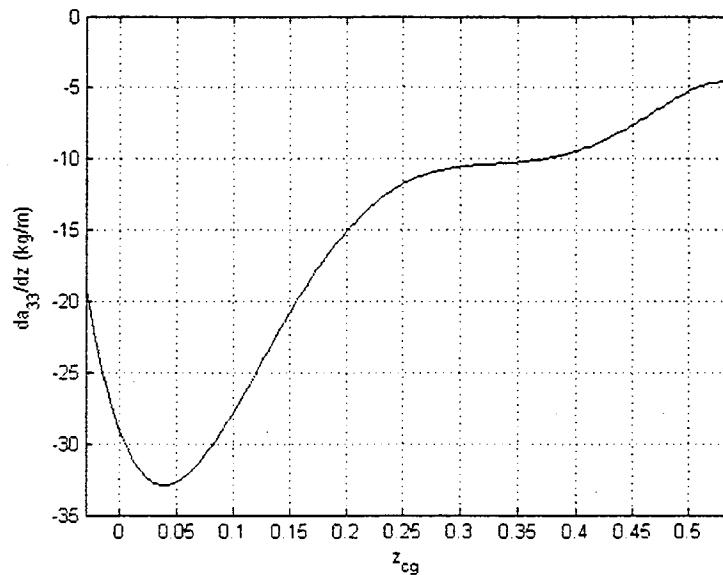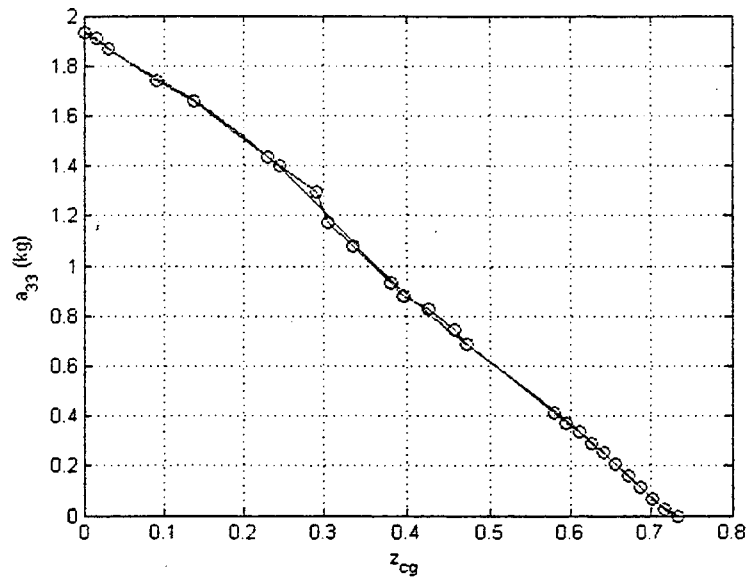Figure 3.20: 6th-order polynomial derivative for pitch -75°
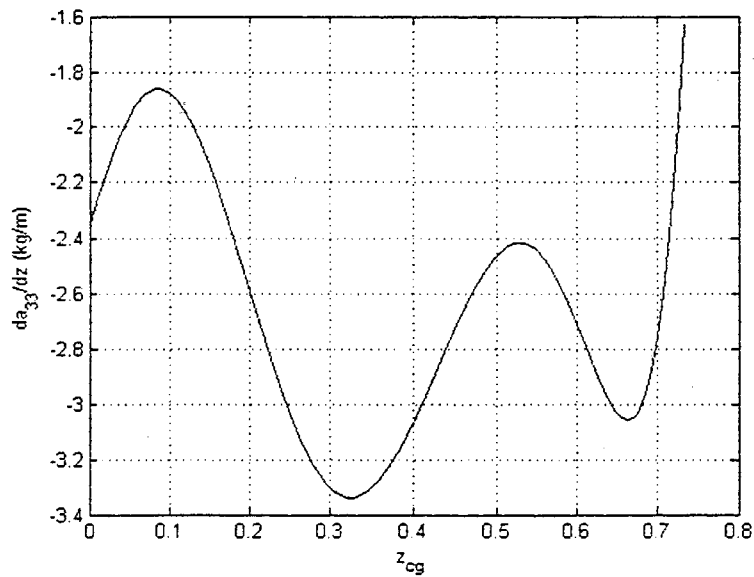
3.2.2 REMUS Heave Acceleration

The polynomial representations of the REMUS heave added-mass and its spatial derivative are used to calculate free-fall heave acceleration via the equation of motion (see

Equation 2.1.8). The MATLAB ODE113 function was used at default settings to perform the numerical integration of the ordinary differential equation with initial conditions of velocity and position. Appendix A.4 contains a function for the evaluation of the equation of motion. Acceleration was computed by taking the first difference of the output velocity data. Figures 3.21 and 3.22 show the results of free-fall from rest at starting heights of five meters and ten meters. Five meters is approximately the typical height of rescue swimmer casts from a hovering helicopter (see section 1.1.2), and ten gives a safety factor of two.

Figure 3.21: REMUS motion at -90° from starting height of 5 meters
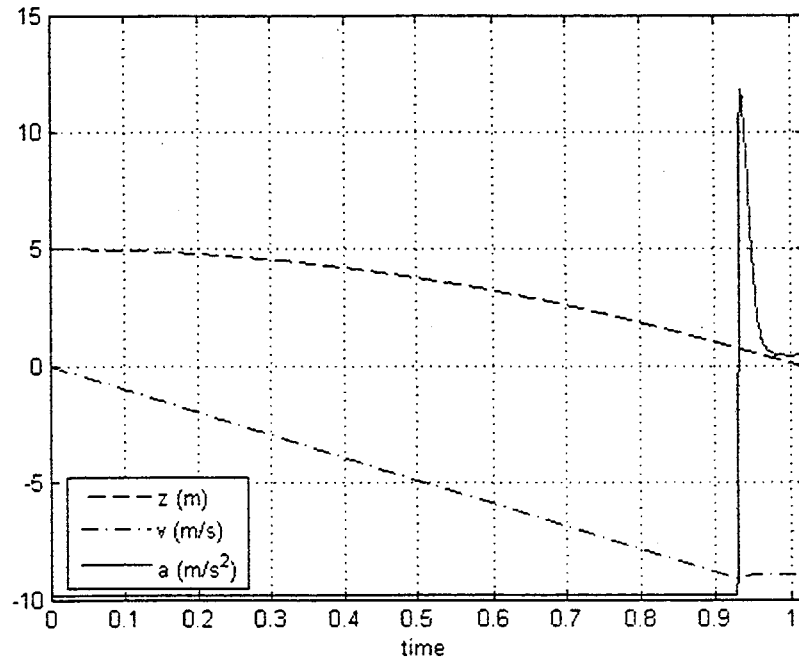
Assuming free-fall from rest, the relationship between starting height and maximum acceleration is linear impact at a constant pitch angle (see Figures 3.23 and 3.24, where heave acceleration is nondimensionalized by gravitational acceleration). For a specified starting height, the largest acceleration is for a flat impact (0°), as expected (see Figure 3.25). However, the smallest maximum acceleration is for impact at -75°, not at the expected value of -90°. The relationship between pitch angle and maximum acceleration is not clear from the data, but a range of pitch angles from -90° to -65° result in maximum acceleration of less than 2 g's from 5

meters. Chapter 4 will show the correlation between numerical and experimental results for selected initial conditions.

Figure 3.22: REMUS motion at -90° from starting height of 10 meters
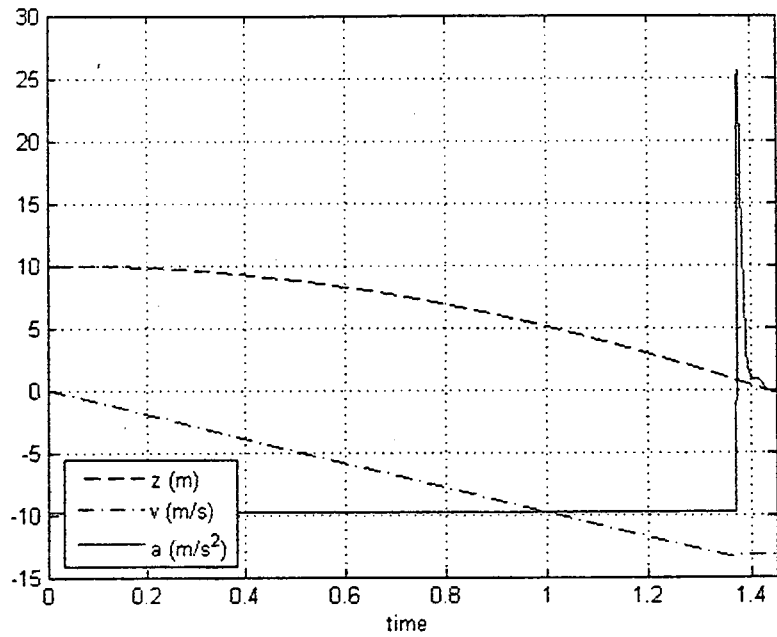
Figure 3.23: REMUS maximum accelerations (g) at -90° from various starting heights
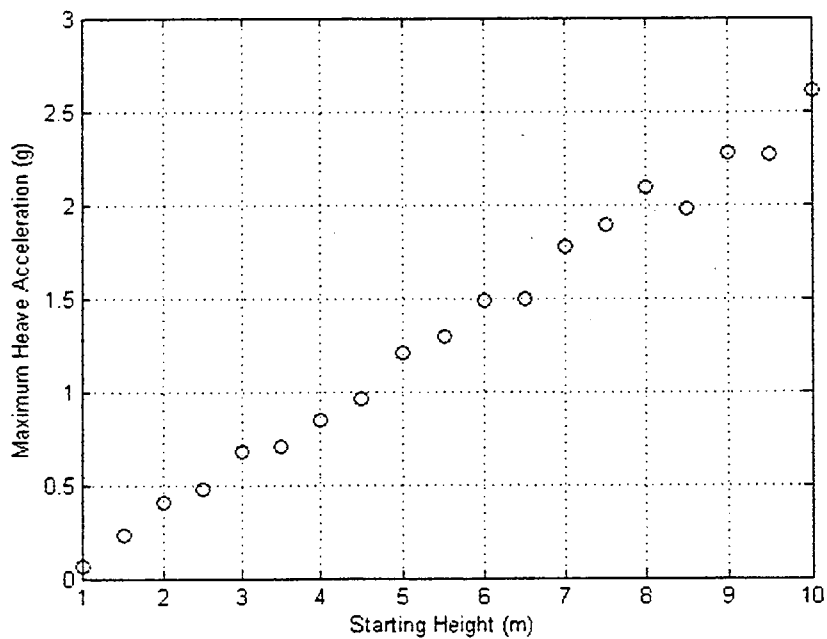
Figure 3.24: REMUS maximum accelerations (g) at 0° from various starting heights
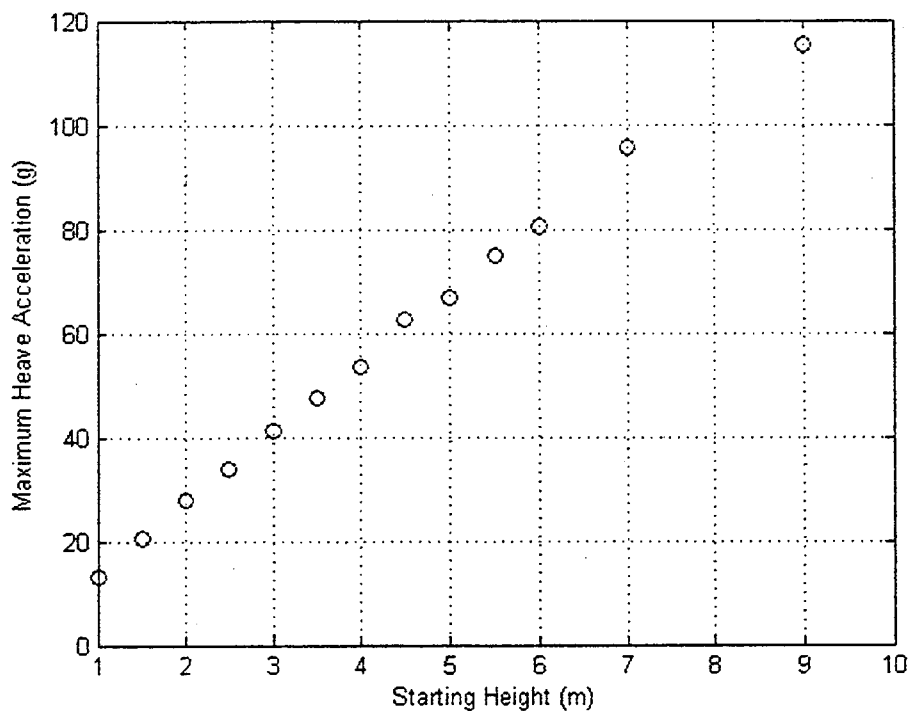
Figure 3.25: REMUS maximum accelerations (g) from 5 meters at various pitch angles
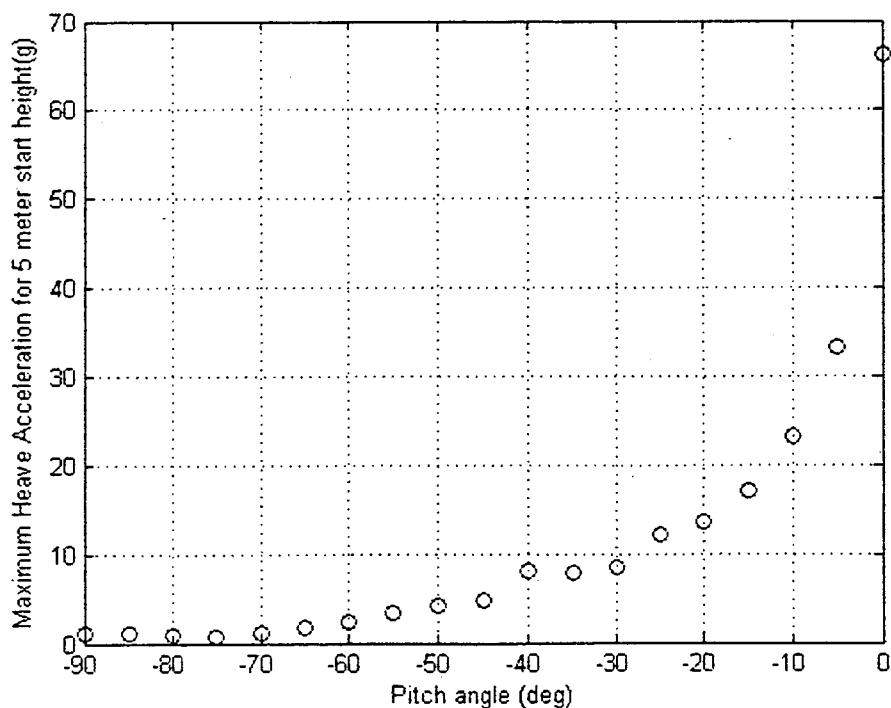
Chapter 4

Experimental Analysis

I conducted REMUS free-fall impact experiments at the Woods Hole Oceanographic Institution during the month of July 2004. This chapter will demonstrate how these experiments were conducted and correlate the results with the numerical predictions discussed in subsection 3.2.2.

4.1 Experimental Setup

4.1.1 REMUS Specifications

REMUS is a small streamlined body of torpedo shape (see Figure 4.1). Table 4.1 shows the geometric and inertial vehicle parameter important to the impact problem.

Table 4.1: REMUS parameters

Parameter	Specification
Vehicle Diameter	19 cm
Vehicle Length	160 cm
Weight in air	37 kg
Displaced Volume	35.7 liters
Nose section	Spheroid
Nose Length	23 cm
Midsection	Cylinder
Midsection Length	84 cm
Tail section	Myring-B profile
Tail Length	52 cm
Tail semi-vertex angle	25 deg

Figure 4.1: Test vehicle
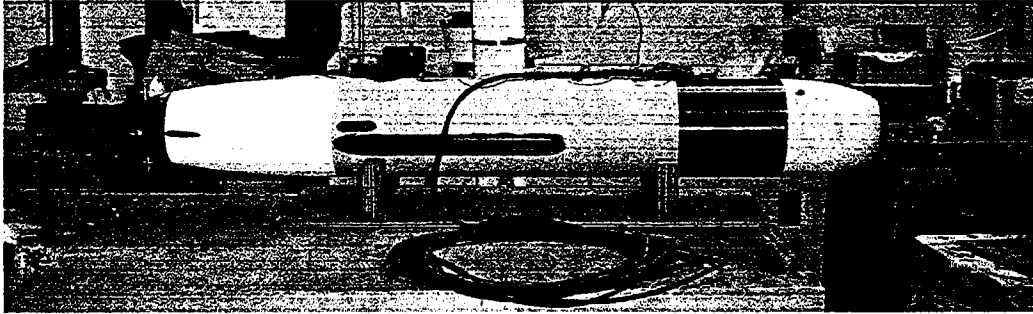
4.1.2 Test Vehicle Setup

The test vehicle differs from REMUS is several important ways. The test vehicle is built on a bare chassis, so lead ballast is positioned to simulate the inertial properties of the batteries and missing circuit boards. REMUS has an Acoustic Doppler Current Profiler (ADCP) section aft of the flooded nose section and end cap (see Figure 4.2). This section has four circular cutouts on the top and bottom for transducers, and is replaced on the test vehicle by a straight ballasted section of slightly longer length that also provided through-hull fitting for the wet-mounting serial data/power cable. The test vehicle also lacks fins, propeller, and the long-baseline navigation transducer under the nose.

Figure 4.2: REMUS ADCP section
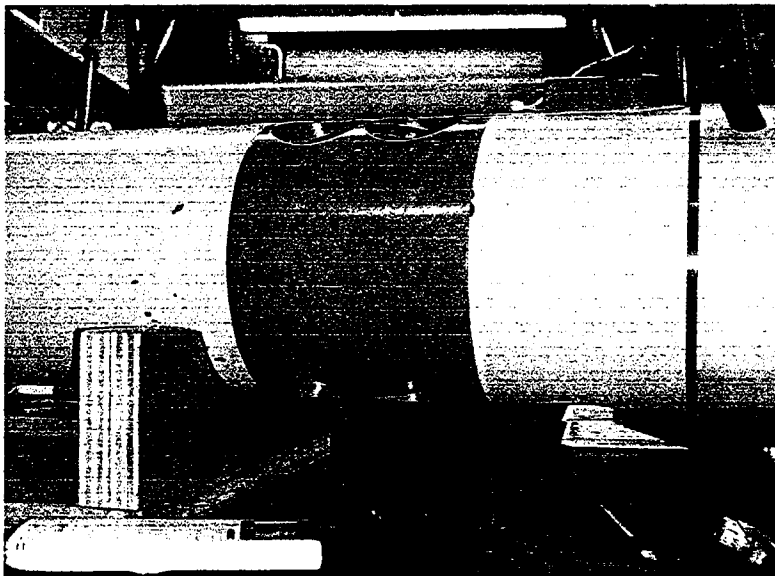
The test vehicle was assembled specifically for these tests and contains only two instruments. The first is an internal pressure sensor used to verify hull watertight integrity prior to drop tests. The second is a Crossbow DMU-AHRS (Attitude and Heading Reference System), located at the vehicle center of mass (see Figure 4.3). The Crossbow is powered externally through the serial data/power cable, and reports real-time attitude and acceleration data via an RS-232 serial link to a laptop computer equipped with data-logging software provided with the unit. The Crossbow sensor will be described in further detail in subsection 4.1.4.

Figure 4.3: Crossbow DMU-AHRS in ballasted test vehicle chassis
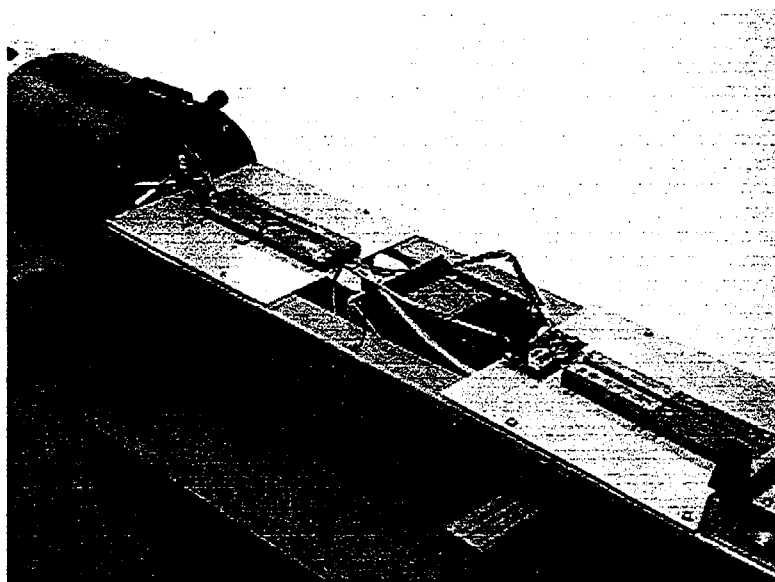
4.1.3 Test Vehicle Launcher

A vehicle launcher was built on the Eel Pond dock next to the Redfield Laboratory at WHOI (see Figure 4.4). The launcher consists of four ten-foot sections of PVC tubing supported by an external frame capable of varying the pitch angle. Three trap-doors are installed at various positions along the frame as starting gates. The test vehicle is loaded into the PVC track, positioned at the desired angle, and then dropped by releasing the trap door. The 50 foot cable trails the test vehicle through the launcher, allowing for real-time data acquisition and providing a tether for vehicle recovery.

Figure 4.4: Test vehicle launcher, with assistants Dave Stuebe and Alex Apotsos
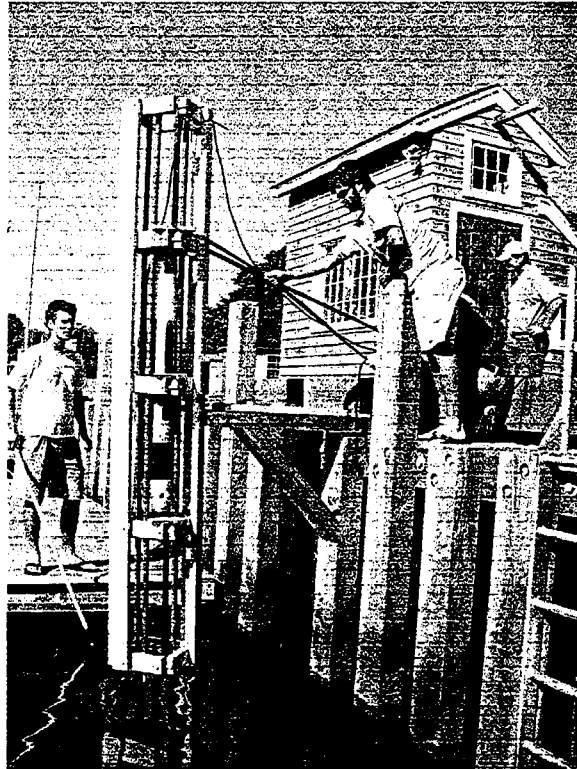
4.1.4 Crossbow DMU-AHRS

The Crossbow DMU-AHRS is a nine-axis measurement system that combines a three-axis accelerometer, a three-axis rotational rate sensor, and a three-axis magnetometer in order to measure stabilized roll, pitch, and yaw in a dynamic environment. The sensor was operated in angle mode, in which the sensor acts as a steady-state vertical gyroscope/artificial horizon with a directional gyroscope heading reference. The analog rotational rates and accelerations are sampled and converted to engineering units and other calculations are performed internally to perform the stabilizing functions to output roll, pitch, and yaw referenced to gravity and magnetic north. These data are output digitally at 80 Hertz.

The acceleration channels are the important data for this study, but these are referenced to a right-handed Cartesian coordinate system centered on the body center of mass, with x' positive forward (along longitudinal axis), y' positive to starboard, and z' positive downward (see Figure

4.5). Roll, pitch, and yaw angles are related respectively with the x', y', and z' axes in the sense that the fingers of the right hand will curl in the positive direction if the thumb is oriented along the associated axis. Angular data is necessary to convert between the body coordinate system and the inertial coordinate system.

Figure 4.5: Body centered coordinate system
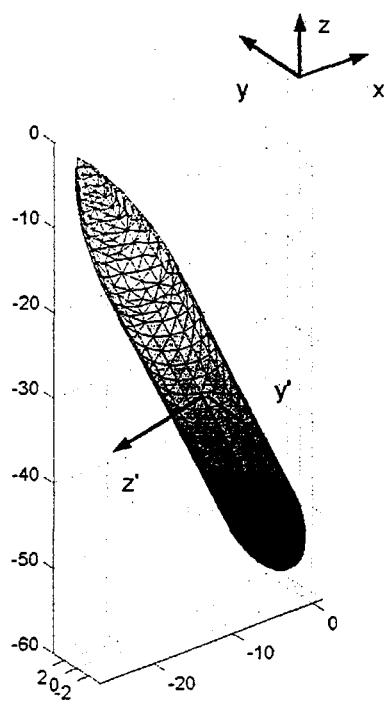
The angle output was generally good while the unit was stable, but not so during motion. There is noise in the rotational rate data that creates an error when the signal is integrated. This is similar to the slow drift error from precession of aircraft flight instrumentation, (this unit is an inertial guidance system suitable for such use). Internally, the unit corrects itself using the vectors of gravity and magnetic north to eliminate this error. However, one of the channels of the magnetometer was inoperable, affecting the stabilized angle output from the unit during motion. Angles can be calculated in post-processing by integrating the angular rate from the initial angles. There is non-zero-sum noise that integrates to erroneous angles when the unit is stationary, and this can be suppressed by artificially zeroing those channels during the times prior to the start of motion. Once the motion starts there is no way to quantify the error, so the

angles become suspect as time elapses, and therefore the coordinate transformation is also suspect.

The Crossbow unit was not an ideal instrument for reasons other than the lack of stabilized angles due to magnetometer malfunction. The unit is intended as a motion-sensing package, and thus the accelerometer range is $\pm 2g$, smaller than would ideally be available to conduct a range of tests. Also, the impulsive and step response transient behaviors are not known. The vehicle experiences a step function in acceleration as the trap door is released. At 80 Hertz it is possible to see an underdamped response in the acceleration data. It is desired to know the physical stimulus separate from the physical system used to measure it. Tests were conducted by allowing the unit to undergo a one-g step function as it is releasing into free-fall. A systems engineering approach can be taken for removing this response and the sensor can be represented as system, g(t), that transforms a known input, r(t) into a measured output, c(t):

$$r(t) \rightarrow \left[g(t) \right] \rightarrow c(t). \tag{4.1.1}$$

If the Laplace transform is used to transform from the time domain, t, into the frequency domain, s, the system is expressed

$$R(s) \rightarrow \left[G(s) \right] \rightarrow C(s), \tag{4.1.2}$$

where R(s), G(s), and C(s) are the transformed input, system, and output. The system G(s) is known as the transfer function,

$$G(s) = \frac{C(s)}{R(s)}. \tag{4.1.3}$$

For a general underdamped system, the output can be expressed

$$C(s) = \frac{\omega_n^2}{s\left(s^2 + 2\zeta\omega_n + \omega_n^2\right)}, \tag{4.1.4}$$

where ω_n is the natural frequency and ς is the damping coefficient. These parameters can be quantified by measuring the rise time and overshoot of the response. The step response can be written in terms of ω_n and ς:

$$c(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \cos\left(\omega_n \sqrt{1-\zeta^2}\, t - \tan^{-1}\left(\frac{\zeta}{\sqrt{1-\zeta^2}} \right) \right). \tag{4.1.5}$$

To remove the step response from the data, a vector is created at the same sampling rate (80 Hz) to represent the pure step response. This vector is deconvolved from the response to recover the step function (see Figure 4.6). This is not a perfect step function, but it is an improvement over the original response in that the overshoot is removed and the steady state value is reached more quickly.

Figure 4.6: Crossbow response to -1 g step, before and after treatment
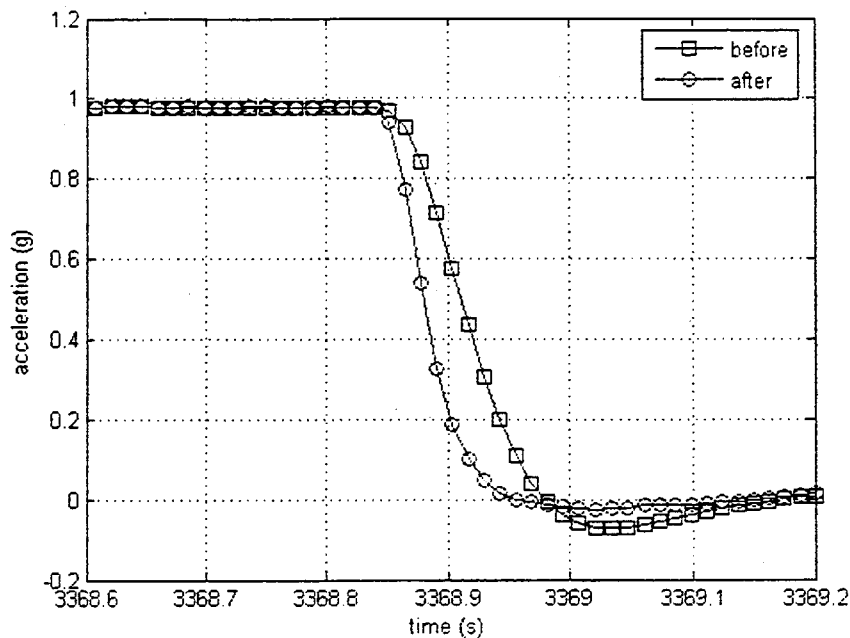
The transfer function represents the transformation of the physical stimulus into the digital data output. When the effect of the transfer function is removed, the data better conveys the physical quantities being measured. In this case, the three axes of the accelerometer are identical in response, so this transformation is performed on all three channels before any further manipulation is done.

4.2 Test Vehicle Experimental Results

Test drops were conducted at pitch angles of -90°, -75°, -60°, -45°, and -30° using the test apparatus described in subsection 4.1.3. The following steps were used to convert the raw data

into usable form. Data were logged as ASCII text files and read into MATLAB. When not in motion, the Crossbow reads the vector components of gravitational acceleration. After the transfer function is removed, the three channels are normalized to gravity for the portion of the data record prior to motion. The noise in the rotation rates is also zeroed before motion begins to eliminate an initial drift error from the initial conditions for the integration of the rates into roll, pitch, and yaw angles. Figure 4.7 shows the data for -90° after these steps have been performed. Note that these data are in body coordinates, such that an x' acceleration of $1g$ occurs when the body is stationary in the launcher with the longitudinal axis straight down. When the trap door is released, the body free falls, then the impact is from $0g$ to about $1.3g$. These data do not account for body rotations, so additional steps are required.

Figure 4.7: Test vehicle 3-axis accelerations in body coordinates
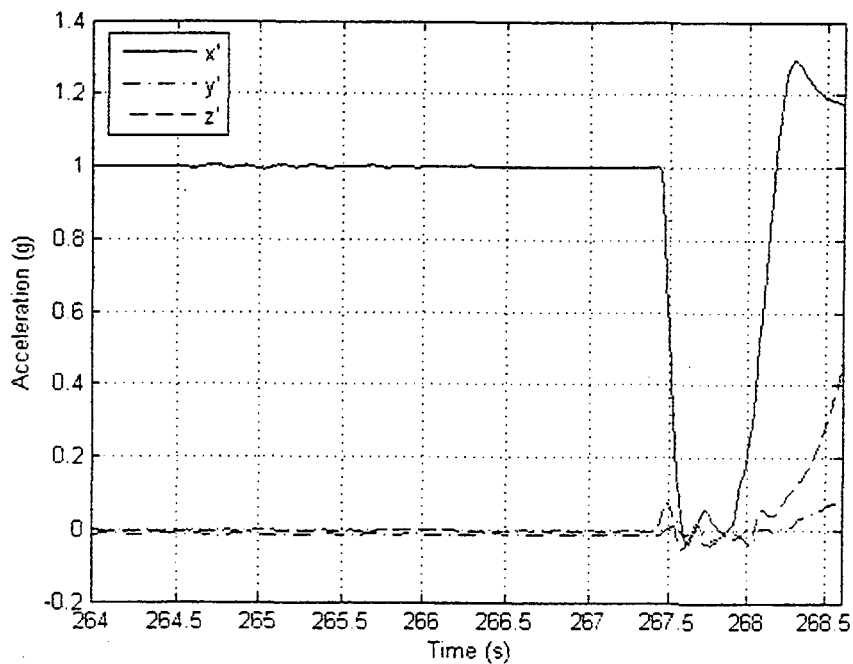
Body angles are calculated, and then the components of gravity are removed from the data at each time step. Finally, a rotational transformation is performed to convert the acceleration components into the inertial coordinate system (see Figure 4.8). This result is for a drop from a height of approximately two meters, referenced as previously to the body center of mass. It is comparable to the numerical results discussed in Chapter 3, and reproduced in Figure

4.9. The experimental result is an initial rise from free-fall to positive 2.99 m/s², as compared to the numerical result, an initial rise from free-fall to 4.74 m/s².

Figure 4.8: Test vehicle 3-axis accelerations in inertial coordinates at -90°
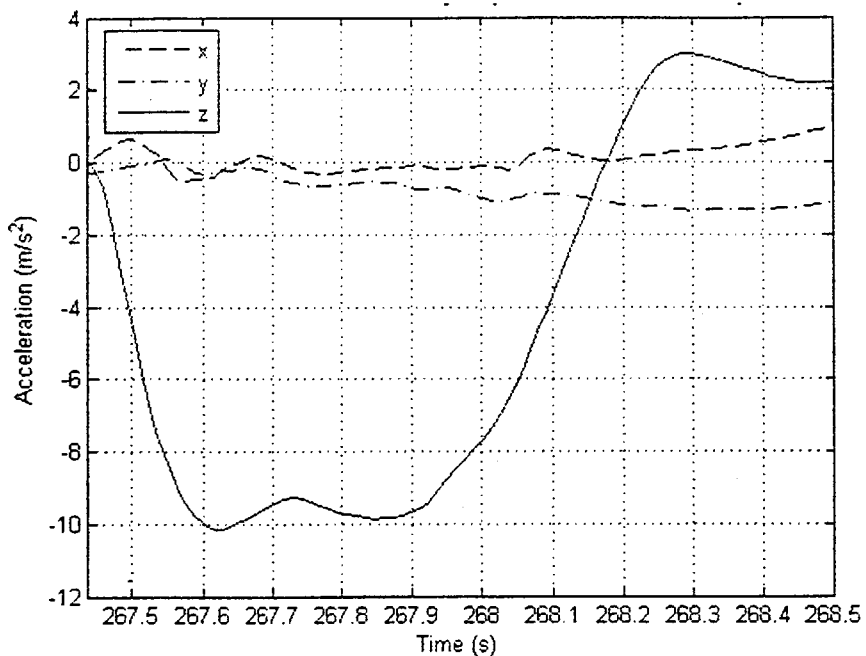
Figure 4.9: Numerical Heave acceleration at -90°
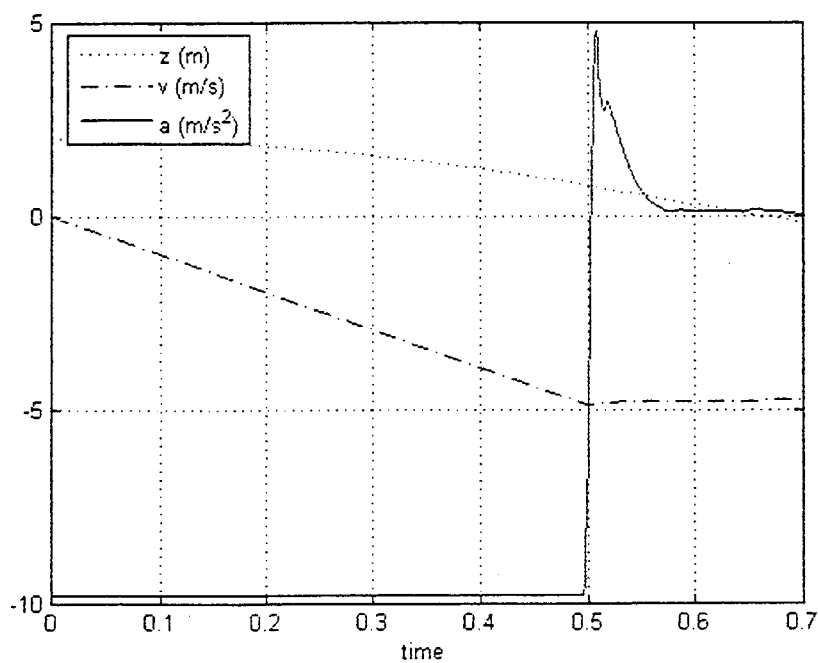
Figures 4.10 to 4.13 show examples of experimental results for the other angles. All of these results are for a drop from the lowest trapdoor, placing the test vehicle center of mass at approximately two meters regardless of how the frame is rotated (see Figure 4.4).

Figure 4.10: Test vehicle 3-axis accelerations in inertial coordinates at -75°
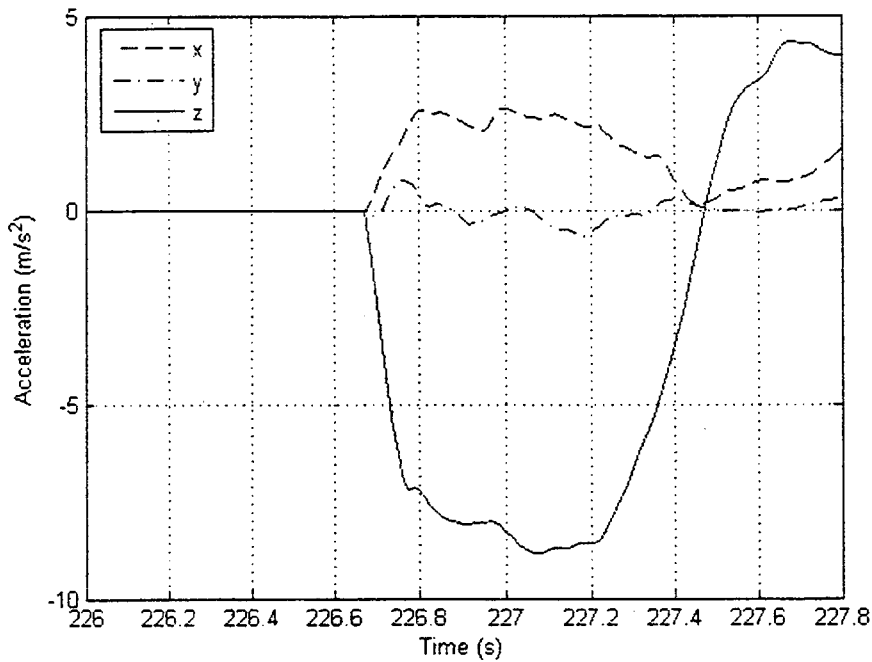
Figure 4.11: Test vehicle 3-axis accelerations in inertial coordinates at -60°
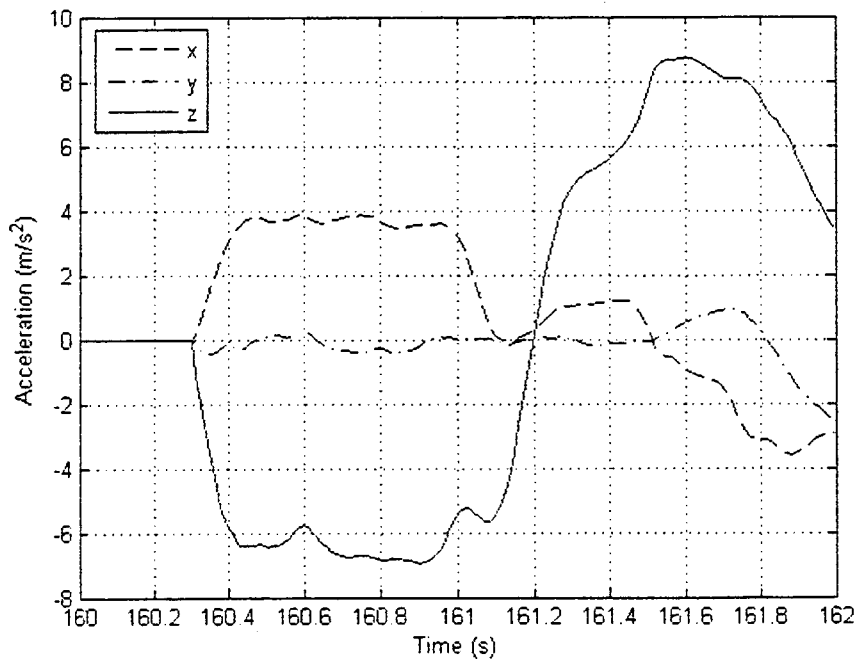
Figure 4.12: Test vehicle 3-axis accelerations in inertial coordinates at -45°
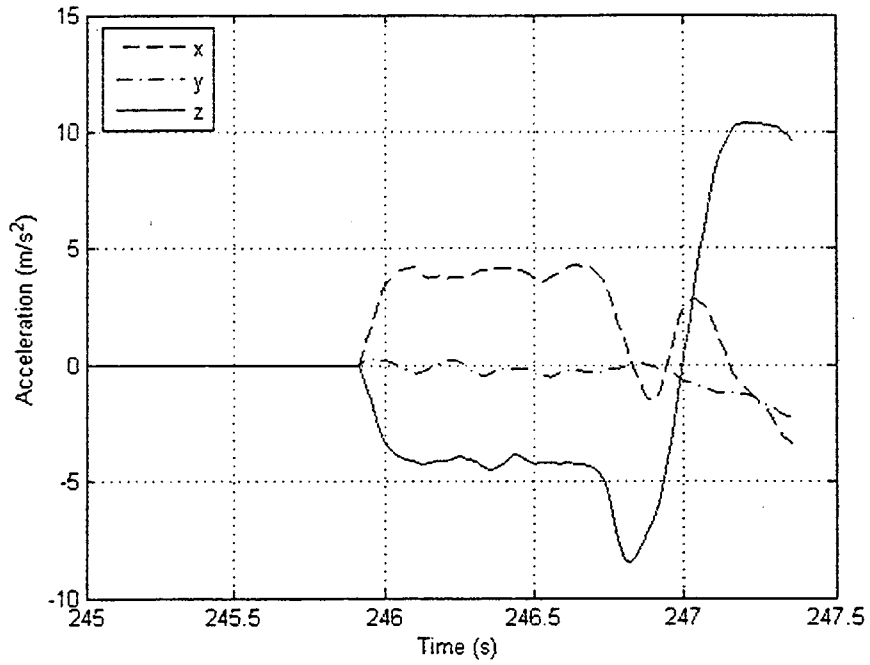
Figure 4.13: Test vehicle 3-axis accelerations in inertial coordinates at -30°
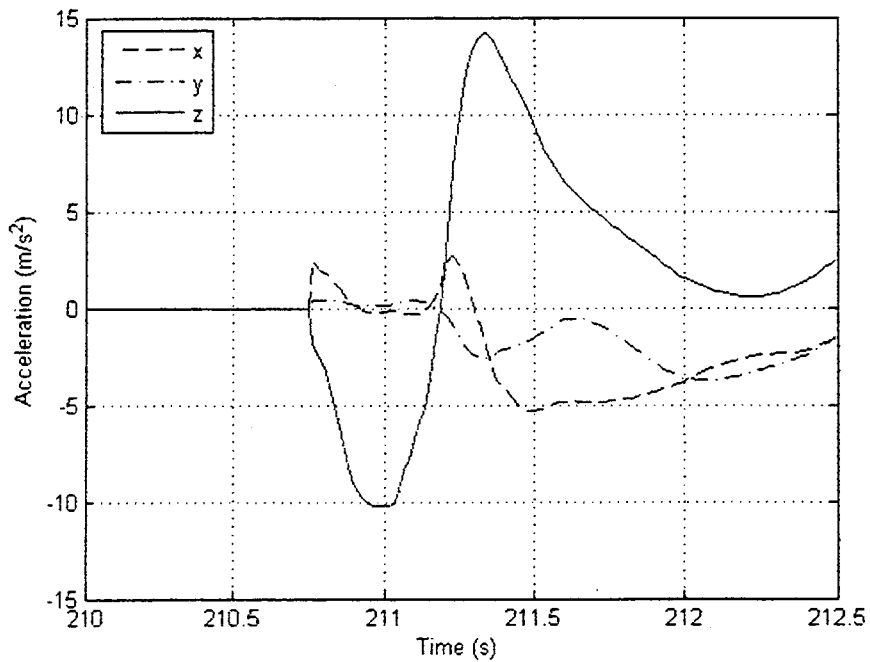
The design of the launcher itself changes the way in which the test vehicles enter the water. The angles listed are the starting pitch of the test vehicle in the launcher. Friction

opposes the acceleration in the x'direction as the vehicle slides down the rails, and a component of the gravitational acceleration accelerates the body in the x direction for angles other than vertical. Therefore the vertical velocity is smaller than for free-fall from the same height and there are acceleration components other than vertical. Also, the closer the angle is to horizontal, the more the pitch changed between leaving the launcher and impact, becoming more vertical as the nose dropped out of the rails while the tail was still supported. This also introduces rotation, which was not modeled numerically.

Experimental results were limited to a total of nine data points for five angles (see Figure 4.14). The three results at -30° show good repeatability, though they are farthest from the numerically predicted value. This may be related to either of the launcher problems discussed above, both of which would lead to smaller accelerations. Additional experiments were conducted, but data was either corrupted by noise or did not contain useful data due to the instrumentation performing a reset at the moment of impact.

Figure 4.14: Comparison between numerical and experimental results
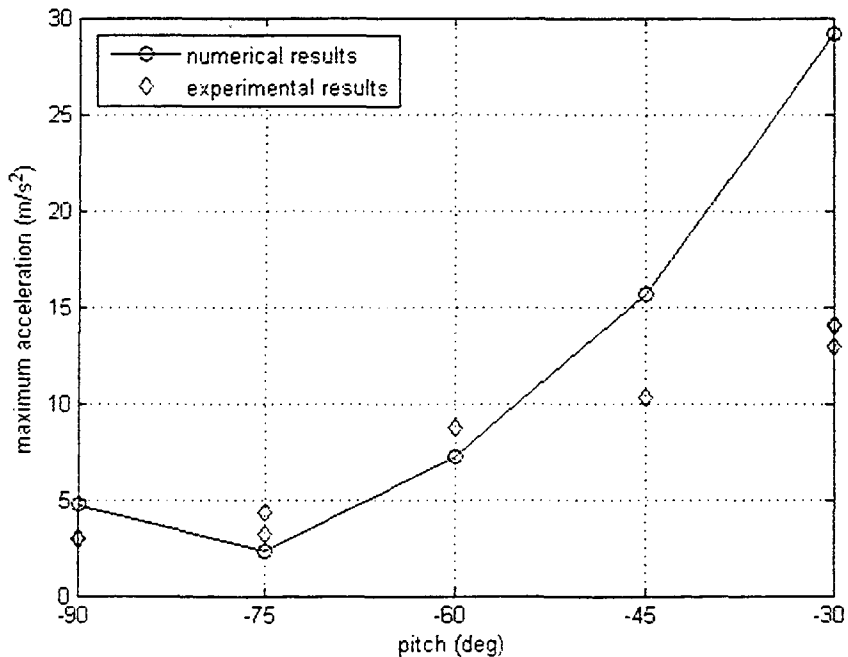
Chapter 5
Conclusions

5.1 Summary

The goal of this thesis was to numerically model the REMUS AUV to determine maximum impact accelerations as related to the suitability of helicopter air-drop deployments. A technique of numerical modeling was developed which accounts for the first-order effects of momentum transfer from a rigid body to the water using the generalized von Karman method, which makes use of the concept of added-mass. Added-mass is a property based solely on body geometry that quantifies the additional inertial effects experienced by a body undergoing acceleration while immersed in a fluid that is assumed to be ideal and irrotational. Added-mass is calculated using a source collocation panel method with panel meshes generated by FEMLAB using MATLAB scripting. A continuous smooth curve is generated for added-mass and its spatial derivative by fitting polynomials to the calculated data points. These polynomials are used to integrate the second-order equation of motion to determine impact accelerations. The method of added-mass calculations was verified against analytical solutions. Experiments were conducted to verify the maximum impact acceleration calculations, and good agreement was found for the nearly vertical values of pitch at which the experiment and the model accurately represent the same physical processes. Other data points do not agree as well, but are within a factor of two.

5.2 Discussion

The model developed in the present work effectively predicts the maximum impact accelerations for REMUS in a vertical free-fall at constant pitch. Calculations were performed for a range of pitch values and a range of starting heights. The relationship between pitch and maximum heave acceleration is nearly the intuitive result. Minimum accelerations occur when the REMUS enters the water in a streamlined fashion similar to a diver presenting minimal surface area. It is not only the straight vertical entry that benefits, but a range of about 30° from vertical where the accelerations are fairly low. This is potentially good news for manually dropping a REMUS from a helicopter, since the vehicle can be dropped vertically without fear that slight pitch changes in midair will result in much greater forces. The angles near horizontal are much different, with the greatest being an order of magnitude larger than the smallest accelerations near vertical. This case is intuitively similar to the "belly-flop."

The relationship between height and maximum heave acceleration is even clearer. This relationship is clearly linear for a given pitch angle. Given these data it is possible to extrapolate an approximate value of the maximum acceleration for starting heights higher than those presented. The heights shown here were chosen to be reasonable altitudes at which a helicopter might hover to drop a REMUS. For these values, the accelerations for the range of pitch near vertical seem reasonable, although no data are presented about possible failure modes or the range of safe acceleration values.

5.3 Future Work

A great deal more work is necessary to determine if REMUS may be routinely and safely deployed from a hovering helicopter. The chief concern is the questions of the failure modes and how much force is required to cause any of these failures. This deserves more work, because the acceleration data presented in this thesis have no meaning unless referenced to a failure mode. I have considered several possibilities, falling naturally into the categories of catastrophic failures or mission failures. A catastrophic failure may entail the loss of a vehicle, such as the loss of watertight integrity. Mission failure may be the breaking of fins, or the failure of internal moving parts like hard drives.

Further work would also be valuable to verify this model with more experimental results. I desired to do more tests, because I was not able to do experiments at a range of heights. A better sensor is recommended due to the problems with the Crossbow motion package, which was clearly out of its element. Its maximum range of $\pm 2g$ would prohibit much higher tests at nearly vertical angles, as well as nearly any tests to verify the numerical results of the nearly flat drops. A better experimental setup could be used for the flatter drops, which would better simulate vertical free-fall. The test vehicle could be suspended at the chosen angle using rope and hose clamps around the body. Free-fall and easy recovery could be accomplished by releasing the rope over a pulley.

Appendix A

MATLAB Code examples

A.1 MATLAB script to setup mesh generation variables for REMUS model

The following script m-file contains the outer loop structure which varies the pitch in five degree increments from minus ninety to plus ninety degrees and the inner loop structure which varies the submergence value for each angle in one-hundred increments from zero to full submergence; these variables are stored in the file pitchsubtable.mat. For each set of values, a FEMLAB meshing script is called like the example listed in Appendix A.2. When called from MATLAB, the mesh is generated and exported to the workspace. The script m-file listed in Appendix A.3 then operates on the mesh as described in Chapter 2.

```
% MeshProcess.m
% Stephen M. Roe
% 7/8/2004

clear; close all; clc

load pitchsubtables

for angle = 1:37;
   offset = Offset(angle);
   for subcounter = 1:100
      sub = Sub(subcounter,angle)
      pitch = Pitch(angle)
      if sub ~= 0;
         if pitch == -90;
            if sub <= 9; m90deg1t9;
            elseif sub > 9 & sub <= 63; m90deg10t63;
            else; m90deg64t99;
```

```
        end
    elseif pitch == -85;
        if sub <= 9; m85deg1t9;
        elseif sub > 9 & sub <= 63; m85deg10t63;
        else; m85deg64t99;
        end
    elseif pitch == -80;
        if sub <= 9; m80deg1t9;
        elseif sub > 9 & sub <= 63; m80deg10t63;
        else; m80deg64t99;
        end
    elseif pitch == -75;
        if sub <= 9; m75deg1t9;
        elseif sub > 9 & sub <= 62; m75deg10t62;
        else; m75deg63t99;
        end
    elseif pitch == -70;
        if sub <= 9; m70deg1t9;
        elseif sub > 9 & sub <= 60; m70deg10t60;
        else; m70deg61t99;
        end
    elseif pitch == -65;
        if sub <= 9; m65deg1t9;
        elseif sub > 9 & sub <= 58; m65deg10t58;
        else; m65deg59t99;
        end
    elseif pitch == -60;
        if sub <= 9; m60deg1t9;
        elseif sub > 9 & sub <= 56; m60deg10t56;
        else; m60deg57t99;
        end
    elseif pitch == -55;
        if sub <= 9; m55deg1t9;
        elseif sub > 9 & sub <= 53; m55deg10t53;
        else; m55deg54t99;
        end
    elseif pitch == -50;
        if sub <= 9; m50deg1t9;
        elseif sub > 9 & sub <= 49; m50deg10t49;
        else; m50deg50t99;
        end
    elseif pitch == -45;
        if sub <= 9; m45deg1t9;
        elseif sub > 9 & sub <= 46; m45deg10t46;
        else; m45deg47t99;
        end
    elseif pitch == -40;
        if sub <= 9; m40deg1t9;
        elseif sub > 9 & sub <= 42; m40deg10t42;
        else; m40deg43t99;
        end
```

```
        elseif pitch == -35;
            if sub <= 9; m35deg2t9;
            elseif sub > 9 & sub <= 38; m35deg10t38;
            else; m35deg39t99;
            end
        elseif pitch == -30;
            if sub <= 7; m30deg2t7;
            elseif sub > 7 & sub <= 33; m30deg8t33;
            else; m30deg34t99;
            end
        elseif pitch == -25;
            if sub <= 6; m25deg2t6;
            elseif sub > 7 & sub <= 28; m25deg8t28;
            else; m25deg29t99;
            end
        elseif pitch == -20;
            if sub <= 5; m20deg2t5;
            elseif sub > 5 & sub <= 24; m20deg6t24;
            else; m20deg25t99;
            end
        elseif pitch == -15;
            if sub <= 19; m15deg1t19;
            else; m15deg20t99;
            end
        elseif pitch == -10;
            if sub <= 15; m10deg05t15;
            else; m10deg16t99;
            end
        elseif pitch == -5;
            if sub <= 12; m05deg05t12;
            else; m05deg13t99;
            end
        elseif pitch == 0;
            if sub <= 8; m00deg05t8;
            else; m00deg9t99;
            end
        MeshProcess2
    end
  end
end
```

A.2 FEMLAB mesh generation script file for REMUS vehicle

```
% FEMLAB Model M-file
% Generated by FEMLAB 3.0 (FEMLAB 3.0.0.181, $Date: 2004/01/29 19:04:14 $)

flclear xfem

% Femlab version
clear vrsn
vrsn.name = 'FEMLAB 3.0';
vrsn.ext = '';
vrsn.major = 0;
vrsn.build = 181;
vrsn.rcs = '$Name:  $';
vrsn.date = '$Date: 2004/01/29 19:04:14 $';
xfem.version = vrsn;

% Geometry 2
g5=ellip2(1.5,1.5,'base','center','pos',[1.5,2.25]);
g6=rect2(5.5,2.25,'base','corner','pos',[0,0]);
g7=rect2(4,1.5,'base','corner','pos',[1.5,2.25]);
g8=geomcomp({g5,g6,g7},'ns',{'E1','R1','R2'},'sf','E1+R1+R2','edge','all');
g9=rect2(33.564,3.75,'base','corner','pos',[5.5,0]);
carr={curve2([39.064,39.736],[3.75,3.738],[1,1]), ...
  curve2([39.736,40.516],[3.738,3.7139],[1,1]), ...
  curve2([40.516,41.416],[3.7139,3.6779],[1,1]), ...
  curve2([41.416,42.412],[3.6779,3.6298],[1,1]), ...
  curve2([42.412,42.412],[3.6298,0],[1,1]), ...
  curve2([42.412,39.064],[0,0],[1,1]), ...
  curve2([39.064,39.064],[0,3.75],[1,1])};
g10=geomcoerce('solid',carr);
carr={curve2([42.412,43.504],[3.6298,3.5577],[1,1]), ...
  curve2([43.504,44.692],[3.5577,3.4615],[1,1]), ...
  curve2([44.692,45.952],[3.4615,3.3293],[1,1]), ...
  curve2([45.952,47.284],[3.3293,3.149],[1,1]), ...
  curve2([47.284,48.688],[3.149,2.9207],[1,1]), ...
  curve2([48.688,48.688],[2.9207,0],[1,1]), ...
  curve2([48.688,42.412],[0,0],[1,1]), ...
  curve2([42.412,42.412],[0,3.6298],[1,1])};
g11=geomcoerce('solid',carr);
carr={curve2([48.688,50.152],[2.9207,2.6202],[1,1]), ...
  curve2([50.152,51.664],[2.6202,2.2716],[1,1]), ...
  curve2([51.664,53.212],[2.2716,1.8269],[1,1]), ...
  curve2([53.212,54.796],[1.8269,1.3101],[1,1]), ...
  curve2([54.796,56.392],[1.3101,0.7091],[1,1]), ...
  curve2([56.392,57.004],[0.7091,0.4327],[1,1]), ...
  curve2([57.004,57.172],[0.4327,0.4087],[1,1]), ...
  curve2([57.172,58],[0.4087,0.4087],[1,1]), ...
```

```
     curve2([58,58],[0.4087,0],[1,1]), ...
     curve2([58,48.688],[0,0],[1,1]), ...
     curve2([48.688,48.688],[0,2.9207],[1,1])};
g12=geomcoerce('solid',carr);
carr={curve2([39.064,44.692,51.664,58],[3.75,3.7139,2.9207,0],[1,1,1,1]), ...
     curve2([58,51.664,47.284,39.064],[0,0,0,0],[1,1,1,1]), ...
     curve2([39.064,39.064,39.064,39.064],[0,1.3101,2.6202,3.75],[1,1,1,1])};
g13=geomcoerce('solid',carr);
clear g10
clear g11
clear g12
clear g9
g16=rect2(37.564,3.75,'base','corner','pos',[1.5,0]);
clear g8
clear g16
g19=rect2(37.564,3.7500000000000004,'base','corner','pos',[1.5,0]);
g20=rect2(1.5,2.25,'base','corner','pos',[0,0]);
g21=ellip2(1.5,1.5,'base','center','pos',[1.5,2.25]);
g22=geomcomp({g13,g19,g20,g21},'ns',{'CO5','R1','R2','E1'},'sf','CO5+R1+R2+E1','edge','all');
flclear fem
clear s
s.objs={g22};
s.name={'CO1'};
s.tags={'g22'};

fem.draw=struct('s',s);
xfem.fem{2}=fem;

flclear fem
fem.sdim = {'x','y','z'};
xfem.fem{1} = fem;

fem=xfem.fem{2};
fem.sdim = {'x','y'};
xfem.fem{2} = fem;

% Multiphysics
xfem=multiphysics(xfem);
% FEMLAB Model M-file
% Generated by FEMLAB 3.0 (FEMLAB 3.0.0.181, $Date: 2004/01/29 19:04:14 $)

% Geometry 2
g2=rect2(56,3.75,'base','corner','pos',[4,0]);
g3=geomcomp({g22,g2},'ns',{'CO1','R1'},'sf','CO1*R1','edge','all');
g4=ellip2(9,3.75,'base','center','pos',[4,0]);
g5=rect2(9,3.75,'base','corner','pos',[-5,0]);
g6=geomcomp({g4,g5},'ns',{'E1','R1'},'sf','E1*R1','edge','all');
g7=geomcomp({g3,g6},'ns',{'CO2','CO1'},'sf','CO2+CO1','edge','all');
g7=move(g7,[5,0]);
fem=xfem.fem{2};
clear s
```

```
s.objs={g7};
s.name={'CO3'};
s.tags={'g7'};

fem.draw=struct('s',s);
xfem.fem{2}=fem;
% FEMLAB Model M-file
% Generated by FEMLAB 3.0 (FEMLAB 3.0.0.181, $Date: 2004/01/29 19:04:14 $)

% Geometry 2
g1=revolve(g7,'Angles',[0,6.283185307179586],'Revaxis',[0 1;0 0],'Wrkpln',[0 0 1;0 0 0;0 1 0]);

% Geometry 1
fem=xfem.fem{1};
clear s
s.objs={g1};
s.name={'REV1'};
s.tags={'g1'};

fem.draw=struct('s',s);
xfem.fem{1}=fem;

% Geometry 2
fem=xfem.fem{2};
clear s
s.objs={g7};
s.name={'CO3'};
s.tags={'g7'};

fem.draw=struct('s',s);
xfem.fem{2}=fem;


% FEMLAB Model M-file
% Generated by FEMLAB 3.0 (FEMLAB 3.0.0.181, $Date: 2004/01/29 19:04:14 $)

% Geometry 1
g1=rotate(g1,0.7853981633974483,[0,-1,0],[0,0,0]);
g1=move(g1,[0,0,offset]);
g2=block3('50','10','50','base','center','pos',{'-23','0','-25'},'axis',{'0','0','1'},'rot','0');
g1=move(g1,[0,0,-sub]);
g3=geomcomp({g1,g2},'ns',{'REV1','BLK1'},'sf','REV1*BLK1','face','none','edge','all');
fem=xfem.fem{1};
clear s
s.objs={g3};
s.name={'CO1'};
s.tags={'g3'};

fem.draw=struct('s',s);
fem.geom=geomcsg(fem);
```

```matlab
% Initialize mesh for geometry 1
fem.mesh=meshinit(fem,'hmaxfact',1.5,'hcutoff',0.02, ...
          'hgrad',1.5,'hcurve',0.5);
xfem.fem{1}=fem;

% Geometry 2
fem=xfem.fem{2};
clear s
s.objs={g7};
s.name={'CO3'};
s.tags={'g7'};

fem.draw=struct('s',s);
xfem.fem{2}=fem;

fem=xfem.fem{2};
fem.sdim = {'x','y'};
xfem.fem{2} = fem;

% Multiphysics
xfem=multiphysics(xfem);
```

A.3 MATLAB script to manipulate the panel mesh and evaluate added-mass

```
% MeshProcess2.m
% Stephen M. Roe
% 9/09/2004

close all

fem = xfem.fem{1};

mesh = [getfield(fem,'mesh')]; % mesh
nodes = [getfield(mesh,'p')]'; % nodes
bdry = [getfield(mesh,'e')]'; % describes which nodes define panels
panels = bdry(:,1:3); % only first three columns needed
clear bdry

for a = 1:3;
   b = find(abs(nodes(:,a)) <= 1e-6);
   nodes(b,a) = 0;
end

npanels = length(panels);

% flatten any error panels that are above the free surface (z = 0)
for r = 1:length(nodes);
   if nodes(r,3) > 0; nodes(r,3) = 0; end
end

figure; clf
trisurf(panels,nodes(:,1),nodes(:,2),nodes(:,3))
axis equal
h = gcf;
set(h,'Renderer','zbuffer','RendererMode','manual')
colormap(gray)
xlabel('X')
ylabel('Y')
zlabel('Z')

% prior to rotation, shift nose to origin
nodes(:,3) = nodes(:,3) + sub - offset;

% rotate coordinate frame of nodes to Remus nose down
Rpitch = [cos(-(90+pitch)*pi/180) 0 -sin(-(90+pitch)*pi/180);0 1 0; sin(-(90+pitch)*pi/180) 0 cos(-
(90+pitch)*pi/180)];
for e = 1:length(nodes);
   nodes(e,:) = [Rpitch*nodes(e,:)']';
```

```
end

% get rid of panels with y>0, so they can be replaced by a mirror image
% across y=0 plane for better symmetry
for f = 1:npanels;
    ysign(f,1) = sign(mean([nodes(panels(f,1),2) nodes(panels(f,2),2) nodes(panels(f,3),2)]));
    % adjust panel order to make normals correct direction
    if nodes(panels(f,1)) > 9 & nodes(panels(f,1)) < 44; panels(f,:) = fliplr(panels(f,:));end
end
ypositive = find(ysign(:,1) == 1);
panels = panels(ypositive,:);
npanels = length(panels);
% y positive half body
Body = zeros(4*npanels + 1,3); % 'body' is in format of addedmass.dat input to addedmass.exe
m = 1;
for n = 2:4:4*npanels;
    Body(n,:) = nodes(panels(m,1),:);
    Body(n+1,:) = nodes(panels(m,2),:);
    Body(n+2,:) = nodes(panels(m,3),:);
    Body(n+3,:) = nodes(panels(m,1),:);
    m = m + 1;
end
% mirror image about y = 0 to give whole body
lbody = length(Body);
Body(lbody+1:2*lbody-1,1) = flipud(Body(2:lbody,1));
Body(lbody+1:2*lbody-1,2) = -flipud(Body(2:lbody,2));
Body(lbody+1:2*lbody-1,3) = flipud(Body(2:lbody,3));
npanels = 2*npanels;

% rotate back to original coordinate system
for g = 2:length(Body);
    Body(g,:) = [Rpitch'*Body(g,:)']';
end
CG = [Rpitch'*[-0.75 0 30]']'; % (in) Center of Gravity position

% calculate the center of buoyancy (center of displaced volume)
panelcenter = zeros(npanels,3);
p = 1;
for q = 1:npanels;
    a1 = p + 1;
    b1 = a1 + 1;
    c1 = a1 + 2;
    p = 4*q + 1;
    panelcenter(q,1) = mean([Body(a1,1) Body(b1,1) Body(c1,1)]);
    panelcenter(q,2) = mean([Body(a1,2) Body(b1,2) Body(c1,2)]);
    panelcenter(q,3) = mean([Body(a1,3) Body(b1,3) Body(c1,3)]);
    ABC(1,1) = sqrt((Body(b1,1)-Body(c1,1))^2 + (Body(b1,2)-Body(c1,2))^2 + (Body(b1,3)-
Body(c1,3))^2);
    ABC(1,2) = sqrt((Body(a1,1)-Body(c1,1))^2 + (Body(a1,2)-Body(c1,2))^2 + (Body(a1,3)-
Body(c1,3))^2);
```

```matlab
    ABC(1,3) = sqrt((Body(a1,1)-Body(b1,1))^2 + (Body(a1,2)-Body(b1,2))^2 + (Body(a1,3)-
Body(b1,3))^2);
    AngleAB = acos((ABC(1,1)^2 + ABC(1,2)^2 - ABC(1,3)^2)/2/ABC(1,1)/ABC(1,2));
    panelarea(q,1) = ABC(1,1)*ABC(1,2)*sin(AngleAB)/2;
end
rCBNose(1,1) = mean(panelarea(:,1).*panelcenter(:,1))/mean(panelarea);
rCBNose(1,2) = mean(panelarea(:,1).*panelcenter(:,2))/mean(panelarea);
rCBNose(1,3) = mean(panelarea(:,1).*panelcenter(:,3))/mean(panelarea);

% move back down to free surface
Body(:,3) = Body(:,3) - sub + offset;
Nose = [0 0 -sub + offset];
CB = rCBNose;
CB(1,3) = CB(1,3) - sub + offset;
CG(1,3) = CG(1,3) -sub + offset;


% flatten any free surface panels (again)
a = find(abs(Body(:,3)) < 0.001);
Body(a,3) = 0;

% remove the z = 0 (free surface) panels prior to copying for the image
c = 1;
for b = 2:4:length(Body);
    if Body(b,3) == 0 & Body(b+1,3) == 0 & Body(b+2,3) == 0 & Body(b+3,3) == 0
        d(c,1) = b;
        d(c+1,1) = b+1;
        d(c+2,1) = b+2;
        d(c+3,1) = b+3;
        c = c + 4;
    end
end
if c ~= 1; Body(d,:) = []; end

% free-surface mirror image
Ibody = length(Body);
Body(Ibody + 1:2*Ibody-1,1:2) = (Body(2:Ibody,1:2));
Body(Ibody + 1:2*Ibody-1,3) = -(Body(2:Ibody,3));
npanels = (length(Body)-1)/4;

% move body to CG
Body(:,1) = Body(:,1) - CG(1,1);
Body(:,2) = Body(:,2) - CG(1,2);
Body(:,3) = Body(:,3) - CG(1,3);
CB(:,1) = CB(:,1) - CG(1,1);
CB(:,2) = CB(:,2) - CG(1,2);
CB(:,3) = CB(:,3) - CG(1,3);
Nose(:,1) = Nose(:,1) - CG(1,1);
Nose(:,2) = Nose(:,2) - CG(1,2);
Nose(:,3) = Nose(:,3) - CG(1,3);
```

```
Center = [CB(1,1)/2 CB(1,2)/2 -CG(1,3)];

Body(1,1:4) = [npanels Center];

figure; clf
plot3(0,0,0,'k.'); hold on
plot3(CB(1,1),CB(1,2),CB(1,3),'.')
plot3(Center(1,1),Center(1,2),Center(1,3),'g.')
n = 2;
for o = 1:(length(Body)-1)/4;
    plot3(Body(n:n+3,1),Body(n:n+3,2),Body(n:n+3,3),'r'), hold on
    n = n + 4;
end
axis equal
title(sprintf('mesh plot at submergence = %1.4g" and angle = %1.4g deg',sub,pitch))
xlabel('X')
ylabel('Y')
zlabel('Z')
legend('Origin: Mass Center','Buoyancy Center','Symmetry Center')
grid on

save addedmass.dat Body -ascii -double
dos addedmass4.exe;
fid = fopen('addedmass.out');
AddedMass(:,:) = [fscanf(fid,'%g %g %g %g %g %g',[6 6])]';
Volume = fscanf(fid,'%g',[1]);
VolumeCenter = fscanf(fid,'%g',[1 3]);
status = fclose(fid);

if pitch <= 0; savefile = sprintf('m%gm%g.mat',-pitch,sub);
else; savefile = sprintf('p%gm%g.mat',pitch,sub);
end
save(savefile,'AddedMass','Body','CB','Nose','Volume','offset','pitch','sub')

clear ABC AddedMass AngleAB Body CB CG Center Nose Rpitch Volume VolumeCenter
clear a a1 ans b b1 c c1 carr d e f fem fid g g1 g13 g19 g2 g20 g21 g22
clear g3 g4 g5 g6 g7 h lbody m mesh n nodes npanels o p panelarea panelcenter

clear panels q r rCBNose s status vrsn xfem ypositive ysign
```

A.4 MATLAB function defining heave equation of motion

The following function evaluates the equation of motion for one of the MATLAB ordinary differential equation solvers, such as ODE45 or ODE113. Polynomial descriptions of added-mass and its spatial derivative are saved in separate –mat files for each pitch angle and called by a function like this one. The matrix *y* contains values of vertical displacement and velocity. The *if, elseif,* and *else* statements define the second order system conditionally based on the vertical displacement relative to the free surface.

```
function dydt = remusimpact90(t,y)

% function to evaluate the water impact of REMUS at -90 deg

load m90deg

m = 81.2*0.4535924;         % (lb -> kg)

if y(1) > z0
    dydt = [y(2);
        -g];
elseif y(1) <= z0 & y(1) >= 0
    a33 = polyval(pa,y(1));     % (kg)
    da33dz = polyval(k,y(1));   % (kg/m)
    dydt = [y(2);
        ( - da33dz*y(2).^2)/(m + a33)];
else
    dydt = [y(2);
        0];
end
```

Bibliography

[1]   Boef, W.J.C. 1992. Launch and impact of free-fall lifeboats. Part I. Impact Theory. *Ocean Engng* 19, 119-138.

[2]   Boef, W.J.C. 1992. Launch and impact of free-fall lifeboats. Part II. Implementation and applications. *Ocean Engng* 19, 139-159.

[3]   Jones, D. Operation Iraqi Freedom: Mine Countermeasures a Success. *Underwater*. July/August 2003. 13-16.

[4]   May, A. 1970. Review of water-entry theory and data. *J. Hydronautics* 4.

[5]   Milgram, Jerome H. 2003. Class notes. 13.024 *Numerical Marine Hydrodynamics*. MIT.

[6]   Miloh, T. 1981. Wave slam on a sphere penetrating a free surface. *J. Eng. Math.* 15: 221-240.

[7]   Miloh, T. 1991. On the initial-stage slamming of a rigid sphere in a vertical water entry. Appl. Ocean Research 13(1): 43-48.

[8]   Miloh, T. 1991. On the oblique water-entry of a rigid sphere. *J. Eng. Math* 15.

[9]   Myring, D.F. 1976. A Theoretical Study of Body Drag in Subcritical Axisymmetric Flow. Aeronautical Quarterly, 27(3): 186-94.

[10]   Newman, J.N. 1977. *Marine Hydrodynamics*. MIT Press, Cambridge, Massachusetts.

[11]   Payne, P.R. 1981. The vertical impact of a wedge on a fluid. *Ocean Engng* 8.

[12]   Szebehely, V.G. 1959. Hydrodynamic impact. *Appl. Mech. Rev.* 12.

[13]   Von Karman, Th. 1929. The impact on seaplane floats during landing. NACA TN 321.

[14]   Wagner, H. 1932. The phenomenon of impact and planning on water. NACA Translation 1366, National Advisory Committee for Aeronautics, Washington, D.C. ZAMM.