INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

The Pennsylvania State University

The Graduate School

Graduate Program in Acoustics

EFFECTS OF REALISTIC OCEAN FEATURES ON SONIC BOOM NOISE PENETRATION INTO THE OCEAN: A COMPUTATIONAL ANALYSIS

A Thesis in

Acoustics

by

Judith L. Rochat

© 1998 Judith L. Rochat

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 1998
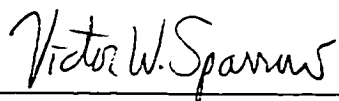
We approve the thesis of Judith L. Rochat.

Date of Signature
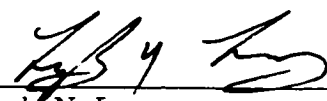
_____
Victor W. Sparrow
Associate Professor of Acoustics
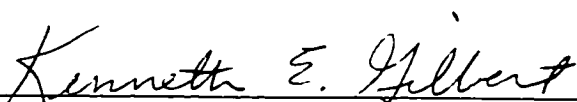Thesis Advisor
Chair of Committee

30 January 1998

_____
Kenneth E. Gilbert
Associate Professor of Acoustics

2 FEBRUARY 1998

_____
Lyle N. Long
Associate Professor of Aerospace Engineering

2 February 1998

_____
Dianna F. McCammon
Professor of Acoustics

10 Feb 1998

_____
Philip J. Morris
Boeing/A.D. Welliver Professor of Aerospace Engineering

3 January 1998

_____
Anthony A. Atchley
Professor of Acoustics
Chair of the Graduate Program in Acoustics

10 February 1998

ABSTRACT

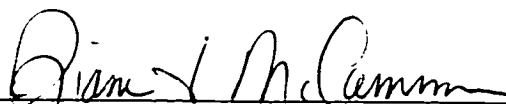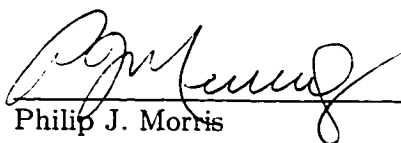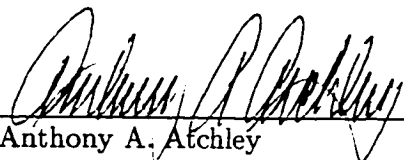The last decade has seen the revival of sonic boom research. a topic which thrived for about 10 years starting in the late 1960s. This revival is a direct result of the extensive projected market for a new breed of supersonic passenger aircraft. its design. and its operation. One area of the research involves sonic boom noise penetration into the ocean: the last decade seen a heightened awareness of noise effects on marine life, one concern being the possible disturbance of marine mammals from the noise generated by the proposed high speed civil transport (HSCT) flyovers. Although theory is available to predict underwater noise due to a sonic boom impinging upon a homogeneous ocean with a flat surface. theory for a realistic ocean. one with a wavy surface and bubbles near the surface. is missing and will be developed in this work. First. several metrics are applied to describe the underwater sonic boom noise. Then a computational method is developed which calculates the underwater pressure and intensity fields: the method is verified with proven theories. higher-order formulations. and recent applications. Using proposed parameters for an HSCT together with each realistic ocean feature. predictions are made for the underwater sound field. The curvature of a wavy ocean surface is shown to focus/defocus the incoming waveform. and the resulting underwater evanescent wave is slightly increased/decreased in pressure amplitude. It is also found that bubbles near the surface of the ocean only negligibly affect the incoming sonic boom and its associated underwater noise. Overall. it is determined that the realistic ocean features affect the penetrating sonic boom noise by modifying the underwater sound levels only one decibel or less.

Table of Contents

List of Figures

List of Tables

Acknowledgments

".   .   .   .   .

Give us this day a stable code.

Let it print numbers that don't explode,

Let it solve problems that no one has done

Since computations have first begun.

.   .   .   ."

— Sound Numbers for Computing Sound, by L.B. Felsen, appears in *Computational Acoustics. Algorithms and Applications* (1986)

Chapter 1.
Introduction

1.1. Motivation

It is well known that sonic booms are produced from aircraft traveling faster than the speed of sound. Noise associated with these high-pressure disturbances is a concern when a supersonic airplane flies over inhabited areas; thus supersonic flight has been restricted to over water or desert areas. Even though humans may not reside in these locations, other animals do.

The 1990s have seen a heightened interest in the impact of noise on marine mammals. One example is research being conducted on marine mammal behavioral response to sonic booms; this research is in response to United States policies detailed in the Marine Mammals Protection Act and the Endangered Species Act.[71] Concern lies in these behavioral responses rather than physiological harm.[14] Marine mammals include several different animals, some living in the water, some on land, but all spend time near the surface of the ocean; thus, each has the potential to hear noise created by supersonic aircraft.

The study of sonic boom noise penetrating an air-water interface is not new to the scientific community. In the 1960s and early 1970s, research on supersonic transport (SST) was in full force; incorporated in this research were several studies focusing on the sonic booms created by SSTs and their corresponding underwater sound. Unfortunately, Congress canceled the U.S. SST program in 1971,[71] leaving the development of a new U.S. supersonic airplane unrealized and studies of sonic boom noise penetration into the ocean incomplete.

In 1989 interest in an SST program was revived; the projected world aviation market showed an increase in the need for overwater flights,[71] an example being Los Angeles to Tokyo. The potential market is between 500 and 1000 new aircraft, a number associated with a profitable venture. As a comparison, there are only about 13 aircraft in the Concorde fleet. Also, the Concorde is approximately 204 ft (62.1 m) in length, holds 100 passengers, flies at a cruising speed of Mach 2 and a cruising altitude between 55,000 and 60,000 ft

(16.765 and 18.288 m), and has a flying range of approximately 3250 nautical miles (n. mi) (3740 mi. 6019 km). The Concorde has not been very economical for the airline industry, so a super Concorde (twice as many passengers) is in development by the English and French. The U.S.-proposed high speed civil transport (HSCT) would be 311 ft (94.8 m) in length,[67] hold 300 passengers.[71,67] fly at a cruising speed [67,71] of Mach 2.4 and a cruising altitude of 60.000 ft (18.288 m).[47,71] and have a range of 5500 n. mi (6329 mi, 10.186 km). This new breed of supersonic passenger aircraft is expected to reach operational status between the years 2005 and 2010.

1.2. Goal of Research

The overall goal of this dissertation is to determine the underwater sound levels due to sonic booms generated by the new HSCT. It would be possible using existing theories to make such calculations for a *homogeneous* ocean with a *flat* surface. What is not yet available are theories which include realistic ocean features: waves on the ocean surface and bubbles beneath the surface (an inhomogeneous ocean).

The addition of realistic ocean features to the study of sonic boom noise penetrating from air into the ocean can be accomplished by formulating a computational method. Using such a method. each ocean feature can be treated separately to find its individual effects on the incoming waveform. A wavy ocean surface can be studied to see if its curvature focuses or defocuses the incoming sonic boom noise, possibly causing a change in the underwater pressure disturbance. An ocean with bubbles near the surface also has the possible effect of altering the underwater pressure field, but the main concern here is the possible increase in the amount of sound transmitted through the air-water interface. A computational method can also incorporate the parameters of an aircraft not yet in existence: it is possible to study sonic booms generated by the proposed HSCT.

Determining underwater sound levels associated with sonic booms hitting the surface of the most realistic ocean possible allows predictions of the noise that marine mammals will hear. Marine biologists could use this information to predict the impact from the HSCT-generated sonic booms on these animals, treating each animal type according to its unique hearing abilities.

1.3. Dissertation Overview

The remainder of this dissertation covers relevant concepts, background information, the development and results of computational simulations, verification of methodology, and conclusions of the research.

Chapter 2 discusses the concept of sonic booms: how they are formed, information about realistic waveforms, and the interaction of sonic booms with the surface of the Earth. Chapter 3 gives the background information for sound penetrating from air into water; some of the reviewed articles will serve as a comparison to the preliminary computational studies. The first results are shown in Chapter 4, where marine mammal issues are discussed and sound levels are explained and calculated to find the appropriate description of the underwater sound due to a sonic boom impinging on the ocean surface. Chapter 5 introduces the computational method used for simulating the full sonic boom/ocean interaction by describing the computational domain parameters and deriving the algorithms used for wave propagation.

The computational simulations for a rounded sonic boom interacting with the ocean surface begin in Chapter 6. This chapter looks at the pressure field throughout the computational domain then introduces the calculation of the intensity field. Results are shown for the case of a homogeneous ocean with a flat surface; relevant studies are used for comparison. The second set of simulations is in Chapter 7 and involves the first realistic ocean feature, waves on the surface of the ocean. This set includes the study of the effects of both simple and complex curvature of the ocean surface on an impinging sonic boom and the study of the strength of the curvature effects related to the "wavelengths" of the ocean waves and sonic boom. The final set of simulations is in Chapter 8 and involves the second realistic ocean feature, bubbles near the surface. Two different bubble layer models are implemented in the computer program, and results are presented on the study of how these bubbles affect the sonic boom penetrating the ocean surface.

Chapter 9 supplies verification of the methodology used throughout the dissertation and the results obtained applying the computational method. Two different types of validation are implemented: a higher-order accuracy computational scheme and a more N-shaped sonic boom as the initial waveform. Finally, conclusions are stated in Chapter 10.

Chapter 2.
Sonic Booms

Before proceeding with any literature reviews, methodology development, simulations, or results, it is first necessary to review the basics of the pertinent waveform, a sonic boom.

Sonic booms are phenomena created by aircraft in supersonic flight. They are typically heard at the ground as a double bang. How they initiate at the aircraft location and evolve into the noise sonic booms create at the ground will be the topic explained in this chapter. This explanation is based heavily on a review of sonic booms given by Plotkin and Sutherland[16] and on Chapter 11 of Pierce's book on acoustics:[14] many of the figures are adapted from ones found in these works. Information in this chapter not related to the stated references is indicated with an alternate citation.

2.1. Initial Disturbance Caused by Aircraft

Aircraft flying faster than the speed of sound generate sonic booms. These high amplitude pressure disturbances start to occur when the airplane is in transonic flight (when the plane makes its transition from subsonic flight—slower than the speed of sound—to supersonic flight—faster than the speed of sound) and are generated continuously during supersonic flight. The *Mach number* ($M = v_{aircraft}/c_o$) is defined as the number by which one multiplies the speed of sound (c_o) in order to calculate the total speed at which an airplane is flying ($v_{aircraft}$): if it is flying at the speed of sound. it is said to be traveling at Mach 1 (twice the speed of sound would be Mach 2). A sonic boom is not a result of engine noise. but rather just of a projectile pushing the surrounding air. The pressure at the front of the projectile builds up while at the rear it expands out. Figure 2.1 depicts what happens to the sound waves as a source increases in speed from subsonic to supersonic flight.

As the aircraft travels at supersonic speeds, the high amplitude pressure disturbance forms a cone with its tip at the nose of the aircraft and its radius increasing in the direction of the rear of the aircraft. The cone has a thickness with its outer surface associated with the nose of the airplane and the inner surface associated with the tail of the airplane. This cone. called the *Mach cone*. has hyperbolic ground intercepts; these will be explained in
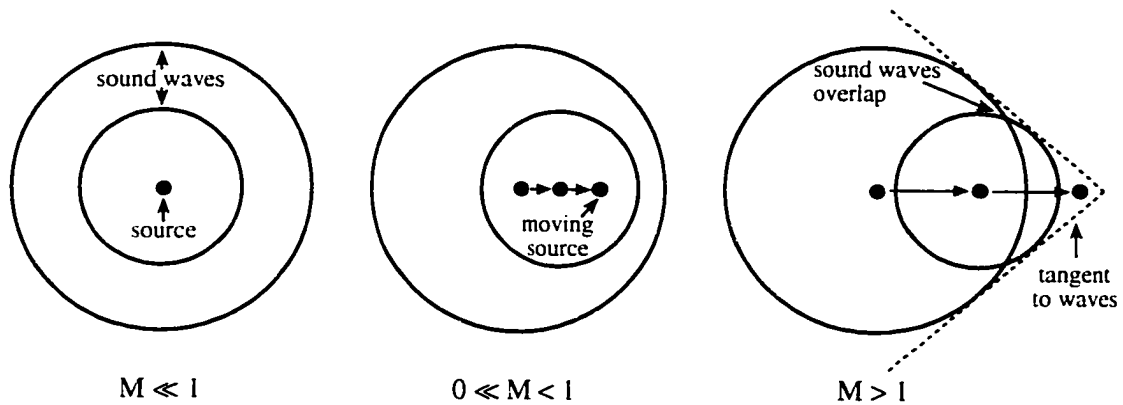
Figure 2.1: Sound waves in subsonic. transonic. and supersonic flight. M is the Mach number.

more detail later in this chapter. The Mach cone can be seen in Fig. 2.2. It should be noted that this is explained in terms of the wave viewpoint not the ray viewpoint. where the aircraft generates a ray cone. formed from rays extending outward and forward from the nose of the aircraft. Both viewpoints describe the same phenomena. the ray cone describing the sound generated at a specific time. and the Mach cone describing the sound existing at a specific time. Details of the differences can be found in the Plotkin and Sutherland review.[46] The angle θ shown in Fig. 2.2 is called the *Mach angle*: this angle sweeps from the line of the horizontal flight path to the edge of the cone. It can be expressed in terms of the Mach number as

$$\theta = \sin^{-1}\left(\frac{1}{M}\right) . \tag{2.1}$$

2.2. Propagation through Atmosphere

Near the supersonic aircraft. the pressure field is directly related to the aerodynamics of the vehicle and reflects particular geometric details of that specific aircraft: the pressure disturbance can resemble a rough sinusoid. As this high amplitude pressure disturbance propagates through the atmosphere, the waveform evolves from a somewhat sinusoidal transient pulse to an N-shaped waveform. This transformation can be explained using nonlinear acoustics.
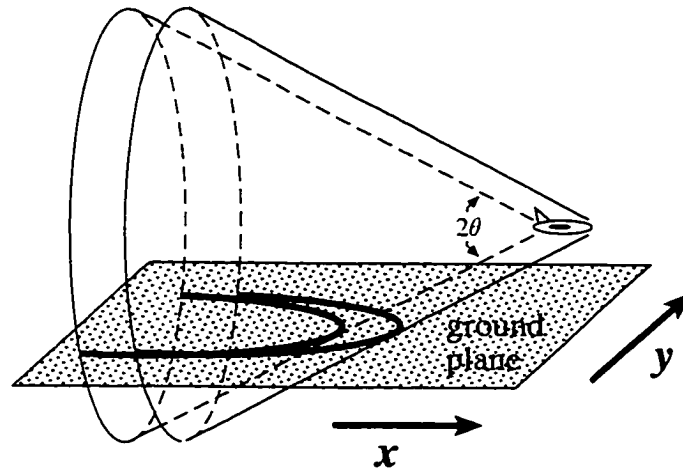
Figure 2.2: Mach cone formed by aircraft in supersonic flight.
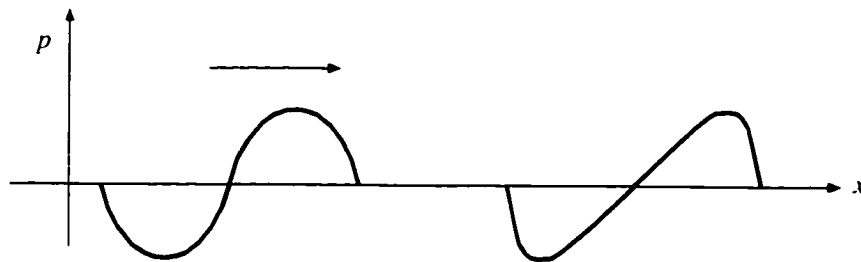
Figure 2.3: A pictorial of waveform steepening. (p is acoustic pressure. x distance.)

Each part of a waveform travels at a speed dependent on its amplitude: the peak travels faster than the speed of the zero-crossing part of the waveform, and the trough travels slower than the zero-crossing. The term for this phenomenon is *waveform steepening*. This actually occurs in linear acoustics, but the pressure values are so low that the effects are insignificant. Please refer to Fig. 2.3 to follow the signature evolution in nonlinear acoustics. Due to waveform steepening the initial sinusoidal pulse slowly transforms as it propagates. How much the waveform steepens depends on the *parameter of nonlinearity*, a function of the medium.

Eventually the waveform wants to become multivalued. but this is physically impossible. This is where one applies the *equal area rule*, valid based on weak shock theory, where nonlinearity dominates over absorption effects. A straight line can be drawn which divides the "triple-valued pressure area" into two parts of equal area. This line defines where the shock will be.[70] Please refer to Fig. 2.4.
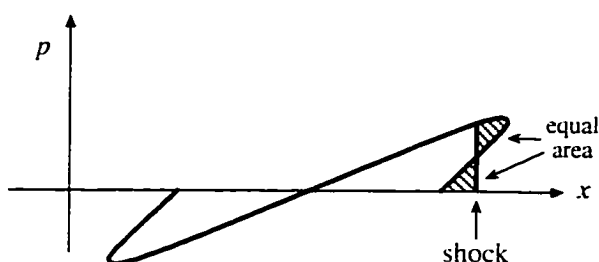
Figure 2.4: Shock formation due to waveform steepening. (p is acoustic pressure. x distance.)

The newly formed shock wave is then in the shape of the letter N. the waveform now termed an N wave. (The shape is that of an N plotted in time and of a vertically flipped N plotted in space.) Figure 2.5 is an N wave plotted in space. where $P(t)$ is the pressure amplitude of the shock as a function of time and $L(t)$ is half the length of the N wave. the length of either the positive or negative part of the waveform. The term *effective wavelength* ($\lambda_{boom} = 2L(t)$) is used here and later in this dissertation to describe the wavelength associated with the sonic boom duration. Over time $L(t)$ increases and $P(t)$ decreases. but $L(t)P(t) = L_o P_o$ always: here L_o and P_o are the initial values (when the waveform first becomes N shaped). This implies that over time, the sonic boom waveform stretches out, its amplitude decreasing and its length increasing. Figure 2.6 illustrates this phenomenon. Although the pressure/length product is conserved, energy in the waveform does decrease over time. Before the shocks appear. nonlinearities redistribute the energy into harmonics. After the shocks appear. energy is lost into the shocks.

Theory predicts a perfect N wave. Actual measurements show less than perfect N-shaped sonic booms. The atmosphere through which a sonic boom propagates determines
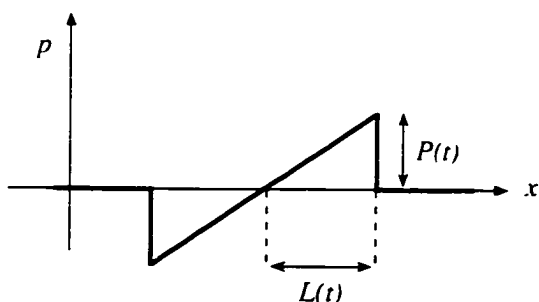
Figure 2.5: N wave where $P(t)$ is the pressure amplitude and $L(t)$ is half the length of the waveform. (p is acoustic pressure. x distance.)
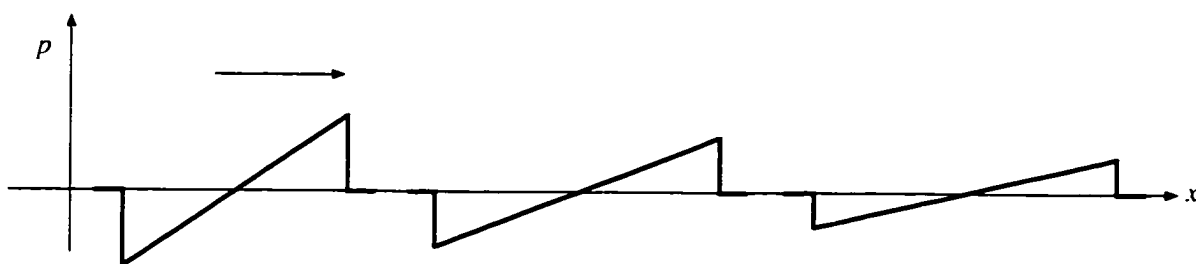
Figure 2.6: N wave stretching over time. (p is acoustic pressure. x distance.)

its signature. Different scales of atmospheric turbulence affect different parts of the waveform:[45] the smallest scales affect the rise time. larger scales cause spiking and rounding near the positions of the two shocks. yet larger scales cause uniform magnification and demagnification of the waveform as a whole. and yet larger scales cause perceptible shifts in the direction of propagation by the time the boom reaches the ground. The shock profile at the ground depends on the entire propagation history.[11] Examples of actual sonic booms will be shown in the next section.

2.3. Sonic Boom at the Ground

2.3.1. Ground intercept and boom carpet

As was seen in Fig. 2.2. the Mach cone intercepts the ground in a hyperbolic shape. A blow-up of this hyperbolic intercept is seen in Fig. 2.7. Here a perfect N wave illustrates the waveform over the entire intercept just to make it clear how the sonic boom waveform relates to the Mach cone. In reality. the signature extending out from the vertex of the hyperbola becomes more rounded. the sharp discontinuities fading.[40]
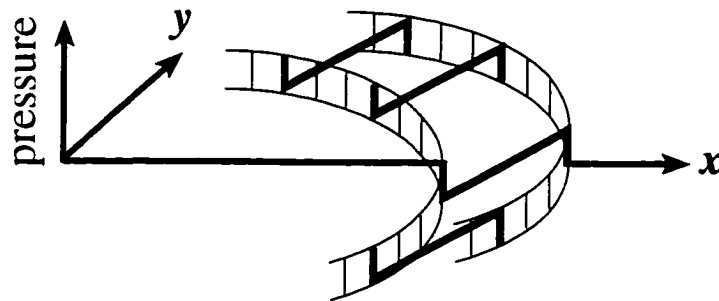
Figure 2.7: Hyperbolic intercept of the Mach cone with the ground.

During supersonic flight. the N wave moves with the aircraft. the Mach cone laying a *carpet* on the earth's surface: this carpet is swept out under the full length of a supersonic flight, its width depending on the flight and atmospheric conditions.[40] Specifically, the width of the sonic boom carpet is a function of the flight altitude. the speed of the aircraft. and the characteristics of the atmosphere; it widens with increasing altitude and Mach number. The *primary boom carpet* contains the normally observed sonic booms and results from wave propagation through only the part of the atmosphere below the aircraft. As an example, an aircraft flying at Mach 2.4 in steady flight would lay a primary boom carpet of width approximately 30 n. miles when flying at an altitude of 20,000 feet and 50 n. miles when flying at an altitude of 60.000 feet. (This information is extrapolated from Figure 17 in

Ref. 40.) A *secondary boom carpet* may exist which involves the portion of the atmosphere both above and below the aircraft: the sonic booms can reach the ground after refraction in the upper atmosphere. The disturbances in the secondary carpet have very low peak pressure values and are in the subaudible frequency range: they can be heard as rumblings inside structures.

2.3.2. Sonic boom signatures and frequency content

Examples of actual sonic boom waveforms measured at the ground are shown in Fig. 2.8[23] plotted as pressure versus time. The data were measured from supersonic aircraft flyovers.[35] The shapes of the waveforms from left to right are a classic N from an SR-71 aircraft, a peaked N also from an SR-71 aircraft, and a rounded N from an F-16 aircraft. The peak pressure amplitudes associated with these waveforms are seen in Table 2.1. For typical sonic booms, including ones other than those shown here, the peak pressure amplitudes range from 30 to 300 Pa (sound pressure level: 124–144 dB re 20 μPa, air reference, and 150–170 dB re 1 μPa, water reference). Also, the duration of sonic booms is typically 100 to 300 ms. Supersonic aircraft can create focused or super booms through maneuvers (e.g., dives, turns, or accelerations): these account for the high end of the peak pressure amplitude range.
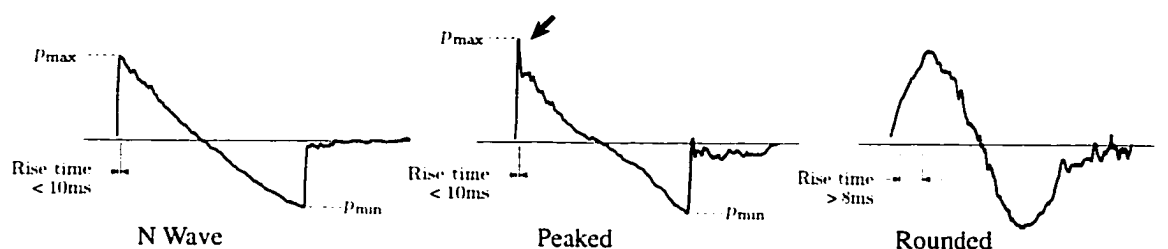
Figure 2.8: Measured sonic boom waveforms: N wave, peaked, rounded; pressure versus time. Graphics taken from Ref. 23.

The frequency content in a sonic boom waveform can be determined by taking the Fourier transform of the time-dependent signature. A perfect N wave with a peak pressure of 100 Pa and a duration of 300 ms has the frequency spectrum shown in Figs. 2.9 and

Table 2.1: Positive and negative peak pressures of actual sonic booms shown in Fig. 2.8.

sonic boom type	pos. peak pressure (Pa)	neg. peak pressure (Pa)
N wave	52.7	-41.6
peaked	91.0	-61.4
rounded	23.9	-21.6

2.10. (The spectrum figures were generated using *Mathematica*.[73]) Figure 2.9 shows the frequency spectrum calculated using a 16.484-point discrete Fourier transform (DFT): here it is apparent that important information lies in the region of 0–500 Hz. Figure 2.10 is a plot of this region. calculations made using a 1028-point DFT: here it is shown that the significant frequency content of the N wave is at the lower frequencies. Upon further inspection, an energy analysis (equations can be found in Ref. 55) reveals the following: approximately 96.8% of the energy is contained in frequencies below 40 Hz. approximately 99.7% below 400 Hz. and only 0.3% above 400 Hz.
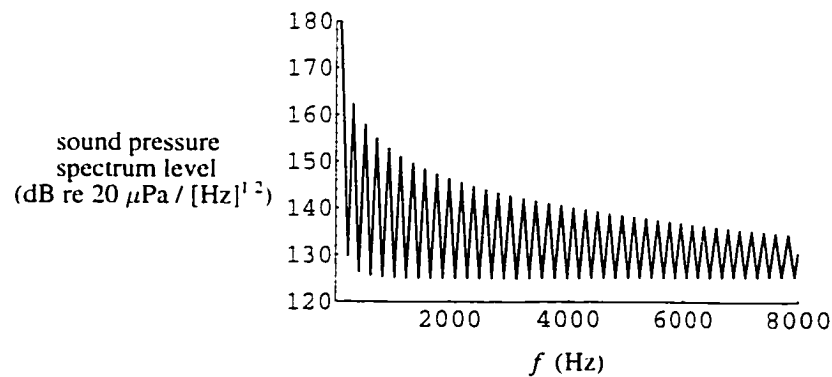
Figure 2.9: Frequency content of 100 Pa. 300 ms N wave: 16.384-point DFT. f is the frequency in hertz.

For a rounded sonic boom with a peak pressure of 100 Pa and a duration of 300 ms the frequency spectrum is shown in Fig. 2.11. It is safe to say that here that the significant frequency content is also below 400 Hz. However, comparing Fig. 2.11 to Fig. 2.10, it is
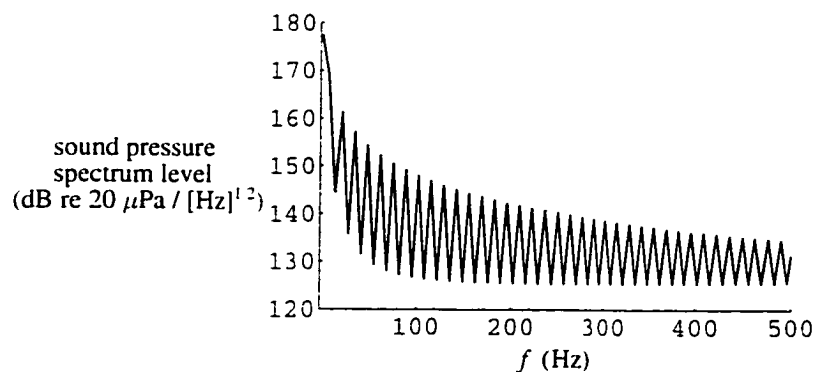
Figure 2.10: Frequency content of 100 Pa. 300 ms N wave: 1.028-point DFT. f is the frequency in hertz.

seen that at 100 Hz. for example. the sound pressure level is about 148 dB for the perfect N wave and only 100 dB for the rounded sonic boom. This is because the perfect N wave is much more broadband in frequency; the rounded waveform does not contain a substantial amount of the higher frequency content.
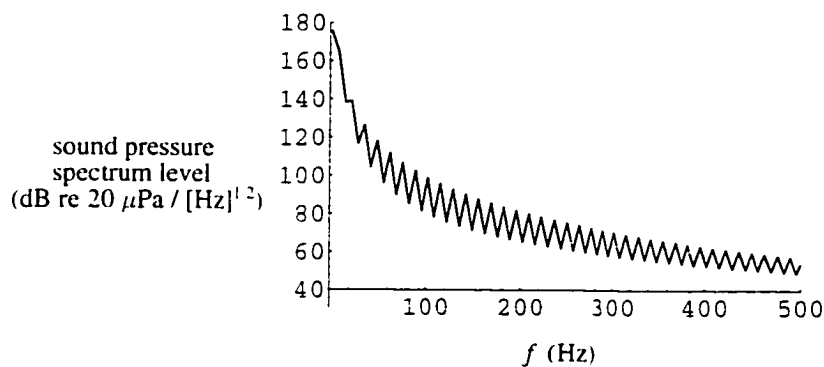
Figure 2.11: Frequency content of 100 Pa. 300 ms rounded sonic boom: 1.028-point DFT. f is the frequency in hertz.

2.3.3. Sonic boom noise

When flying over land. sonic boom effects on structures are a major concern. Sonic boom damage has included minor occurrences like broken windows (the most common complaint) and cracked plaster. to more serious rare events like the collapse of a 15th century church or the triggering of avalanches. Besides these concerns. the perception of the sonic boom noise. either by humans or other animals. is also an important issue to address.

As mentioned earlier. a sonic boom can be heard as a double bang: this is possible if its duration is on the order of 0.1 seconds or longer.[40] This amount of time is necessary for human auditory sensors to detect two distinct sounds. one for the front shock and one for the tail shock. Although no physical harm (ear drum rupture) will occur to humans until the peak pressure reaches 330 psf (15,840 Pa). typical sonic booms (0.6-6 psf or 30-300 Pa) tend to be startling and annoying. The noisiness of these waveforms is objectionable to humans since there is a startle-related reaction due to such large and rapid changes in noise level in an interval of one second.[32] When inside their homes, people tend to be more annoyed by the structure rattling than the boom itself. Sonic booms are thought to be louder when their duration exceeds 200 ms and more acceptable when the peak pressure is kept below 1 psf (48 Pa).

How loud or noisy a sonic boom can be has been a topic of intense research.[6,34,55] In several studies. the primary interest was determining if a shaped sonic boom would be more acceptable to people. Design configurations for a supersonic aircraft can be optimized to create a "minimum boom." It was shown that with this shaped boom, one that has a longer rise time or flattened peaks, the sonic boom is perceived as decreasing in loudness and increasing in acceptability. Although it is possible to build such an airplane, it probably would not be economically viable.

Besides the human response to sonic boom noise. several studies have focused on animal response. A review of many of these studies and a discussion on wildlife response to noise is found in Bowles:[4] most of the following statements are taken from this work. In animals, the response corresponding to human annoyance is aversion. usually measured by avoidance responses. Aversive levels of noise might cause wild animals to become irritable, affecting

feed intake. social interactions. or parenting. An impulsive sound—sonic booms would fall into this category—causes a startle response which is considered to be aversive. An intense startle response can include a panic reaction such as flying or running away, temporarily leaving a nest. Most terrestrial mammals cannot detect low-frequency noise, but some birds. marine mammals. and fish respond to these sounds. Since the concern that helped motivate the current research involves marine life. hearing and reactions of marine animals will be discussed in more detail in Chapter 4.

2.3.4. Sonic booms impinging upon water

A sonic boom from an aircraft in steady. level flight hits the surface of the Earth at an incident angle (angle from the line normal to the surface) equal to the Mach angle θ. Using Eq. (2.1) it is possible to calculate the incident angle for a particular Mach number. For example. a supersonic aircraft traveling at Mach 1.4 would produce a sonic boom with an incident angle θ of 45.6°: for Mach 2.4. $\theta = 24.6°$: and for Mach 4.4. $\theta = 13.1°$. Figure 2.12 depicts the incident angle formed by the sonic boom impinging upon the surface.
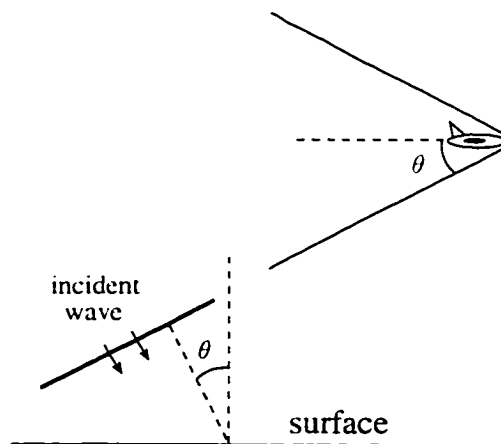
Figure 2.12: Incident angle of sonic boom impinging upon the Earth's surface.

Since a sonic boom often approximates a N-shaped plane wave by the time it reaches the Earth's surface. plane wave theory can be applied to determine transmission/reflection

of the sonic boom at the surface of the ocean. Pierce[44] applies the concepts of trace velocity, specific radiation impedance, and trace-velocity matching principle to the example of a plane wave incident on an interface between two fluids. The trace velocity v_{tr} at the interface is $c_1/\sin\theta_1$. where c_1 is the speed of sound in medium 1 and θ_1 is the incident angle. The disturbance generated within the second fluid must have the same trace velocity. Snell's law is formed from the trace-velocity matching principle:

$$\frac{\sin\theta_1}{c_1} = \frac{\sin\theta_2}{c_2}.$$
(2.2)

where c_2 is the speed of sound in medium 2 and θ_2 is the refracted angle. Figure 2.13 illustrates a plane wave traveling in medium 1 and hitting medium 2. The reflected wave (a propagating wave) in this illustration travels back into medium 1 with the reflection angle equal to the incident angle: this is governed by the law of mirrors. The transmitted wave can be either a propagating wave or an evanescent wave. the result determined by the incident angle.
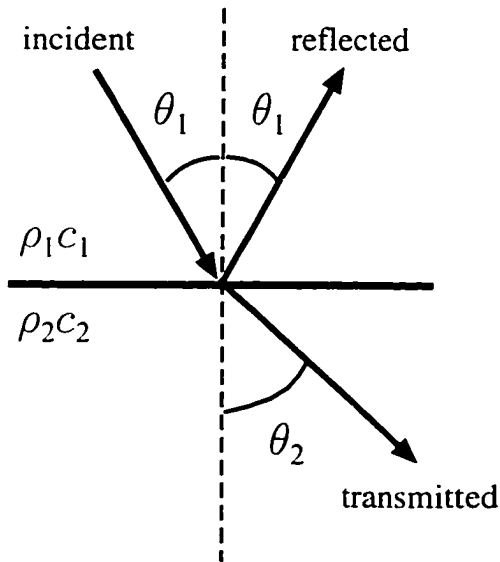
Figure 2.13: Incident. reflected, and transmitted waves going from medium 1 to medium 2. ρ_1 and c_1 are the density and speed of sound in medium 1 and ρ_2 and c_2 are the density and speed of sound in medium 2.

Using Eq. (2.2) the critical angle for determining transmission/reflection can be calculated. If $\theta_2 = 90°$ there would be no transmission of a propagating wave from medium 1 to

medium 2. Thus the *critical angle* θ_c, the incident angle which determines whether or not a propagating wave can transmit from medium 1 to medium 2, can be expressed as

$$\theta_c = \sin^{-1}\left(\frac{c_1}{c_2}\right) . \tag{2.3}$$

Using the parameters of $c_1 = 343$ m/s for the speed of sound in air and $c_2 = 1500$ m/s for the speed of sound in water, the critical angle is calculated to be $13.2°$. This incident angle corresponds to a Mach number of approximately 4.4. What this implies is that, unless the supersonic aircraft is traveling faster than Mach 4.4, the impinging waveform will be totally reflected. Recall that the projected HSCT will be flying at a speed of Mach 2.4, a Mach number associated with total wave reflection.

Even though the incident wave associated with an aircraft flying at Mach 2.4 in level flight is totally reflected, an evanescent wave penetrates the water's surface. This evanescent wave decays as a function of depth: it is *not* a propagating wave nor does it have an associated propagation angle (angle of refraction), but rather descends in a direction normal to the surface. One can think of the incident wave pushing on the surface of the water, creating a local disturbance. Technically it is realized by the boundary conditions at the air-water interface: continuity of the normal particle velocity and acoustic pressure.

In Pierce's book[44] equations are given for calculating the transmission/reflection coefficients for propagating and evanescent waves. He states that the disturbance in medium 2 is equivalent to what would be produced by a traveling flexural wave moving along the interface. He uses the analysis of sound radiation by a flexural wave moving along a wall to find the radiation impedance at the interface, Z_2, for the two-fluid case. The impedance is expressed as $Z_2 = \hat{p}/\hat{v}_z$, where $\hat{p}$ is the complex pressure amplitude and $\hat{v}_z$ is the complex amplitude of the normal particle velocity, the variables that must be continuous across the interface. Assuming medium 2 is unbounded, the solution for Z_2 around the interface is

$$Z_2 = \begin{cases} \frac{\rho_2 c_2}{\cos\theta_2} & \text{if } \sin\theta_1 < \frac{c_1}{c_2} \\ \frac{-i\rho_2 c_2}{\beta_2} & \text{if } \sin\theta_1 > \frac{c_1}{c_2} \end{cases} . \tag{2.4}$$

where $-\beta_2^2 = 1 - \left(\frac{c_2}{c_1}\right)^2 \sin^2\theta_1$. The case applicable for this work is $\sin\theta_1 > \frac{c_1}{c_2}$, where the impedance is purely reactive. Also, $Z_1 = \frac{\rho_1 c_1}{\cos\theta_1}$. The reflection coefficient is then written

$$\mathcal{R}_{1,2} = \frac{Z_2 - Z_1}{Z_2 + Z_1} . \tag{2.5}$$

and the transmission coefficient is $T_{1,2} = 1 + R_{1,2}$. The expression for the underwater evanescent pressure field for an incident plane wave of constant frequency can be written

$$\hat{p} \propto T_{1,2} e^{i(\omega/c_1)(\sin\theta_1)x} e^{-(\omega/c_2)(\beta_2)z} . \tag{2.6}$$

where again $\hat{p}$ is the complex pressure amplitude. Although the incident wave in this dissertation is a transient waveform, Eq. (2.6) is important to the understanding of the underwater pressure field: both β_2 and $T_{1,2}$ are functions of θ_1 so their values are Mach number dependent, a point which will be referred to in Chapter 6. It is also apparent in Eq. (2.6) that the pressure decays with depth, the higher frequencies decaying more rapidly than the lower frequencies $(e^{-i(\omega/c_2)(\beta_2)z})$.

The pressure amplitude at the air-water interface is double that of the amplitude in the air: the impedance change is so large going from air to water that the incident wave sees a hard reflecting boundary. For an incident wave that is totally reflected, $R_{1,2} = 1$: this implies that $T_{1,2} = 2$ or that the pressure amplitude has been doubled. The pressure doubling at the surface is applied to the penetrating evanescent wave. As an example, if an incident wave has a peak pressure amplitude of 50 Pa, then its amplitude at the interface would be 100 Pa: the peak pressure amplitude of the evanescent wave just under the surface would be 100 Pa which then exponentially decays with depth.

Figure 2.14 shows an airborne wave incident on and reflected from the surface of the water. Also shown is the underwater evanescent wave. Three chronological times are represented in order to visualize the progression of the plane wave across the surface of the water.

The underwater evanescent wave discussed in this section is what will be calculated in later chapters of this dissertation, the effects of the ocean waves and bubbles on the evanescent wave being studied. The following chapter will review published research which encompasses sound penetration from air into water.

reflected
plane wave

incident
plane wave

evanescent
wave

air

water

time 1   time 2   time 3
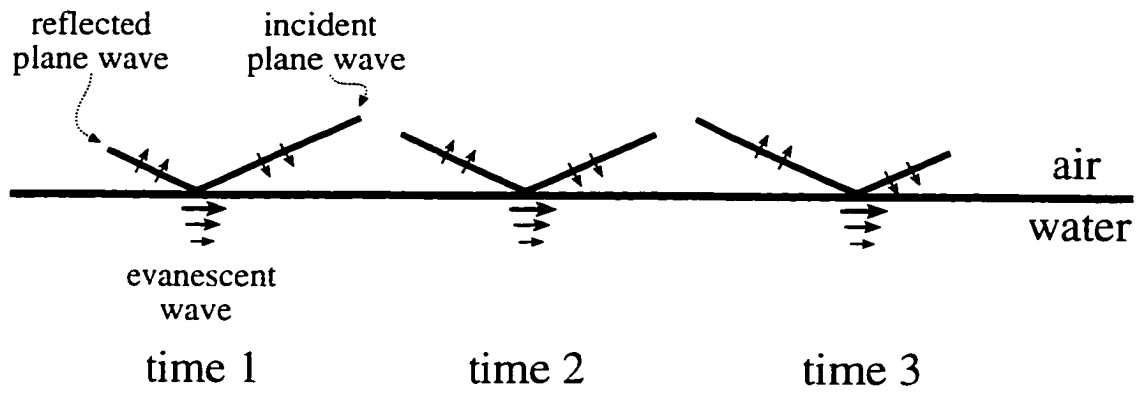
Figure 2.14: Incident. reflected. and evanescent waves along the air-water interface. Here it is assumed that the incident angle is greater than 13.2°. Three chronological times are represented.

Chapter 3.
Sound Penetration from Air into Water: Background

Starting in the late 1960s. the study of sound in air penetrating into water became a topic of interest that spanned a decade. The prospect of supersonic aircraft fueled this interest. It led to analytical and experimental research on noise sources in air. some specifically being sonic booms. The interest in noise penetration into the ocean has since been revived because the last decade has seen not only the design of a new high speed civil transport aircraft but also the need to address marine environmental issues.

Many of the articles reviewed in this chapter will be pertinent in developing applications and making comparisons later in this dissertation. Some of the articles are reviewed solely to gain a sense of where research stands on underwater noise due to an airborne source.

3.1. Sound Source in Air, Receiver in Water

Although basic research on sound transmission through an air-water interface was accomplished as early as 1838.[24] several researchers addressed the issue in the 1970s. A brief citation of relevant studies follows chronologically. Each author addresses the issue of sound penetration into water. some dealing with a rough ocean surface.

A study on the transmission of sound from a monopole source through a finite. corrugated boundary between fluid media was completed by Macaluso in 1970.[39] For this work. the boundary was a sinusoidal surface. The author developed a mathematical model and computational procedure to predict the acoustic field in the fluid then succeeded in acquiring experimental data for sound being transmitted from air into water. The theory was validated by the experiment and was shown to apply to frequencies down to those associated with acoustic wavelengths equal to the corrugation. or sinusoidal, surface wavelength.

In 1972 Urick[66] studied the sound emanating from a subsonic aircraft. He showed that it can be important to take into account all of the paths that sound takes traveling from a source in air to a receiver in water. He used wave theory rather than ray theory in order

to include an evanescent wave (also known as a lateral or inhomogeneous wave) as one of these paths.

In 1973 Medwin, et al.[12] studied the spectral characteristics of sound transmission through a rough sea surface. Applying Helmholtz-Kirchhoff theory to transmission from air into water they provided useful predictions of the dependence of transmitted sound pressure level on, among other things, surface roughness. They showed that for low roughness, transmission is essentially the same as for a smooth surface, but for higher roughness further studies needed to be done which take into account the evanescent wave.

Lubard and Hurdle[38] in 1976 conducted an experimental investigation on acoustic transmission from air into a rough ocean. The objective of their experiment was to determine transmission loss for a source in the air to an underwater hydrophone as a function of, among other parameters, sea state and incident angle. Data taken with wind speeds ranging from 3-15 knots showed more transmission for these rough ocean surfaces than predicted for a smooth surface, with the excess increasing with smaller incident angles. The underwater source strength increased with increasing ocean roughness.

In 1978 Meecham[43] looked at point source transmission through a sinusoidal ocean surface. He was interested in the effects of ocean swell (curvature of the ocean surface) on the underwater pressure field. Using a high-frequency approximation, he analytically found that for most cases there was relatively little change in the received sound field as a result of the sinusoidal surface. However, due to various diffracted components being transmitted into the water, the underwater sound field could be moderately enhanced.

Although the research mentioned in this section deals with a sound source in the air and a receiver in the water, it should be noted that many of the authors included propagating waves penetrating the ocean surface, not just evanescent waves, and at least one focused only on high frequencies.

3.2. Sonic Boom in Air, Receiver in Water

Important work was done in the late 1960s and early 1970s on the study of sonic booms interacting with an air-water interface. Through this research, analytical methods were developed to characterize the sound field under a flat ocean surface. A few years later, experiments were performed with a flat water interface, the results validating the analytical theories. In recent years, research has included not only additional flat surface studies but has also added some more realistic ocean features when studying the penetration of sonic boom noise into the ocean.

Reviews of the relevant studies will be more thorough than in the previous section since these researchers have concentrated specifically on sonic boom-type waveforms interacting with the ocean.

3.2.1. Historical applications

3.2.1.1. Analytical theories

In 1968, Sawyers[53] derived an expression for pressure in a homogeneous fluid half-space due to a sonic boom interacting with the surface. This study made several assumptions including the following: the fluid has an undisturbed, flat surface; the sonic boom is idealized as a perfect N-shaped waveform; the N wave is infinitely long in the direction perpendicular to its direction of propagation; the speed of the aircraft is less than the speed of sound in the water; and that linear acoustics is valid because the sound levels are low enough in amplitude by the time the sonic boom propagates to the Earth's surface.

Within a two-dimensional geometry, Sawyers' derivation starts with the acoustic wave equation, solves the Fourier-transformed equation, then inverts the transform to yield an expression for sound pressure as a function of time t, horizontal distance x in meters along

surface. and depth z in meters into the water. This expression is given as

$$\left(\frac{\pi}{p_o}\right)p\left(x,z,t\right) = \arctan\left[\frac{m\left(t-T+x/V\right)}{z}\right] + \arctan\left[\frac{m\left(t+x/V\right)}{z}\right]$$
$$+ \frac{2}{T}\left(t-T+\frac{x}{V}\right)\arctan\left[\frac{m\left(t-T+x/V\right)}{z}\right]$$
$$- \frac{2}{T}\left(t+\frac{x}{V}\right)\arctan\left[\frac{m\left(t+x/V\right)}{z}\right] + \frac{z}{mT}\log\frac{z^2+m^2\left(t+x/V\right)^2}{z^2+m^2\left(t-T+x/V\right)^2} . \qquad (3.1)$$

where

$$m = V\left(\frac{1-V^2}{c^2}\right)^{-\frac{1}{2}} .$$

In these equations. p is the acoustic pressure. p_o is the peak pressure of the sonic boom. T is the duration of the sonic boom. V is the speed of the aircraft. and c is the speed of sound in the water. The values of the arctan functions are contained in the interval $(-\pi/2, \pi/2)$. It should also be noted that for a single frequency the pressure field decays as $e^{-|\omega|z/m}$, where ω is the angular frequency.

Sawyers nondimensionalizes Eq. (3.1) then plots the the nondimensional pressure values at several depths. These results indicate. as he states. that whereas the shock wave in air has an abrupt head and tail. the pressure disturbance underwater is seen to begin and end gradually: the abruptness of the pressure wave diminishes with increasing depth (i.e. the high frequencies die out quickly). In other words. the underwater pressure field is evanescent. the waveform decaying as a function of depth. Figure 3.1 shows an example of the evanescent wave at depths of 0. 10. 25. and 50 meters. These waveforms were generated in *Mathematica* using functions written by Sparrow:[58] these functions are based on Eq. (3.1).

In 1970. Cook[12] also studied a perfectly N-shaped sonic boom waveform interacting with a flat water surface. One of the focuses of the study was to find the sound pressure distribution underwater caused by the incidence of this sonic boom onto the water surface. Here, it was again assumed that the aircraft speed was less than the speed of sound in the water. One of the differences between Cook's work and Sawyers' work is that Cook includes the effect of the different ambient densities in air and water.

The mathematical derivation starts with the underwater pressure, first expressed as an evanescent wave. its amplitude decreasing exponentially with depth. This expression, in combination with the Fourier transform of the function for the waveform in a moving coordinate system. gives the equation for the underwater pressure in terms of integrals
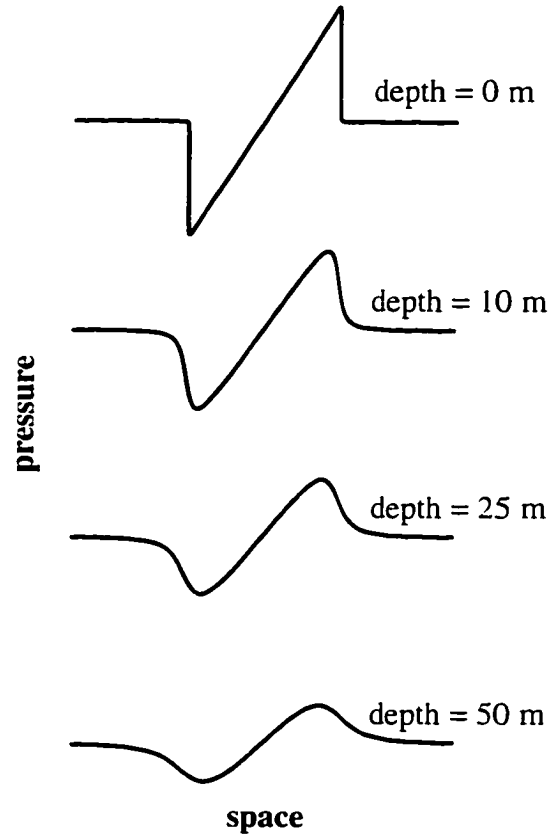
Figure 3.1: Evanescent wave at depths of 0. 10. 25. and 50 meters.

which can be solved using Laplace transforms. This underwater pressure p_w as a function of x'. the horizontal component in a moving coordinate system. and h. a function of the vertical component. the speed of the aircraft M. and the sound speeds of the media. can then be expressed as

$$p_w(x'.h) = 2\cos\Delta\,(I_1\cos\Delta + I_2\sin\Delta)\ .$$

where

$$\pi I_1(x'.h) = -\frac{h}{2}\log\left[\frac{h^2 + (x'+1)^2}{h^2 + (x'-1)^2}\right] + x'\arctan\left[\frac{2h}{h^2 - 1 + (x')^2}\right]\ .$$

$$\pi I_2(x'.h) = -\frac{x'}{2}\log\left[\frac{h^2 + (x'+1)^2}{h^2 + (x'-1)^2}\right] + h\arctan\left[\frac{2h}{h^2 - 1 + (x')^2}\right] - 2\ .$$
(3.2)

$$h = \mu z'\ .$$

$$\mu = \left(1 - \frac{M^2}{W^2}\right)^{\frac{1}{2}}\ .$$

In these equations Δ is the phase angle. z' is the vertical component of the moving coordinate system. and W is the ratio of the speed of sound in water to the speed of sound in the atmosphere. The term $Q = \arctan\left[\frac{2h}{h^2-1+(x')^2}\right]$ is a multiple-valued function. For $h \neq 0$. $0 < Q < \pi$. For $h = 0$. $Q = 0$ for $-\infty < x' < -1$ and $1 < x' < \infty$: $Q = \pi$ for $-1 < x' < 1$: and $Q = \pi/2$ at $x' = \pm 1$.

As stated in Cook's paper. Sawyers expression for the underwater pressure agrees with the I_1 part of Eq. (3.2). It was determined that although the function I_2 is also a part of the complete underwater sound field. its contribution is negligible. Cook also illustrates the sonic boom waveform decaying as a function of depth.

3.2.1.2. Experiments

In 1968 Young[74] presented a paper on an experiment he performed which measured sonic boom penetration into the ocean. The sonic boom source used was an F-8 aircraft flying at Mach 1.1 or Mach 1.15 over a relatively flat ocean (2-ft. swell). Sound pressure and sound exposure levels were plotted as a function of frequency for the sound above the water and at several depths under water. Results show that a decaying waveform exists underwater due to the sonic boom impact. Numbers indicate that for the peak sound pressure level at frequencies above 125 Hz. a hydrophone located at a depth of 20 feet measures about 25 dB less than that in air: for the sound exposure level the difference is from 15 to 20 dB. The reductions at a 160-foot depth are about 5 dB greater than the 20-foot depth results.

In the early 1970s Waters and Glass[68,69] experimentally simulated the penetration of sonic boom energy into the ocean. They performed an acoustically scaled experiment using dynamite caps to produce spherically spreading N waves which impinged on a small body of water. The following assumptions were applied: the aircraft is in horizontal steady flight: the aircraft speed is less than the speed of sound in the water: the ocean surface is flat. the ocean itself is homogeneous: only symmetrical N waves having zero rise time are produced: and the results apply only to small regions of the intersection of the Mach cone with the ocean surface. where the plane wave approximation is valid. Results from the performed experiments indicated that in the underwater sound field the amplitude of the N wave decreased rapidly with depth: also there was a rounding of the initial waveform as a

function of depth. They concluded that the measured behavior was generally that predicted by the existing theories.[53,12]

Also in the early 1970s, Intrieri and Malcolm[29,41] performed a ballistic range investigation of sonic boom overpressures in water. Their experiments included measuring underwater pressure produced from flights of small gun-launched models over a water surface in a ballistic range. The models flew at speeds both less than and greater than the speed of sound in water. It was assumed that for frequencies pertinent to full-scale sonic booms, viscosity effects are essentially non-existent regardless of depth. It was also assumed that the radius of curvature of the hyperbolic intercept of the Mach cone at the vertex is large in comparison to the length of the N wave and depths of water of interest; it was therefore safe to consider the incident wave to be planar. Results indicated that for a propagating wave, where the speed of the model is greater than the speed of sound in water, there was essentially no loss in peak pressure due to water depth. Where the speed of the model was less than the speed of sound in water, an evanescent wave was created which matched the existing analytical theories.[53,12]

3.2.2. Recent applications

3.2.2.1. Theories: analytical and numerical

In 1995 Sparrow[58] numerically studied the effect of supersonic aircraft speed on the penetration of sonic boom noise into a deep ocean. Sawyers' theory[53] was written into *Mathematica*,[73] a symbolic manipulation program; here it was assumed that the water surface is flat and that the aircraft is flying at a fixed altitude. By varying the Mach number and studying the pressure at a fixed depth, it was shown that faster aircraft speeds produce higher peak pressures at this fixed depth. Then by looking at different Mach numbers and studying the pressure as a function of depth, it was shown first, that the pressure field reveals a decaying waveform—as expected from Sawyers' theory—and second, that for increasing Mach number the peak pressure amplitude increases at each of the fixed depths. In addition, the observation was made that at greater depths, the effect of increasing the Mach number was enhanced. There was a difference of 15 dB peak sound pressure level

going from Mach 1.4 to Mach 3.4 at a depth of 100 m: at a depth of 10 m. however, the increase was only 5 or 6 dB.

In 1996 Cheng, et al.[8,9] performed a study of theoretical and computational issues relating to sonic boom propagation and its submarine impact. They focused on the needed improvements in the known sonic boom prediction methods: the improvements include, among other things, the prediction of sonic boom noise penetration into the water. This improvement extends a flat water analysis to one which incorporates ocean waves. Besides the references just cited, Cheng and Lee[10](1997) also address the ocean surface waviness issue. For the ocean surface it is assumed that the wave height to wavelength ratio is small ($\ll 1$) and is assumed to have a sinusoidal profile. Lifting-line theory from aerodynamics was employed to study the flat water case, then the use of oscillating supersonic airfoils adds waviness to the problem. Results indicate that unlike the penetration depth for their flat surface model which is only a fraction of the sonic boom effective wavelength, the disturbance through the curvy surface can reach 5 to 10 times the sonic boom effective wavelength. They concluded that ocean surface waviness augments the underwater pressure field. It should be noted that it is unclear in these papers how the parameters they use for ocean waves compare to realistic ocean waves or the approximations thereof found in the open literature.

Sparrow and Ferguson[60] in 1997 numerically implemented a method based on Cook's theory[12] to study the penetration of shaped sonic boom noise into a flat, homogeneous ocean. The analytical method by Cook was expanded in order for the initial waveform to take any arbitrary shape instead of just a perfect N wave. At the point where Cook applies Laplace transforms, they performed a Fourier synthesis using discrete Fourier transforms: the method for finding pressure as a function of depth was then programmed in *Mathematica*.[73] Using a perfect N wave as the initial waveform, the underwater results matched those calculated by theory. Several other initial waveforms were applied: a hypothetical F-22 airplane sonic boom (close to an N-shaped waveform) revealed a sound level decrease with depth much greater than for a perfect high speed civil transport (HSCT) N wave; a hypothetical flat-top HSCT sonic boom showed a decrease in sound level approximately the same as for the perfect N wave: for an experimentally measured F-15 airplane sonic boom, the rough perturbations in shape immediately leveled to the same rounded waveform as seen

for a perfect N wave under water: and finally. for an experimentally measured F-16 airplane sonic boom. the spikes in its U-shaped waveform did not persist in the underwater sound field. Overall it was concluded that atmospheric turbulence effects or other perturbations on incident waveforms have a minimal effect on the underwater sound field below 16 m.

3.2.2.2. Experiments

A paper by Desharnais and Chapman[17](1997) describes underwater measurements of a sonic boom. During a sea trial on the Scotian Shelf. a vertical array of hydrophones. there to record ambient ocean noise samples. unexpectedly recorded a sonic boom. They extrapolated that the source of this sonic boom was an Air France Concorde in flight from Paris to New York. The conditions of the experiment included a sand bank at 76 m depth. low winds and calm seas (0.25 m swell. 10 kn wind). the plane was flying at a speed of approximately Mach 2.0. and the hydrophones were located at depths ranging from 16.5 m to 65 m under water. Signal measurements indicated that an evanescent somewhat-N-shaped waveform was recognizable in the water. with the typical exponential decay of the amplitude with depth. Although the flight was not directly overhead. they applied a modified version of Sawyers' theory[53] which accounts for the path offset. The first part of the observed waveform was modeled successfully using this theory. Here it should be noted that since these recordings were unintentional. there was not a microphone in the air to record the initial waveform. There are some oscillations immediately following the decaying sonic boom waveform which might be better explained if the initial waveform were known but instead have been speculated to be the excitation of a low-frequency seismic mode at the ocean/seabed boundary and not reflections from the ocean bottom.

3.3. Conclusions and Remarks

Several researchers have addressed the issue of noise penetration from air into water. For a general noise source in the air. it was shown that evanescent waves are a definite contribution to the underwater noise and that curvature on the water's surface enhances this underwater pressure field. Theory has been developed for a perfect N-shaped sonic boom penetrating a flat air-water interface: the underwater evanescent wave has been validated by simulated experiments. Actual underwater data has been recorded of sonic booms where the ocean surface was relatively flat: results seem to follow the known theories. Recently, additions to the known theory include effects of aircraft speed. surface waviness. and shaped sonic booms.

The inclusion of realistic ocean features in the problem of sonic boom noise penetration into the ocean still needs to be addressed. Research on a wavy ocean surface is ongoing. Cheng. et al.[8-10] have presented initial results using an analytical method and the current author simultaneously has addressed the problem computationally, where the methodology and results will be revealed later in this dissertation. Also the effects of an inhomogeneous ocean—an ocean containing bubbles near the surface—have not yet been determined: these also will be discussed later in this work.

Chapter 4.
Sound Levels Under Water

As was previously mentioned. the last decade has seen a new awareness of marine environmental issues. Noise affecting marine mammals could include sonic booms from the proposed HSCT. It is not clear what exactly a marine mammal would hear or how it would be affected. but sound levels underwater can be calculated. weightings applied to get a better sense of what is being heard rather than just a physical quantity.

4.1. Environmental Concerns: Marine Mammals

This section is based heavily on Ref. 49. Information in this section not related to Ref. 49 is indicated with an alternate citation.

The marine mammals of interest are those whose eating grounds, mating grounds, or migration paths coincide with the sonic boom carpet created from the supersonically flying aircraft. This includes two orders of marine mammals: cetacea and carnivora. The two main groups of living ceteceans include odontocete. or toothed. whales and mysticete. or baleen, whales. Toothed whales include such animals as killer whales. beluga whales. dolphins. and porpoises. Examples of baleen whales are bowhead whales. humpback whales. blue whales. and gray whales. Types of marine carnivores. the second order. include pinnipeds. sea otters. and polar bears. Pinnipeds are such animals as seals. sea lions. and walruses. Because of the diversity among all marine mammals. the potential effects of sonic booms depends on the type of animal involved.

In trying to evaluate the effects of noise on marine mammals, it is first important to describe the ambient noise in the ocean. Types of environmental background noise are wind waves. precipitation. seismic activity. biological. sea ice. and thermal (molecular agitation). Background noise also includes distant human activities: shipping traffic noise, industrial plants or construction activities on shore. offshore oil industry activities. naval operations, and various types of marine research involving sound emission. Although varying. an approximate overall ambient underwater noise level (sound pressure level) is 90 dB re 1 μPa.

Since part of this research shows the effects of wind wave height on a sonic boom waveform, it is appropriate to mention that ambient noise levels tend to increase with increasing wind speed and wave height. The change in decibels from a sea state of 0 (wave height 0 m) to a sea state of 6 (wave height ≈ 4 m) is approximately 25 dB for all frequencies. (Sea state values and descriptions are found in the Beaufort Scale—available in many references including Refs. 49, 7, 3, and 22.) It is therefore relevant to keep in mind, for later chapters, that when studying increases in underwater sound levels due to surface roughness, ambient noise levels are 25 dB higher than for the calm ocean case. It should also be noted that ambient noise is dominated by wind waves for the frequency range of a few hundred hertz to approximately 30 kHz.

Sounds and hearing of marine mammals are quite varying for each species. There is a fair amount of data available on sounds, types ranging from songs and moans to tonal whistles and pulsed sounds used in echolocation. Although audiograms (graphs relating sensitivity to frequency) are available for many different marine mammals, there still is a significant amount of data to be collected in order to fully understand the hearing of all marine mammals. A very brief review of the different types of marine mammals follows. Data show that most baleen whales emit sound pressure levels of 150-190 dB re 1 μPa (at 1 meter); some have fundamental frequencies as low as 20 Hz. Their most sensitive hearing is in the range of 20-50 Hz, but they can hear even lower frequencies well. For toothed whales, most energy of their sounds is typically near 10 kHz at a level of 100-180 dB re 1 μPa. However for echolocation, these animals can vocalize pulsed sounds in the frequency range of 30-100 kHz, where the peak level can reach 210 dB re 1 μPa or higher. Toothed whales' hearing is most sensitive above 10 kHz where the upper limit is anywhere between 64 and 100 kHz. They can hear at low frequencies, but the sensitivity is poor. Pinnipeds vocalize mostly at frequencies ranging from less than 1 to 10 kHz; the source levels range from 95-178 dB re 1 μPa. These animals' hearing varies, but a typical sensitive range is 1-50 kHz.

With any of the marine mammals, it is also important to look at the critical signal-to-noise ratio for particular frequencies. The critical ratio (CR) is defined as the amount by which a pure-tone signal must exceed the spectrum level background noise in order to be audible. CR values typically range from 15 to 40 dB; however, in certain cases, the CR

value may actually be slightly negative. If a signal does not match or exceed the CR it will be masked behind the background noise, and the animal will not perceive the sound.

In addition to the hearing of the marine mammals. it is very important to include the psychoacoustic effects involved with different types of noise. Two terms are defined here to help understand disturbance reactions. *Behavioral habituation* is defined as the progressive waning of responses to stimuli that are learned by the animal to lack significance. *Sensitization* is defined as the increasing responsiveness over time. An animal that has habituated no longer behaviorally reacts to a man-made noise source: in contrast, an animal that has become sensitized to a noise reacts more strongly with each occurrence.

The man-made noise of consequence here is aircraft noise. Most documented disturbance reactions due to aircraft involve pinnipeds. Effects of sonic booms on pinnipeds have been studied. but effects on cetaceans have yet to be evaluated. The existing data on cetaceans involve just aircraft overflights. Effects of sonic booms from supersonic aircraft are special cases because of their high levels and sudden onsets. For seals. it was observed that sonic booms caused a startle reaction involving some movement or stampedes into the water: there was no evidence of mortality due to trampling or abandonment. Toothed whales may react to aircraft overflights (not sonic booms) by diving, submerging for longer periods. slapping the water with flukes or flippers. or swimming away from the aircraft track. Baleen whales may react to low flying aircraft by hasty dives. turns. or other changes in behavior. responsiveness depending on the activities and situations of the whales. Whales actively engaged in feeding or social behavior often seem rather insensitive: whales in confined waters. small groups. or with calves sometimes seem more responsive.

To get some idea of how a whale may react to a sonic boom, one can evaluate their reactions to other impulsive sounds. Following are some documented cases of reactions to impulsive sounds. It was found that 90% of gray whales behaviorally reacted to a broadband source (an airgun) of sound pressure level 180 dB re 1 μPa. In a different study, it was found that gray whales. when exposed to an airgun source at or greater than a sound pressure level of 160 dB re 1 μPa. were deflected from their migration path along the California coastline.[20] Humpback whales seemed startled when an airgun source reached between 150 and 169 dB re 1 μPa. Gray whale reactions to explosions ranged from seemingly unaffected to interrupted migration or elicited "snorkeling" behavior (breathing out under water).

Captive false killer whales showed no obvious reaction to single noise pulses from small charges where the received level was 185 dB.

As is seen by the above mentioned decibel levels. sonic boom noise levels (typically 150–170 dB re 1 μPa) may cause a behavioral reaction in marine mammals. It is unlikely. however. that these levels would cause any hearing damage.[14]

Overall. noise from human activities sometimes causes pronounced short-term behavioral reactions and temporary displacement. The continued presence of various marine mammals in certain areas. despite much human activity. suggests that many marine mammals are quite tolerant of noise and other human activities. However. there may be no suitable alternative locations for some marine mammals: they may be just tolerating the noise and/or experiencing stress.

4.2. Sound Levels

Recently. sound levels were calculated as a function of depth in order to determine the volume of the underwater sound field.[58] But it should be emphasized that what was calculated was peak sound pressure level. a purely physical quantity that does not account for the change in spectral content of the penetrating sound wave as a function of depth. This purely physical quantity also does not account for the way in which a sound is perceived. either by humans or other mammals.

It is necessary to apply frequency weighted spectra to determine the depth-dependent sound levels perceived by a listener. The weightings or noise descriptors used are determined by the type of sound and the listener.

Even though the significant frequency content in a sonic boom is below 400 Hz. the most significant being below 40 Hz, a sonic boom is characteristically impulsive: this implies that high frequency content does exist. As explained in Chapter 2. most of the sonic boom energy, about 99.7%. is contained in frequencies below 400 Hz: about 0.3% of the energy is in frequencies above 400 Hz. For an evanescent wave associated with a sonic boom impinging on the ocean surface the higher frequencies diminish rapidly, leaving a waveform composed mostly of lower frequencies. These distorted waveforms, somewhat less impulsive

than N-waves. still have a significant amplitude and short duration: it remains appropriate to apply known sonic boom noise descriptors.

Currently there are no noise descriptors available for marine mammals to help evaluate how they may perceive a sonic boom. In an effort to examine a wide range of potential descriptors, both unweighted and weighted metrics are employed. Although the weighted metrics were extrapolated from human data and anatomy. the calculated sound descriptors may help determine some realistic measure of annoyance experienced by marine mammals. Applying several different frequency weightings to the distorted sonic boom waveforms should show how relative measures of loudness or noisiness change with depth.

For a subjective description of sonic boom loudness. it is most preferred to use the Mark VII perceived level (PLdB) of Stevens[62] and next A- or C-weighted sound exposure levels (L_E).[6] The physical description of sonic booms is better represented by peak sound pressure levels or unweighted (flat) or C-weighted sound exposure levels.[6] Applying both the subjective and physical descriptors allows for a comprehensive evaluation of sonic boom noise.

The Mark VII perceived level. the most recommended sonic boom noise descriptor. is closely aligned with the behavior of the human auditory system in the subjective evaluation of acoustic energy. It utilizes a set of frequency-weighting contours based on an average of 25 experimental contours. Version VII was extended to the lower frequencies in order to facilitate the calculation of the perceived level of impulses and sonic booms. A simple equation does not exist for this noise metric.

Sound exposure level is defined as the time integral of sound level over the course of a single event.[54] It is written in equation form as

$$L_E = 10 \log \int_{-\infty}^{\infty} \frac{p_{rms}^2}{p_{ref}^2} dt / \tau_{ref} \ . \tag{4.1}$$

where p_{rms} is the rms pressure. p_{ref} is the reference pressure, and τ_{ref} is the reference time, taken to be 1 second. Since this research considers a sonic boom to be 2 events (2 bangs), 3 dB is subtracted: only the energy from half the waveform, either the positive pressure or the negative pressure, should be included in the single event. (Equation (4.1) includes the energy of the entire waveform.) The equation that will actually be used for calculations is

in a discrete form: it is written as

$$L_E = 10 \left[\log \Delta t \sum_{i=1}^{\text{NoSamp}} \frac{\left(p_{peak}\left(i \right) / \sqrt{2} \right)^2}{p_{ref}^2} \right] - 3 \ . \tag{4.2}$$

where Δt is the time interval between samples. p_{peak} is the peak pressure, and NoSamp is the number of pressure samples. Unweighted sound exposure level. written L_{UE}, is equivalent to L_E.

Two different weightings are recommended for the sound exposure levels as descriptors of sonic boom noise: A weighting (mentioned in Ref. 54) and C weighting (mentioned in Ref. 6). An A-weighted sound exposure level is written as L_{AE} and the C-weighted sound exposure level as L_{CE}. The A weighting roughly accounts for the frequency response characteristics of human hearing. while C weighting places more emphasis on sounds of low frequency. A plot of the different weightings can be seen in Fig. 4.1.[44] Here $\Delta L(f)$ is the decibel amount to be subtracted from the unweighted measurement at a particular frequency. Simple equations for both A and C weighting are found in ANSI Standards.[2] The equation for C weighting is

$$W_C = 10 \log \left(\frac{K_1 f^4}{\left(f^2 + f_1^2 \right)^2 \left(f^2 + f_4^2 \right)^2} \right) \ . \tag{4.3}$$

and the equation for A weighting is

$$W_A = 10 \log \left(\frac{K_3 f^4}{\left(f^2 + f_2^2 \right)^2 \left(f^2 + f_3^2 \right)^2} \right) + W_C \ . \tag{4.4}$$

where f is the frequency (in hertz) to be weighted. and the constants are

$$f_1 = 20.598997. \qquad K_1 = 2.242881 \times 10^{16}.$$
$$f_2 = 107.65265. \qquad K_3 = 1.562339 \ .$$
$$f_3 = 737.86223.$$
$$f_4 = 12194.22,$$
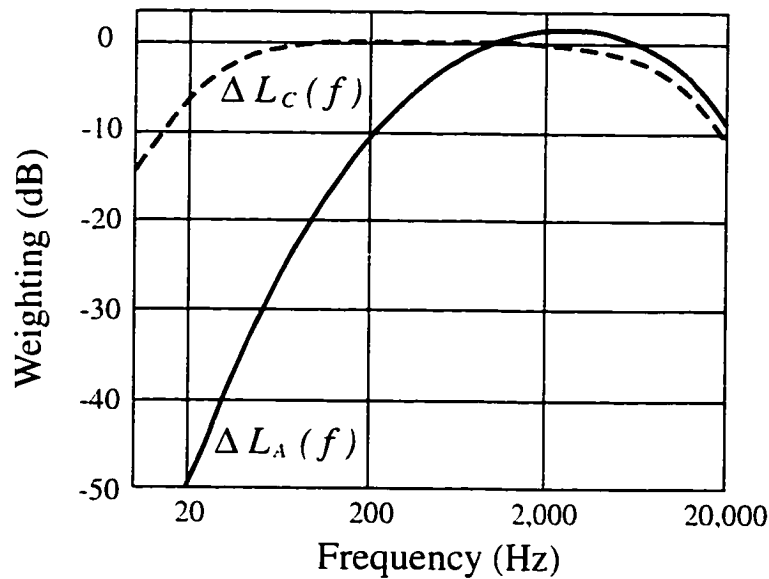
Figure 4.1: Relative response functions for A and C weightings. Plot adapted from Ref. 44.

4.3. Weighted Sound Levels of Evanescent Wave

Now that it is understood why it is important to determine underwater sound levels and how it is possible to quantify the physical and perceived sound. this section will provide a description of how calculations are made of sound levels of a decaying sonic boom waveform.

4.3.1. Calculating the sound levels

For this research, the sound descriptors applied are peak pressure level (peak dB). unweighted sound exposure level (L_{UE}), C-weighted sound exposure level (L_{CE}), A-weighted sound exposure level (L_{AE}). and Stevens' Mark VII perceived level (PLdB). All of these sound descriptors are calculated as a function of depth using a program written in *Mathematica*[73] and a Fortran program developed at the NASA Langley Research Center.[55] Functions for A- and C-weighted filters were developed in *Mathematica*: these functions are based on Eqs. (4.3) and (4.4). Also developed in *Mathematica* was the sound exposure level function which is based on Eq. (4.2). The Fortran program was used to calculate the perceived level of sound (PLdB).

All of the *Mathematica* functions can be found in Appendix A. In addition to the ones developed for this research. functions for generating the pressure data as a function of time. depth. and sonic boom parameters were utilized. These functions were those used by Sparrow;[58] they are based on Eq. (3.1). the equation developed by Sawyers[53] for the underwater evanescent pressure field.

The calculations proceeded as follows: the pressure waveform was generated (either as an N wave at the surface or the decaying waveform with depth). the peak pressure sound level was calculated. a Fourier transform was applied to enter the frequency domain. weighting filters were applied where necessary. an inverse Fourier transform was applied to get back to the time domain. and finally the sound exposure levels were calculated. Examples of this procedure can be found in Appendix A.

It should be noted that the pressure data were composed of 16.384 samples corresponding to a sampling frequency of approximately 10 kHz. The tails of the distorted (underwater) sonic boom waveform were linearly forced to zero to avoid enormous calculations which would result from using a window large enough to capture the entire waveform: please refer to Fig. 4.2. Also the reference pressure was 20 μPa.
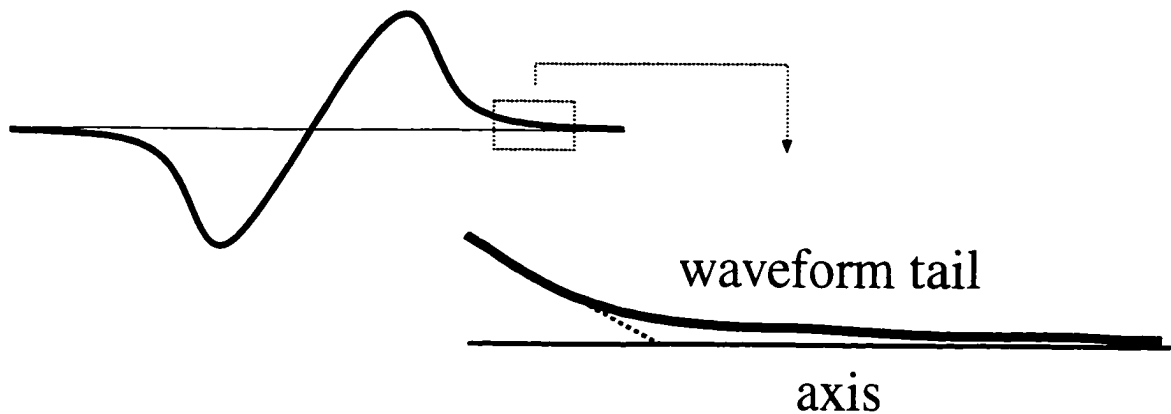
Figure 4.2: Cutting off the tails of the distorted (underwater) sonic boom waveform.

The A- and C-weighted sound exposure level functions were tested using one set of known sonic boom measurements where weighted sound exposure levels were stated. In

1989 Young[75] measured sonic booms from a flyover of the space shuttle Atlantis four minutes before its landing at Edwards Air Force Base. Sound exposure levels in the air were calculated using various instruments. Two A-weighted sound exposure levels L_{AE} were measured at 83.2 dB and 82.6 dB and two C-weighted sound exposure levels L_{CE} at 102.4 dB and 101.8 dB. Since these measurements included all energy from the sonic boom, 3 dB should be subtracted to account for the two single events of the long duration signature; this would yield values of 80.2 dB and 79.6 dB for A weighting and 99.4 dB and 98.8 dB for C weighting. After subtracting the 3 dB it is now possible to compare values to ones calculated using the *Mathematica* program. With N wave parameters approximated using a plot of the measured sonic boom waveform, L_{AE} was calculated to be 77.7 dB and L_{CE} to be 100.0 dB. The differences in magnitude of 1.9–2.5 dB for the A-weighted and 0.6–1.2 dB for the C-weighted sound exposure levels could be explained by the many approximations implemented and also a low sampling frequency. This comparison was performed simply to see if the values were somewhere in the range of one set of known sonic boom measurements; it was determined that, in fact, the numerically calculated values were sufficient.

4.3.2. Program runs

In order to generate normalized sound descriptor results, it was first necessary to pick realistic parameters as inputs to the programs. It was assumed that the hypothetical supersonic aircraft was traveling at Mach 2.4 and created an N-shaped sonic boom with a duration of 300 ms and a peak pressure of 50 Pa (approximately 1 lb/ft^2). Because of the great impedance mismatch between air and water, the pressure at the water surface was doubled creating a peak pressure of 100 Pa both just above and below the surface.

Using this information, the *Mathematica* and Fortran programs calculated the sound descriptors as decibel values for various depths. The Fortran program's option of adding a short rise time to the waveform was used in calculating the PLdB metric. The data in Table 4.1 represent the sound descriptors at the surface of the water (referenced to air and water) due to the sonic boom generated by the hypothetical supersonic aircraft. It is observed that the unweighted peak sound pressure level is the highest and L_{AE} is the lowest decibel value. It is also seen that the weighted sound descriptors, L_{CE}, L_{AE}, and PLdB, are all lower in decibel level than the unweighted sound exposure level, L_{UE}.

Table 4.1: Decibel levels at surface of water using hypothetical parameters.

depth (meters)	reference pressure	dB peak	L_{UE} (dB)	L_{CE} (dB)	L_{AE} (dB)	PLdB (dB)
0	20μPa	131.0	118.0	108.0	94.4	106.1
0	1μPa	157.0	144.0	134.0	120.4	132.1

After calculating the surface sound levels. values for the various sound levels at several different depths were examined. The same sound descriptors were applied to the sonic boom noise at 1. 2. 4. 8. 16. 32. 64. and 128 meters. The results for these depths were then normalized relative to those at the water's surface.

4.3.3. Results

The relative sound descriptors as a function of depth are presented in Table 4.2. It is seen that L_{CE}. L_{AE}. and PLdB all drop off more rapidly with depth than do the peak and unweighted levels. L_{AE} values decrease much more quickly with depth than L_{CE} or PLdB values. Examining the levels at a depth of 16 m. the peak and L_{UE} metrics have only decreased a few dB. which to humans is barely perceptible. The weighted levels of L_{CE}. L_{AE}. and PLdB have all decreased approximately 20 dB or more. which is an easily detectable change. Although the calculations of the sound level at different depths were made assuming a reference pressure of 20 μPa. the relative values in Table 4.2 are also valid if one instead uses any other reference pressure. such as 1 μPa.

A plot of all the different sound descriptors as a function of depth relative to the ocean surface is found in Fig. 4.3: values for this plot were taken directly from Table 4.2. It is obvious from these curves that the weighted noise descriptors show a much more drastic decrease in decibel level with depth than the unweighted descriptors. especially the A-weighted sound exposure level.

The results described here for the sound level work are also found in Rochat and Sparrow.[51]

Table 4.2: Decibel levels as a function of depth relative to those at the surface, found in Table 4.1.

depth (meters)	dB peak	L_{UE} (dB)	L_{CE} (dB)	L_{AE} (dB)	PLdB (dB)
0	0.0	0.0	0.0	0.0	0.0
1	-0.9	-0.3	-6.8	-18.6	-7.9
2	-1.4	-0.5	-8.6	-25.1	-11.7
4	-2.0	-0.9	-11.3	-33.3	-17.5
8	-3.0	-1.6	-15.1	-43.6	-24.7
16	-4.6	-2.7	-20.1	-56.2	-30.7
32	-7.2	-4.7	-26.1	-71.0	-35.0
64	-11.6	-8.1	-33.0	-84.8	-38.7
128	-18.4	-13.4	-42.0	-91.5	-42.4

4.3.4. Discussion of results

This section has examined several sound descriptors as a function of depth for an N-shaped sonic boom penetrating a flat ocean surface. The parameters used correspond to a hypothetical supersonic passenger aircraft. At the surface, the sound level re 20 μPa of a 50 Pa peak pressure, 300 ms duration sonic boom ranged from 94.4-131.0 dB (120.4-157.0 dB re 1 μPa), depending on the metric applied. At 128 meters under the surface, the sound level dropped off as much as 91.5 dB for the A-weighted sound exposure level, approximately 42.0 dB for the other weighted metrics, and 13.4-18.4 dB for the unweighted metrics. Since A weighting does not favor the lower frequencies, the decibel drop-off from the surface is quite drastic: the high frequencies diminished rapidly with depth. The other weighted metrics also decayed more rapidly than those unweighted, but not as rapidly as the A-weighted one since more of the lower frequencies are emphasized. This work has shown that the perceived noise levels may decrease rapidly with depth from those at the ocean surface, and the impact experienced may be significantly diminished compared with that predicted using unweighted levels. This result may be important in evaluating the possible sonic boom noise impact on marine mammals. Clearly the metrics used in this section were those appropriate for humans, and metrics should be developed specifically for pinnipeds
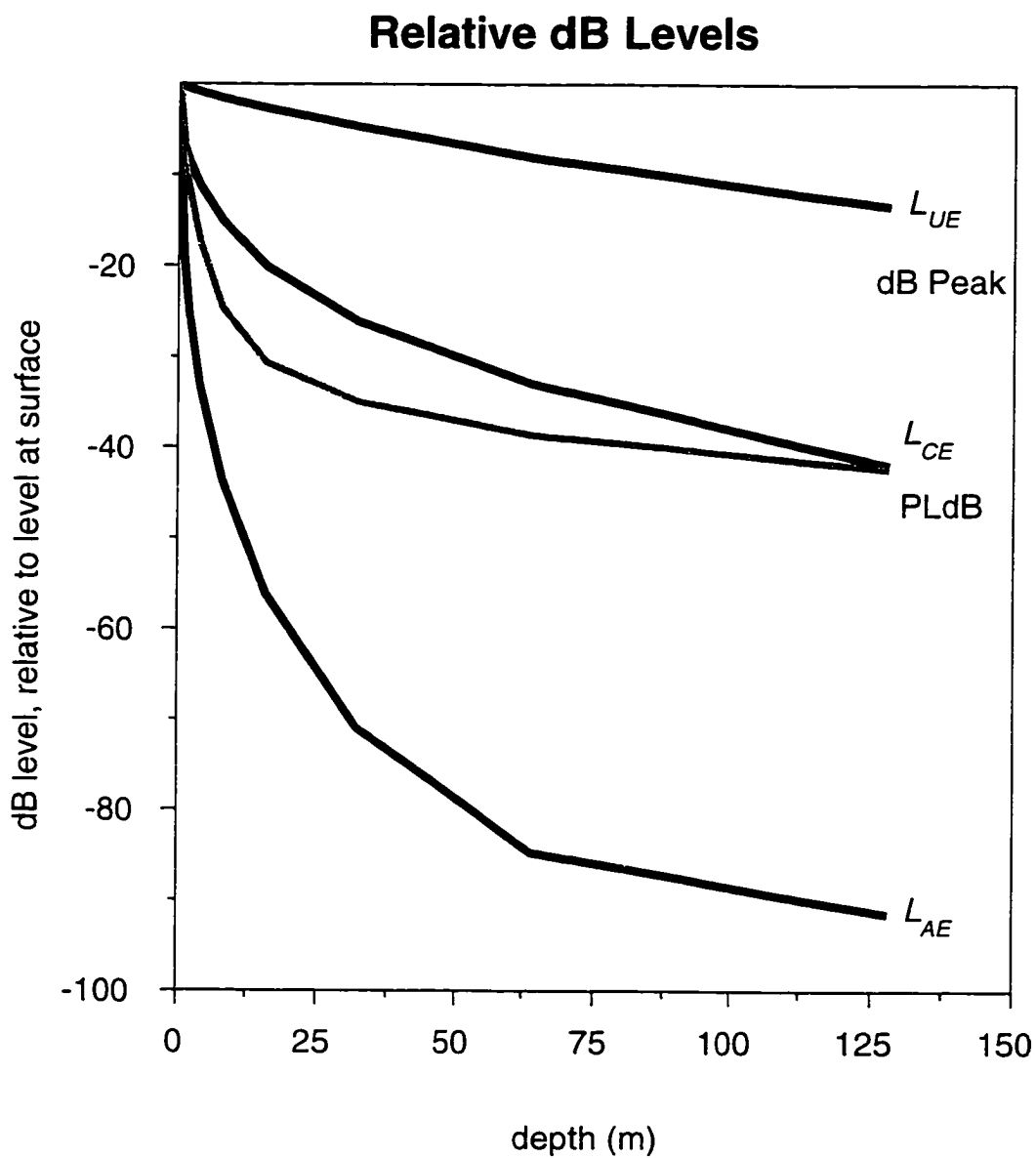
Relative dB Levels

Figure 4.3: Plot of decibel levels as a function of depth relative to those at the surface. Values of all sound descriptors are taken from Table 4.2.

and cetaceans. At the present time, however, the noise descriptors for humans provide us our only prediction of the potential annoyance experienced by marine mammals.

Chapter 5.
Computational Aspects

Chapter 4 described the underwater sound due to a sonic boom impinging upon a flat ocean surface. What now needs to be quantified is the underwater sound due to a sonic boom interacting with a more realistic ocean. one with a wavy surface or bubbles near the surface. Simple theory cannot be applied in these cases. as it does not yet exist.

It is necessary to choose a method for analysis. either analytical. experimental. or computational. to solve the problem at hand. Since the supersonic airplane motivating this research does not yet exist. it is important to have the flexibility of varying several parameters in order to find overall trends for the underwater sound field. It is obvious that an experimental solution is not yet possible for an actual HSCT. It is. however. possible to obtain data from other supersonic aircraft. such as the Concorde or military aircraft. but achieving all of the appropriate combinations of aircraft speed and ocean wave height or bubbles (discussed in detail later in this work) would be a costly task. If one were to derive an analytical solution. it would have to be general enough such that the aircraft and ocean parameters could all be input variables: this would be quite complicated. One method of analysis remains: computational. Here difficulties can arise because of the computational grid approximation of the ocean surface and because of the huge impedance change going from air to water. However. as long as the difficulties involved can be overcome and a stable and accurate code realized. a computational approach can analyze a sonic boom interacting with the ocean surface and its corresponding underwater sound field. including the ability to vary the parameters with ease. In the end. the method chosen for analysis in this research was computational.

5.1. Choosing a Computational Method

Today many accurate methods in computational aeroacoustics are available to solve a wide variety of physical problems. In recent years several of these methods have been successfully implemented.[28] Each method uses chosen schemes and boundary conditions and has advantages over others according to the problem at hand. In choosing a method, one must assess the current problem and limit the range of possibilities.

5.1.1. Key elements in choice of method

In computational work it is desired to find schemes which have minimal numerical dispersion and dissipation errors: since linear, lossless acoustic waves are nondispersive and nondissipative in their propagation, they should be modeled as such.[27] Dispersion is caused by numerical artifacts that selectively alter phase shifts among the component wavelengths at each time step:[15] it produces unwanted added ripples or small spurious waves. Dissipation is caused by an attenuation of part of the answer at each time step; the acoustic wave is dissipated due to truncated terms in the discretized wave equation. Generally, higher-order schemes are usually less dispersive and dissipative, but these schemes come with a price: the higher the order, the more computationally intensive it becomes. So it is a question of accuracy versus efficiency and what is best for a specific problem.

Besides the order of the scheme, there are three key elements in choosing a method: the mesh (structured or unstructured, fixed or adaptive), the space discretization (centered or upwind), and the time derivatives (explicit or implicit).[26] A brief description of each of these elements follows. A structured mesh allows several methods to easily run, while an unstructured mesh may be hard to incorporate into particular schemes; the mesh (grid) really has to be formed for a particular problem, possibly adding refinement to areas in the domain where intricacy is important. One of the choices for space discretization is centered difference methods: because they lack mechanisms to suppress high frequency spurious oscillations, they sometimes require added dissipation terms which have to be calibrated to each problem for optimum performance. For upwind methods, more arithmetic operations are required, making the program more time consuming; however, they are problem-independent

formulations in that they utilize a mechanism for preventing oscillations and instabilities. If artificial dissipation is easily added or not necessary, a centered difference method should be chosen. Choosing between explicit and implicit time derivatives is very problem dependent. Explicit methods are good for time-dependent/unsteady problems: these methods are simple to program and have a minimal cost per time step, but stability conditions severely limit the maximum allowable time step. Implicit methods generally satisfy time-independent/steady-state problems, although they can be applied to unsteady problems as well; these methods allow unlimited time steps (although not always producing accurate solutions) but at an increased computational cost and more laborious programming.

5.1.2. Tools for discretization

Three different tools can be utilized for the discretization of differential operators: finite difference, finite element, and finite volume.[25] Finite difference methods are the oldest methods applied to obtain numerical solutions of differential operators. They are based on properties of Taylor expansions and on the definition of derivatives. Although these are the simplest of the tools, they require a high degree of regularity in the computational grid (mesh). On these grids, the grid points are distributed along families of nonintersecting lines. Although past criticism has stated that finite difference methods are not widely accepted as a numerical approach to wave propagation problems, there are many current methods which successfully overcome the faults of their predecessors.[21] Many robust methods have become available, each being more or less applicable to a particular problem. It should be noted that a significant amount of documentation on acoustic propagation problems solved with computational analysis involves finite difference methods.

The next tool, the finite element method, has been traditionally used for structural analysis. Finite element approximation requires the definition of an integral formulation. In these methods the space domain is discretized by the subdivision of the continuum into elements of arbitrary shape and size, usually triangular or quadrilateral figures. Within each element points called nodes are positioned along the sides. Derivatives need to be determined at each one of these nodes; this requires heavy bookkeeping. An advantage of the finite element method is that it is flexible for use with unstructured grids.

The finite volume tool also requires an integral formulation: here the conservation laws in integral form are discretized. (The behavior of a physical system is completely determined by conservation laws: during the evolution of a fluid, a certain number of properties, such as mass, generalized momentum, and energy, are "conserved.") This method is easily adapted to any arbitrary, unstructured grid.

5.1.3. What to choose?

In addition to the material covered in this section, consideration should be given to whether or not the method is well-documented. Derivations and examples can be very useful when trying to formulate a unique scheme for a specific problem. Also, what should always be kept in mind is how reasonable the method is: simplicity is best for understanding and implementing the method, and efficiency is best for making computations affordable. One final choice to be made concerns the boundaries of the computational domain. Instead of listing the many possible boundary conditions here, the boundary conditions applied to this particular problem will be discussed in detail later in this chapter.

Hirsch[26] sums up well the process of choosing a computational method:

> A large variety of efficient methods are available but objective recommendations seem hardly possible since none of the approaches has come out with undisputed advantages over others. Hence, the final selection will most probably be a combination of subjective judgment, experience and personal non-rational choice.

5.2. Chosen Method

The present research involves a sonic boom propagating in air near the surface of the ocean: it then interacts with the air-water interface penetrating to the area of interest, the underwater sound field. A two-dimensional simulation is desired with the possibility of upgrading to three dimensions. Also refinement in the grid (making it nonuniform) becomes necessary when adding curvature to the ocean surface. With this information it is possible to choose an appropriate computational method.

Wave propagation is inherently time-dependent: an immediate decision is to apply an explicit method that is easily upgradeable to three dimensions. Also, due to the needed

refinement, a nonuniform grid is desirable. This grid can still remain quite structured, having only a small area of refinement in one direction of the grid. (Details will be given later regarding the refinement.) Finite differences are chosen because they are: straightforward: well documented for acoustic wave propagation: and can be modified to model both nonlinear propagation. appropriate for very loud sound booms, and inhomogeneous media, appropriate for turbulence in the air and bubbles in the water. Centered difference methods are chosen for the space discretization. The problem is initially solved without adding dissipation: it is found that the method causes no unwanted spurious oscillations which interfere with the solution. so the added dissipation is not necessary for this research.

Two different finite difference schemes are necessary for the chosen computational method. In the air and in the water a 2nd-order centered difference scheme is applied. Since the incident wave sees a huge impedance change when hitting the air-water interface. a conventional finite difference scheme goes unstable at this point: a scheme that can handle such an impedance change. one similar to a geophysics method. is applied right at the interface. The computational domain variables. two finite difference schemes. the boundary conditions. and a stability analysis will be discussed in the remaining sections of this chapter.

5.2.1. Computational domain variables

There are many different variables to consider in the current problem: these include computational variables (grid parameters) and physical variables (sonic boom parameters. ocean wave heights. bubble parameters. and media). Before discussing the method used on the computational domain. the domain variables will be given. The physical variables will be discussed as they are introduced in the following chapters.

The computational domain in this work is a rectangular grid. This grid physically represents a region with these dimensions: 799 meters in the horizontal direction (x direction) and 781 meters in the vertical direction (z direction). The 781 meters are divided into 340 m in the air and 441 m in the water. The number of grid points does not match the number of meters: there are 800 grid points in the x direction and in the z direction. This nonuniform grid has a refined region surrounding the air-water interface. Grid refinement is a necessity in areas of the domain having detailed features. A wavy ocean surface and

bubble layers are going to be added to the domain to represent realistic ocean features: it is known from the programming of these features that grid refinement is a necessary addition for the program's stability. In the refined region a grid block spans 1 meter in the x direction ($\Delta x = 1$ m) and 0.25 m in the z direction ($\Delta z = 0.25$ m); this refined region physically extends 3 m above and 3 m below the mid-line of any ocean interface. Elsewhere in the domain. a grid block spans 1 m in the x direction ($\Delta x = 1$ m) and also 1 m in the z direction ($\Delta z = 1$ m). An example of how the computational grid in the region of the air-water interface would look is illustrated in Fig. 5.1: this illustration does not represent the actual number of grid points contained in the refined region.
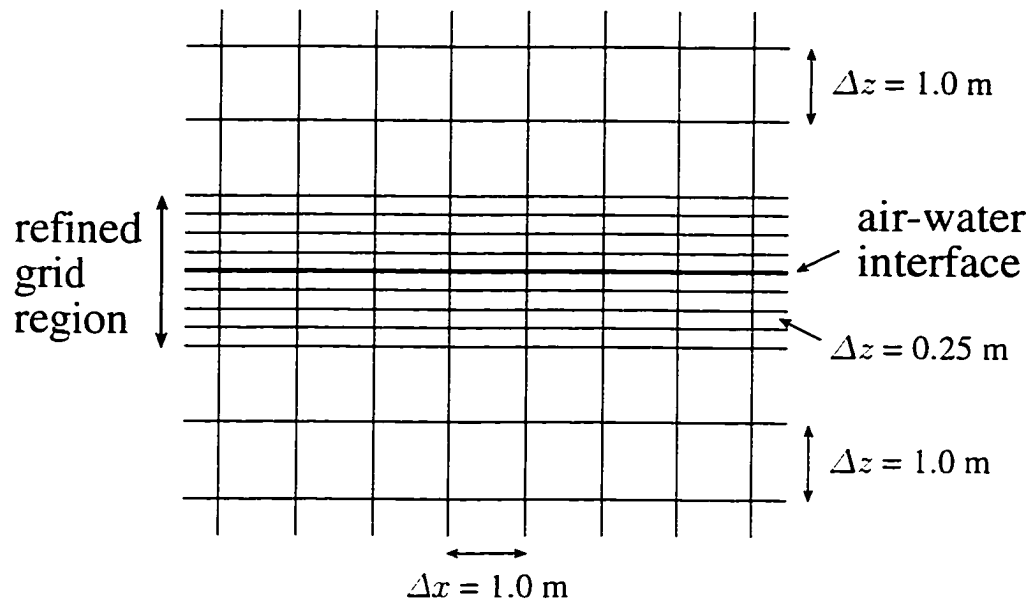
Figure 5.1: Illustration of computational grid near in the region of the air-water interface: this illustration does not represent the actual number of grid points contained in the refined region.

5.2.2. Air/Water scheme

In the air and water the method of wave propagation consists of 2nd-order centered difference schemes for both time and space. This method evaluates the acoustic wave equation directly using both basic finite difference approximations for a uniform grid and special finite difference approximations for a nonuniform grid. Although this is not a high-order method. it seems to give reliable results in an efficient manner. Issues regarding higher-order schemes are addressed later in Chapter 9.

5.2.2.1. Basic finite difference derivative

The basic finite difference method for calculating a derivative is based on the definition of a derivative.[25]

$$p_x = \frac{\partial p}{\partial x} = \lim_{\Delta x \to 0} \frac{p(x + \Delta x) - p(x)}{\Delta x} . \tag{5.1}$$

Here p is pressure. For computational grids it is obvious that Δx is not zero but rather a small finite value. and the equation must be written

$$p_x = \frac{\partial p}{\partial x} = \frac{p(x + \Delta x) - p(x)}{\Delta x} + \text{truncation error} . \tag{5.2}$$

The approximation can be improved by reducing Δx. The power of Δx with which this error tends to zero is called *the order of the difference approximation* and can be obtained from a Taylor series expansion of $p(x + \Delta x)$ around point x.

$$p(x + \Delta x) = p(x) + \Delta x p_x(x) + \frac{\Delta x^2}{2} p_{xx}(x) + \dots . \tag{5.3}$$

where p_{xx} is the second derivative. Equation (5.3) can be rewritten as

$$\frac{p(x + \Delta x) - p(x)}{\Delta x} = p_x(x) + \frac{\Delta x}{2} p_{xx}(x) + \dots \tag{5.4}$$

or

$$\frac{p(x + \Delta x) - p(x)}{\Delta x} = p_x(x) + \mathcal{O}(\Delta x) . \tag{5.5}$$

Here. $p_x(x)$ is expressed in first-order accuracy where $\mathcal{O}(\Delta x)$ is the truncation error.

5.2.2.2. Derivation of air/water finite difference scheme

The two-dimensional acoustic wave equation is

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2}\right) p - \frac{1}{c_o^2}\frac{\partial^2 p}{\partial t^2} = 0 \ . \tag{5.6}$$

where p is the acoustic pressure and c_o is the speed of sound in the medium. This is a *hyperbolic* differential equation. The physics implies that the solution at some time t is dependent only on past times; therefore, this equation can be solved using a time-marching scheme.[30] Two different derivations will be shown: the first is a centered finite difference approximation for a uniform grid for the terms $\frac{\partial^2 p}{\partial x^2}$ ($\equiv p_{xx}$) and $\frac{\partial^2 p}{\partial t^2}$ ($\equiv p_{tt}$), the term used for the time marching, and the second is a centered finite difference approximation for a nonuniform grid for the term $\frac{\partial^2 p}{\partial z^2}$ ($\equiv p_{zz}$). The uniform grid derivation will be given just in terms of x, not t, so as not to be repetitive.

The centered finite difference approximation for a *uniform* grid starts with two Taylor series expansions.

$$p(x + \Delta x) = p(x) + \Delta x p_x(x) + \frac{\Delta x^2}{2}p_{xx}(x) + \frac{\Delta x^3}{6}p_{xxx}(x) + \dots \ . \tag{5.7a}$$

$$p(x - \Delta x) = p(x) - \Delta x p_x(x) + \frac{\Delta x^2}{2}p_{xx}(x) - \frac{\Delta x^3}{6}p_{xxx}(x) + \dots \ . \tag{5.7b}$$

Adding together Eqs. (5.7a) and (5.7b) produces the equation

$$p(x + \Delta x) + p(x - \Delta x) = 2p(x) + \Delta x^2 p_{xx}(x) + \frac{\Delta x^4}{12}\frac{\partial^4 p(x)}{\partial x^4} + \dots \ . \tag{5.8}$$

Solving for p_{xx} in Eq. (5.8), the second derivative can then be written as

$$p_{xx} = \frac{p_{i+1} - 2p_i + p_{i-1}}{\Delta x^2} + \mathcal{O}\left(\Delta x^2\right) \ . \tag{5.9}$$

where the index i represents the grid point in the x direction ($p(x) \equiv p_i$ and $p(x \pm \Delta x) \equiv p_{i\pm 1}$), and the argument (x) has been suppressed.

The 2nd-order centered finite difference approximation for a *nonuniform* grid, derived using Hirsch,[25] starts with two Taylor series expansions.

$$p(z + \Delta z_{j+1}) = p(z) + \Delta z_{j+1}p_z(z) + \frac{\Delta z_{j+1}^2}{2}p_{zz}(z) + \frac{\Delta z_{j+1}^3}{6}p_{zzz}(z) + \dots \ . \tag{5.10a}$$

$$p(z - \Delta z_j) = p(z) - \Delta z_j p_z(z) + \frac{\Delta z_j^2}{2}p_{zz}(z) - \frac{\Delta z_j^3}{6}p_{zzz}(z) + \dots \ . \tag{5.10b}$$

As was explained earlier. Δz is grid-placement dependent and must be represented using the vertical grid index j. Multiplying Eq. (5.10a) by Δz_j and Eq. (5.10b) by Δz_{j+1} then adding the two equations together yields

$$\Delta z_j p_{j+1} + \Delta z_{j+1} p_{j-1} = (\Delta z_j + \Delta z_{j+1}) p_j + \left(\frac{\Delta z_j^2 \Delta z_{j+1}}{2} + \frac{\Delta z_j \Delta z_{j+1}^2}{2} \right) (p_{zz})_j$$
$$+ \left(\frac{\Delta z_j \Delta z_{j+1}^3}{6} - \frac{\Delta z_j^3 \Delta z_{j+1}}{6} \right) (p_{zzz})_j + \dots \quad (5.11)$$

Here. $p(z) \equiv p_j$. $p_{zz}(z) \equiv (p_{zz})_j$. $p_{zzz}(z) \equiv (p_{zzz})_j$, $p(z + \Delta z_{j+1}) \equiv p_{j+1}$. and $p(z - \Delta z_j) \equiv p_{j-1}$. Taking Eq. (5.11) and multiplying it by $\frac{2}{\Delta z_j \Delta z_{j+1}(\Delta z_j + \Delta z_{j+1})}$ then performing several steps of manipulation. an expression for the second derivative on a nonuniform grid becomes

$$(p_{zz})_j = \left[\frac{p_{j+1} - p_j}{\Delta z_{j+1}} - \frac{p_j - p_{j-1}}{\Delta z_j} \right] \frac{2}{\Delta z_{j+1} + \Delta z_j} - \frac{1}{3}(\Delta z_{j+1} - \Delta z_j)(p_{zzz})_t + \dots \quad (5.12)$$

As a check to see if Eq. (5.12) is correct. the grid for the p_{zz} derivative is made uniform. $\Delta z_j = \Delta z_{j+1} = \Delta z$. and Eq. (5.12) becomes equivalent to Eq. (5.9).

Now that approximations have been derived for all the derivatives. it is possible to reveal the finite difference scheme which represents the acoustic wave equation in the air and in the water:

$$p_{i,j}^{n+1} = 2p_{i,j}^n - p_{i,j}^{n-1} + c_o^2 (\Delta t)^2$$
$$\times \left[\frac{p_{i+1,j}^n - 2p_{i,j}^n + p_{i-1,j}^n}{(\Delta x)^2} + \left(\frac{p_{i,j+1}^n - p_{i,j}^n}{\Delta z_{j+1}} - \frac{p_{i,j}^n - p_{i,j-1}^n}{\Delta z_j} \right) \frac{2}{\Delta z_{j+1} + \Delta z_j} \right] . \quad (5.13)$$

where the index n represents time and the indices i and j. the grid points in the x and z directions. respectively. represent space. The variable p is the acoustic pressure. c_o is the speed of sound (different for each medium). Δt is the time increment. Δx is the spatial increment in the x direction (horizontal). and Δz is the spatial increment in the z direction (vertical). In general. $\Delta z_j = z_j - z_{j-1}$. z_j being the physical location at index j on the grid. As is indicated by Eq. (5.13). the computational grid must be uniform in the x direction but allows for nonuniformity in the z direction. This air/water finite difference scheme can be found in Appendix B: Appendix B contains all of the Fortran code for the two-dimensional 2nd-order simulations: the air/water finite difference scheme appears in the subroutine sochdiff2d.

This finite difference scheme requires a 3-point stencil for each of the spatial derivatives. In other words. information at three consecutive grid points (3 horizontal and 3 vertical)

will be used at every time step to calculate a derivative. Figure 5.2 represents this 3-point stencil. A 3-point stencil is typical for a 2nd-order centered derivative: more points are required for higher-order centered derivatives. The time derivative also involves three values: the pressure at the current time (what is being calculated) and the pressure at two previous times ($t - \Delta t$ and $t - 2\Delta t$).
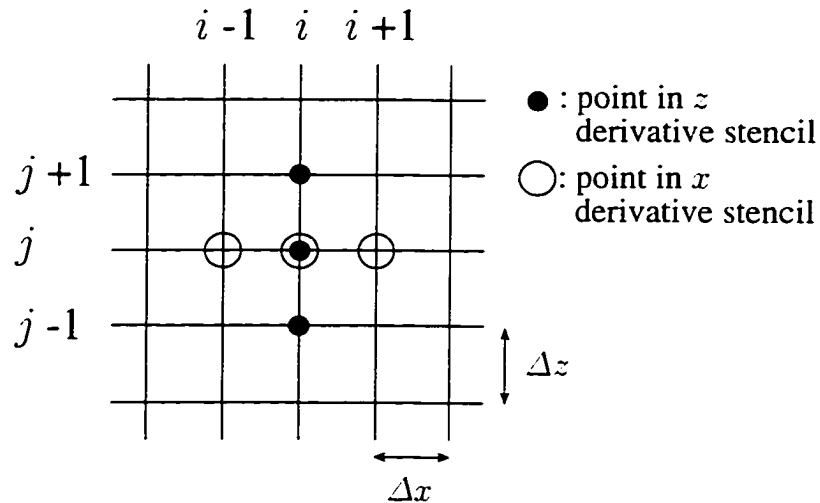
Figure 5.2: Illustration of a 3-point stencil for each spatial derivative on the computational domain.

It is purely the two-dimensional wave equation which is numerically represented. An attenuation term is not added for underwater propagation. Although this attenuation can be quantified in decibel loss over a distance, such as found in Jensen, et al.,[30] it is determined that this loss is insignificant. The attenuation is frequency dependent. The significant frequency content in a sonic boom is below 400 Hz (most significant being below 40 Hz), as was stated in Chapter 2: for these frequencies, the loss over 800 m is at most calculated to be 0.01 dB, an amount which does not affect the results which will be presented in this dissertation.

5.2.3. Interface scheme

Since the air-water interface denotes a huge impedance change. it is necessary to implement a scheme capable of handling the changes in speed of sound and density. Although finite differences can easily handle the sound speed change. a factor of 4.4 between air and water. most finite difference schemes will go unstable with the huge change in ambient density. a factor of approximately 826. For example. if used alone Eq. (5.13) will go unstable at the air-water interface.

To overcome this difficulty of instability. a finite difference method different than Eq. (5.13). similar to one used in seismology. is applied at all grid points within $\pm\frac{1}{2}\Delta z$ of the air-water interface. (If the interface does not intersect any grid blocks, the scheme is actually applied right on the interface.) The scheme is derived by integrating the acoustic wave equation across the interface: this ensures proper reflection and transmission coefficients. This method was originally applied by Sochacki. et al.[56] for acoustic and elastic wave propagation.

The derivation. which follows that in Refs. 56 and 59. begins with the inhomogeneous wave equation.

$$\frac{1}{\rho_o\left(x.z\right)c_o^2\left(x.z\right)}\frac{\partial^2 p}{\partial t^2} - \nabla\cdot\left(\frac{1}{\rho_o\left(x.z\right)}\nabla p\right) = 0 .\tag{5.14}$$

Notice that the density ρ_o and speed of sound c_o are dependent on the grid location. Rewritten. letting $a(x.z) = \frac{1}{\rho_o(x.z)c_o^2(x.z)}$ and $b(x.z) = \frac{1}{\rho_o(x.z)}$. Eq. (5.14) becomes

$$a\left(x.z\right)\frac{\partial^2 p}{\partial t^2} - \nabla\cdot\left(b\left(x.z\right)\nabla p\right) = 0 .\tag{5.15}$$

For simplicity. the remainder of the derivation will be for the one-dimensional wave equation: the two-dimensional formulation will be stated at the end of the section. In this one-dimensional derivation. Eq. (5.15) is written

$$ap_{tt} = \left(bp_z\right)_z = 0 .\tag{5.16}$$

with the arguments suppressed. Integrating this wave equation across the air-water interface yields

$$\int_{-\epsilon}^{\epsilon} ap_{tt}\,dz = \int_{-\epsilon}^{\epsilon} \left(bp_z\right)_z\,dz .\tag{5.17}$$

where $|\epsilon|$ is a small distance from the interface. The two terms in Eq. (5.17) can be written as

$$\int_{-\epsilon}^{\epsilon} a p_{tt}\, dz = a^- \int_{-\epsilon}^{0} p_{tt}\, dz + a^+ \int_{0}^{\epsilon} p_{tt}\, dz \tag{5.18}$$

and

$$\int_{-\epsilon}^{\epsilon} (bp_z)_z\, dz = bp_z|_{-\epsilon}^{\epsilon} = b\,(\epsilon)\, p_z\,(\epsilon, t) - b\,(-\epsilon)\, p_z\,(-\epsilon, t)$$
$$= b^+ p_z\,(\epsilon, t) - b^- p_z\,(-\epsilon, t) \ . \tag{5.19}$$

where a^+ and b^+ are calculated using the variables in the medium just above (ϵ above) the interface, air, and a^- and b^- using the variables in the medium just below (ϵ below) the interface, water.

Using the mean value theorem for integrals,

$$\int_{y_1}^{y_2} f\,(y')\, dy' = f\,(y)\,(y_2 - y_1). \qquad y_1 < y < y_2 \ . \tag{5.20}$$

Eq. (5.18) can be equated to

$$a^- p_{tt}\,(z^-, t)\,\epsilon + a^+ p_{tt}\,(z^+, t)\,\epsilon \ . \tag{5.21}$$

where $-\epsilon < z^- < 0$ and $0 < z^+ < \epsilon$. Letting $|\epsilon| \to \frac{1}{2}\Delta z$. and $z^+, z^- \to z$. the integrated wave equation can now be expressed as

$$\frac{\Delta z}{2} \left[a^- + a^+\right] p_{tt}\,(z, t) = b^+ p_z \left(\frac{\Delta z}{2}, t\right) - b^- p_z \left(-\frac{\Delta z}{2}, t\right) \ . \tag{5.22}$$

Using a centered finite difference approximation for p_{tt} and one-sided differences for p_z, the inhomogeneous wave equation becomes

$$\left(\frac{a^- + a^+}{2\Delta t^2}\right) \left(p_j^{n+1} - 2p_j^n + p_j^{n-1}\right) \Delta z = \frac{b^+}{\Delta z} \left(p_{j+1}^n - p_j^n\right) - \frac{b^-}{\Delta z} \left(p_j^n - p_{j-1}^n\right) \ . \tag{5.23}$$

and the one-dimensional 2nd-order finite difference scheme is

$$p_j^{n+1} = 2p_j^n - p_j^{n-1} + \frac{1}{a_j} \frac{\Delta t^2}{\Delta z^2} \left[b_{j+\frac{1}{2}} p_{j+1}^n - \left(b_{j+\frac{1}{2}} + b_{j-\frac{1}{2}}\right) p_j^n + b_{j-\frac{1}{2}} p_{j-1}^n\right] \tag{5.24}$$

where $a_j = \frac{a_{j-\frac{1}{2}} + a_{j-\frac{1}{2}}}{2}$.

Now it is possible to introduce the two-dimensional finite difference scheme. The finite difference scheme applied at the interface is

$$p_{i,j}^{n+1} = 2p_{i,j}^n - p_{i,j}^{n-1}$$
$$+ \frac{\Delta t^2}{a_{i,j}} \left\{ \frac{1}{\Delta x^2} \left[b_{i+\frac{1}{2},j} p_{i+1,j}^n - \left(b_{i+\frac{1}{2},j} + b_{i-\frac{1}{2},j}\right) p_{i,j}^n + b_{i-\frac{1}{2},j} p_{i-1,j}^n\right] \right.$$
$$\left. + \left[\frac{b_{i,j+\frac{1}{2}}\left(p_{i,j+1}^n - p_{i,j}^n\right)}{\Delta z_{j+1}} - \frac{b_{i,j-\frac{1}{2}}\left(p_{i,j}^n - p_{i,j-1}^n\right)}{\Delta z_j}\right] \frac{2}{\Delta z_{j+1} + \Delta z_j}\right\} \ , \tag{5.25}$$

where p, Δt, Δx, and Δz are the same as in Eq. (5.13). The additional variables are $a = 1/\rho_o c_o^2$ and $b = 1/\rho_o$, ρ_o being the ambient density for each medium. As previously stated, both a and b change depending on the location of the calculation. If the calculation is made in air, $c_o = 343$ m/s and $\rho_o = 1.21$ kg/m^3, and in water, $c_o = 1500$ m/s and $\rho_o = 1000$ kg/m^3. However, at the interface a and b are theoretically complicated[56] and can be difficult to implement, depending on whether or not the interface line intersects grid blocks or outlines them. Since the current interface will be grid block approximated (the interface line does not intersect a grid block), the functions can be written simply as $a = \frac{1}{2}(a_{air} + a_{water})$ and $b = \frac{1}{2}(b_{air} + b_{water})$. This interface finite difference scheme is also found in the Fortran code in Appendix B in the subroutine `sochdiff2d`.

It should be pointed out that, like the air/water scheme in Eq. (5.13), the interface scheme in Eq. (5.25) allows for a nonuniform grid in the z direction. Although the interface scheme is applied just at the interface, this allowance is incorporated just so that it has the flexibility to be used anywhere on the computational grid.

5.2.4. Boundary conditions

Since the computational simulations are calculated on a finite domain, it is necessary to update the grid points on the edges of the rectangle separate from the interior grid points: these points cannot be calculated using the finite difference schemes since these schemes require a 3-point stencil. Applying boundary conditions at the exterior grid points allows them to be updated for the next time step.

For the computational simulations in this work, hard reflecting boundaries are implemented wherever possible for simplicity. Hard reflecting boundaries cause the propagating wave to reenter the computational domain where it physically should have exited; this type of boundary condition should only be applied if the reflected waves do not interfere with the region of interest in the domain. For this research the initial waveform is inserted in the upper left-hand corner of the rectangle; the wave then propagates toward the lower right-hand corner. Reflections off the left, top, and right sides do not interfere with the simulations for the times of interest. Therefore, hard reflecting boundary conditions are applied on these sides. At a rigid boundary, the normal velocity is zero; the use of the linear version of Euler's equation, $\rho_o \frac{\partial \vec{v}}{\partial t} + \nabla p = 0$, implies that $\nabla p = 0$. At the edge of the

domain $\nabla = \frac{\partial}{\partial x}$ on the left and right sides, and $\nabla = \frac{\partial}{\partial z}$ on the top. As an example using x,

$$\frac{\partial p}{\partial x} = \frac{p(x + \Delta x) - p(x)}{\Delta x} = 0 . \tag{5.26}$$

which implies that $p(x + \Delta x) = p(x)$ at the outermost grid points. Figure 5.3 shows an example of where the theoretical rigid boundary lies, here on the left hand side of the computational domain.
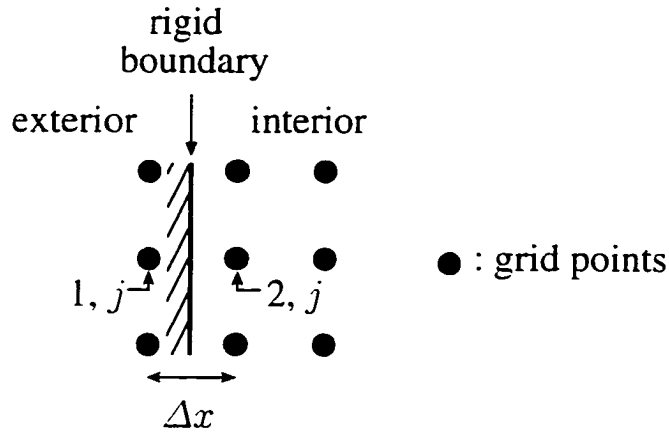
Figure 5.3: Illustration of rigid boundary on the left hand side of the computational domain.

Thus, the following boundary conditions are applied:

left side of grid	$p_{1,j} = p_{2,j}$	(5.27a)
top of grid	$p_{1,zmax} = p_{1,zmax-1}$	(5.27b)
right side of grid	$p_{xmax,j} = p_{xmax-1,j}$.	(5.27c)

where $xmax$ and $zmax$ are the total number of grid points in the x and z directions, respectively.

The bottom of the computational domain requires special consideration. The sound speed in water is 4.4 times that in air, and false reflections from a rigid bottom boundary would interfere with the present analysis of the sound levels at the interface. Two possible solutions to avoid this interference are: first, to extend the domain by a factor of 4.4 below the water surface or second, to implement an absorbing boundary on the bottom of the domain. The latter was chosen in order to make the program more efficient.

The absorbing boundary condition applied at the bottom of the domain is one taken from Sparrow and Raspet.[61] The actual boundary condition from Ref. 61 used in the program is labeled the Bayliss-Turkel B_1 condition.

$$\frac{\partial p}{\partial t} = -c_o \left(\frac{1}{r} + \frac{\partial}{\partial r} \right) p \ .$$ (5.28)

which is the outgoing part of the spherical wave equation. The goal is to let this outgoing part go to zero. For the Cartesian coordinates used in this research. the boundary condition is written

$$\frac{\partial p}{\partial t} = -c_o \left(\frac{1}{\sqrt{x^2 + z^2}} + \cos\phi \frac{\partial}{\partial z} + \sin\phi \frac{\partial}{\partial x} \right) p \ .$$ (5.29)

The discretized version of Eq. (5.29) takes the form

$$p_{i,j}^{n+1} = \left(1 - \frac{c_o \Delta t}{\sqrt{x^2 + z^2}} \right) p_{i,j}^n - \frac{c_o \Delta t}{\Delta z} \left(\cos\phi + \frac{\sin^2\phi}{\cos\phi} \right) \left(p_{i+1,j}^n - p_{i,j}^n \right) \ .$$ (5.30)

where x. z. and ϕ can be seen in Fig. 5.4. The bottom boundary condition can then be implemented as $p_{i,1}$ equated to Eq. (5.30).
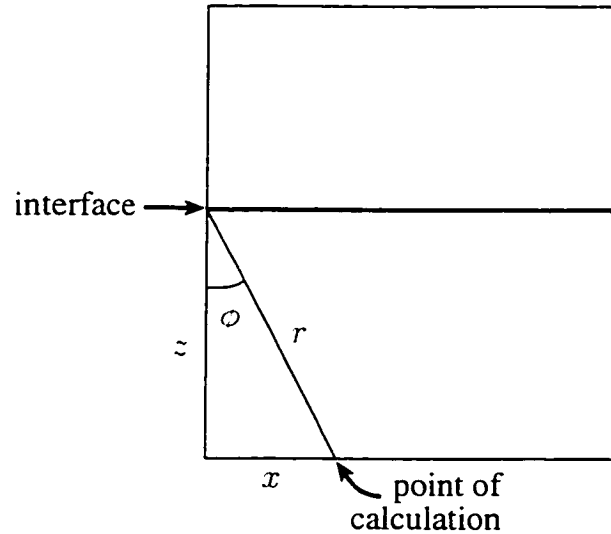
Figure 5.4: Illustration of variables used to calculate the bottom boundary condition.

All boundary conditions for the two-dimensional simulation are found in the Fortran code in Appendix B.

5.2.5. Stability of total finite difference method

The accuracy of a finite difference approximation is associated with two different errors, the *discretization error* and the *rounding error*.[30] The first error is introduced by the grid spacing; the exact solution p_D to the discretized equation is an approximation to the exact solution of the original differential equation for p. The associated error is called the discretization error ε_D ($= p - p_D$). The second error is introduced by round-off errors in the numerical solution; the difference between the exact solution to the discretized equation p_D and the numerical solution p_N is called the rounding error ε_R ($= p_D - p_N$) (can be called stability error). The total error of the finite difference solution is then the sum of these two errors, $\varepsilon = \varepsilon_D + \varepsilon_R$. If the total error ε is controlled the finite difference formulation can remain stable and sufficiently simulate actual wave propagation.

For a finite difference solution to be numerically stable, the time step Δt must be smaller than the time it takes an acoustic wave to traverse one grid cell in the computational grid.[30] The finite difference approximation of the acoustic wave equation is *conditionally stable*: if certain stability requirements are satisfied the formulation is stable, otherwise it is unstable. The condition under which a scheme is stable is known as the *Courant-Friedrichs-Lewy* or *CFL* condition.[25] For example, the CFL condition for a one-dimensional finite difference formulation of the wave equation is

$$\frac{c_o \Delta t}{\Delta x} \leq 1 . \tag{5.31}$$

where c_o, Δt, and Δx are the same as previously defined. The ratio $\frac{c_o \Delta t}{\Delta x}$ is labeled the *Courant number*.

Since, for this work, all of the computational domain except for the area representing the air-water interface uses the air/water finite difference scheme, Eq. (5.13), it is appropriate to find the stability condition for this scheme. This stability condition will be applied throughout the computational simulations. A classical method for determining the stability of a finite difference scheme is a *Von Neumann stability analysis*; this method of analysis was developed during World War II by Von Neumann and is based on a Fourier representation of the error distribution.[25,30] A Von Neumann stability analysis will now be performed on Eq. (5.13). This analysis uses Refs. 1, 25, 30, and 50 as guides.

Because every time step in a simulation involves calculations including all computational domain parameters. the CFL number must include the most extreme case variables to insure stability. For this reason c_o is set to be 1500 m/s and $\Delta z = 0.25$ m. Since only one Δz is being applied in the analysis. Eq. (5.13) can be written

$$p_{i,j}^{n+1} = 2p_{i,j}^n - p_{i,j}^{n-1} + \alpha_x \left(p_{i+1,j}^n - 2p_{i,j}^n + p_{i-1,j}^n \right) + \alpha_z \left(p_{i,j+1}^n - 2p_{i,j}^n + p_{i,j-1}^n \right) . \quad (5.32)$$

where

$$\alpha_x = \left(\frac{c_o \Delta t}{\Delta x} \right)^2 \qquad \alpha_z = \left(\frac{c_o \Delta t}{\Delta z} \right)^2 . \quad (5.33)$$

An arbitrary error distribution can be represented by the Fourier series

$$\varepsilon_{mn} \left(x. z. t \right) = \sum_{m=0}^{xmax} \sum_{n=0}^{zmax} A_{mn} \left(t \right) e^{ik_r x} e^{ik_z z} . \quad (5.34)$$

where k_x and k_z are the wavenumbers in the x and z directions. respectively. Since the error amplitudes A_{mn} are arbitrary. each term must propagate in time with decreasing amplitude for the solution to be stable: in order to demonstrate this stability. it is sufficient to examine individually the propagation of each term in the expansion. Upon substituting an individual error term into the finite difference formulation. Eq. (5.32). it is possible to determine the CFL condition. Letting the error amplitude A equal e^{lt}. we seek a solution of the form

$$\varepsilon \left(x. z. t \right) = e^{lt} e^{ik_r x} e^{ik_z z} . \quad (5.35)$$

Substituting Eq. (5.35) into Eq. (5.32) gives

$$e^{l(t+\Delta t)} e^{ik_r x} e^{ik_z z} = 2e^{lt} e^{ik_r x} e^{ik_z z} - e^{l(t-\Delta t)} e^{ik_r x} e^{ik_z z}$$
$$+ \alpha_x \left(e^{lt} e^{ik_r (x+\Delta x)} e^{ik_z z} - 2e^{lt} e^{ik_r x} e^{ik_z z} + e^{lt} e^{ik_r (x-\Delta x)} e^{ik_z z} \right) \quad (5.36)$$
$$+ \alpha_z \left(e^{lt} e^{ik_r x} e^{ik_z (z+\Delta z)} - 2e^{lt} e^{ik_r x} e^{ik_z z} + e^{lt} e^{ik_r x} e^{ik_z (z-\Delta z)} \right) .$$

Dividing Eq. (5.36) by $e^{lt} e^{ik_r x} e^{ik_z z}$ then yields

$$e^{l\Delta t} = 2 - e^{-l\Delta t} + 2\alpha_x \left(\frac{e^{ik_r \Delta x} + e^{-ik_r \Delta x}}{2} - 1 \right) + 2\alpha_z \left(\frac{e^{ik_z \Delta z} + e^{-ik_z \Delta z}}{2} - 1 \right) . \quad (5.37)$$

Applying the trigonometric formula $\cos \phi = \frac{e^{i\phi} + e^{-i\phi}}{2}$ to Eq. (5.37) gives

$$e^{l\Delta t} = 2 - e^{-l\Delta t} + 2\alpha_x \left(\cos k_x \Delta x - 1 \right) + 2\alpha_z \left(\cos k_z \Delta z - 1 \right) . \quad (5.38)$$

Letting $G = e^{I\lambda t}$, multiplying Eq. (5.38) by G, and applying the trigonometric formula $\sin^2 \phi = \frac{1}{2}(1 - \cos 2\phi)$ yields the quadratic equation

$$G^2 + \gamma G + 1 = 0 . \tag{5.39}$$

where G is the *amplification factor*. and

$$\gamma = -2 + 4\alpha_r \sin^2 \left(\frac{k_x \Delta x}{2}\right) + 4\alpha_z \sin^2 \left(\frac{k_z \Delta z}{2}\right) . \tag{5.40}$$

The solution to Eq. (5.39) is

$$G = -\frac{1}{2}\gamma \pm \frac{1}{2}\left(\gamma^2 - 4\right)^{\frac{1}{2}} . \tag{5.41}$$

Stability is assured if both roots satisfy the condition $|G| \leq 1$. where the amplitude of any error harmonic does not grow in time. Using the positive root. this leads to

$$-\frac{1}{2}\gamma + \frac{1}{2}\left(\gamma^2 - 4\right)^{\frac{1}{2}} \leq 1 \qquad \longrightarrow \qquad \gamma \leq 2 . \tag{5.42}$$

which. after substituting in Eq. (5.40). becomes

$$\alpha_r \sin^2 \left(\frac{k_x \Delta x}{2}\right) + \alpha_z \sin^2 \left(\frac{k_z \Delta z}{2}\right) \leq 1 . \tag{5.43}$$

Since the maximum value of $\sin^2(\frac{k_x \Delta x}{2})$ or $\sin^2(\frac{k_z \Delta z}{2})$ is 1. it is implied that $\alpha_r + \alpha_z \leq 1$. This will be satisfied if

$$\alpha_r \leq \frac{1}{2} \qquad \text{and} \qquad \alpha_z \leq \frac{1}{2} \tag{5.44}$$

which become the CFL conditions

$$\left(\frac{c_o \Delta t}{\Delta x}\right)^2 \leq \frac{1}{2} \qquad \text{and} \qquad \left(\frac{c_o \Delta t}{\Delta z}\right)^2 \leq \frac{1}{2} \tag{5.45}$$

that yield

$$\Delta t \leq \frac{1}{\sqrt{2}}\frac{\Delta x}{c_o} \qquad \text{and} \qquad \Delta t \leq \frac{1}{\sqrt{2}}\frac{\Delta z}{c_o} . \tag{5.46}$$

It should be noted that Sochacki. et al.[56] give the same CFL condition as Eq. (5.45) for their finite difference scheme; therefore. even though Eq. (5.45) was derived from the air/water finite difference scheme. it also applies to the interface scheme.

As previously stated. in order to calculate a Δt which would allow stable calculations over the entire computational domain. the smallest spatial increment and largest speed of sound must be employed. The calculated time increment in seconds is

$$\Delta t = \frac{1}{\sqrt{2}}\frac{\Delta z}{c_o} = \frac{1}{\sqrt{2}}\frac{0.25}{1500} \approx 1.1785 \times 10^{-4} . \tag{5.47}$$

This time increment is calculated and used in the two-dimensional simulations: again. the Fortran code can be found in Appendix B.

Chapter 6.
Homogeneous Ocean with Flat Surface: Computational Simulations

A stable finite difference code has been developed which has the ability to propagate a wave going from air into water. Before adding realistic features to the ocean, the computer program is run for the case of a homogeneous ocean with a flat surface; these are the conditions under which previous studies have been done. Through a set of runs, it is possible to make comparisons to the results of Cook,[12] Sawyers,[53] and Sparrow,[58] all their studies concerning a sonic boom interacting with a flat homogeneous ocean.

6.1. Computer Program

As mentioned in Chapter 5, there are many different variables to consider when implementing the wave propagation simulations; these include the computational variables and the physical variables. The computational variables have already been stated: grid parameters were assigned specific values, and the boundaries of the grid have been defined. The physical parameters—sonic boom parameters and ocean parameters—are chosen according to the desired simulation and are inputs to the computer program. Refer to Appendix B for an example of the input file.

6.1.1. Physical parameters: sonic boom

The waveform which propagates in these simulations is a rounded N-shaped sonic boom. As was explained in Chapter 2, realistic sonic booms vary in shape; the rounded N-shaped waveform was chosen for its lack of sharp discontinuities that a perfect N wave would possess. A perfect N wave would require a computationally intensive simulation, possibly using a very fine grid. The issue of a more N-like waveform being used in the simulations is addressed in Chapter 9.

The rounded sonic boom is calculated by multiplying a straight-line function by an appropriate window. The straight-line function is simply

$$\text{line}\,(x.\,z) = p_{peak}\frac{\zeta}{c\tau} \, . \tag{6.1}$$

where

$$\zeta\,(x.\,z) = z - \left\{ \frac{\sin\left[\sin^{-1}\left(\frac{1}{M}\right)\right]}{\cos\left[\sin^{-1}\left(\frac{1}{M}\right)\right]} \right\} x \, . \tag{6.2}$$

The variable ζ (appears as const in Fortran code) is the physical position on the grid; p_{peak} is the peak pressure of the sonic boom. c is the speed of sound in air. τ is half the duration of the sonic boom. and M is the Mach number. The multiplier is a Hanning window. also seen in the Appendix B Fortran code as the function w.

$$\text{window}\,(x.\,z) = \begin{cases} 0.5 + 0.5\cos\frac{\pi\zeta}{c\tau} & |\zeta| \le c\tau \\ 0 & |\zeta| > c\tau \end{cases} \, . \tag{6.3}$$

The initial pressure p_{init} is then

$$p_{init}\,(x.\,z) = 1.5\,(\text{line}) \times 2.5\,(\text{window}) \, . \tag{6.4}$$

The initial pressure is calculated at two times since the time derivative used in the finite difference schemes requires a 3-point stencil. In the simulation code in Appendix B these initial pressures appear as p ($p_{i,j}^{n}$) and ptmp2 ($p_{i,j}^{n-1}$).

The straight line. Hanning window. and the resulting rounded N wave appear in Fig. 6.1.
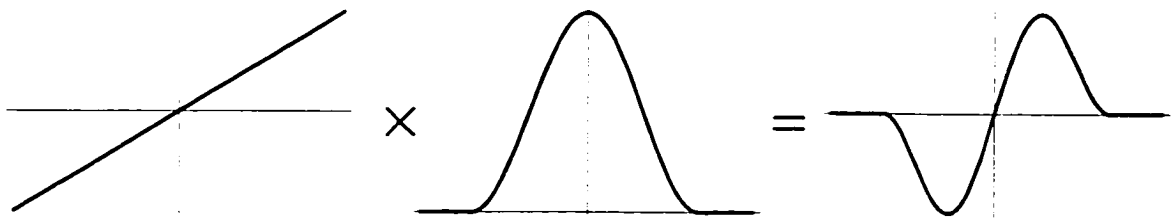
Figure 6.1: The rounded sonic boom waveform and the functions used to construct it. These are plots of pressure as a function of space.

The sonic boom has a specified peak pressure. duration. and angle of incidence which. as previously mentioned. is a function of the Mach number. These values are specified for each run.

6.1.2. Physical parameters: ocean

Two ocean parameters are options in the simulations: the ocean surface profile and the ocean (in)homogeneity. For the ocean surface profile. the choices are flat or curved, the curvature discussed in detail in the following chapter. The flat surface profile is chosen for this chapter's simulations. The ocean can contain ocean bubble layers near the surface: this is not an option chosen for this chapter's simulations. but will be discussed in detail in Chapter 8.

6.1.3. Initial conditions

A rounded N-shaped sonic boom waveform is inserted in the upper left corner of the computational domain. This sonic boom then propagates toward the lower right corner: during its progression it is interrupted by the air-water interface. For all the simulations in this chapter. the peak pressure of the sonic boom is set to be 50 Pa and the duration to be 300 ms. These values correspond to a hypothetical HSCT[67] in steady. level flight (peak pressure chosen from typical sonic booms under these conditions[23]). Several different Mach numbers are used to find trends in the simulations with a flat homogeneous ocean.

Figure 6.2 shows the pressure field for the initial condition for two cases. Mach 1.4 and Mach 2.4: here the dark band is the positive pressure in the waveform. and the light band is the negative pressure. The text above the graphics will be explained in Sect. 6.3. Part (a) of Fig. 6.2 shows the full computational domain for the Mach 1.4 case and part (b) of Fig. 6.2 shows the full domain for the Mach 2.4 case. The white horizontal line represents the air-water interface: the vertical axis. the z direction. is divided by this interface. 340 meters in the air. the rest in water. The horizontal axis is simply the distance in meters in the x direction. Also shown. in Fig. 6.3 (the next figure). is a horizontal slice of the pressure field for the Mach 1.4 case. part (a) of Fig. 6.2: the rounded sonic boom waveform is seen to have a peak pressure of approximately 50 Pa.

pressure; M=1.4, wh=0.0 m; #0

pressure; M=2.4, wh=0.0 m; #0

a)   pressure (Pa)

b)   pressure (Pa)
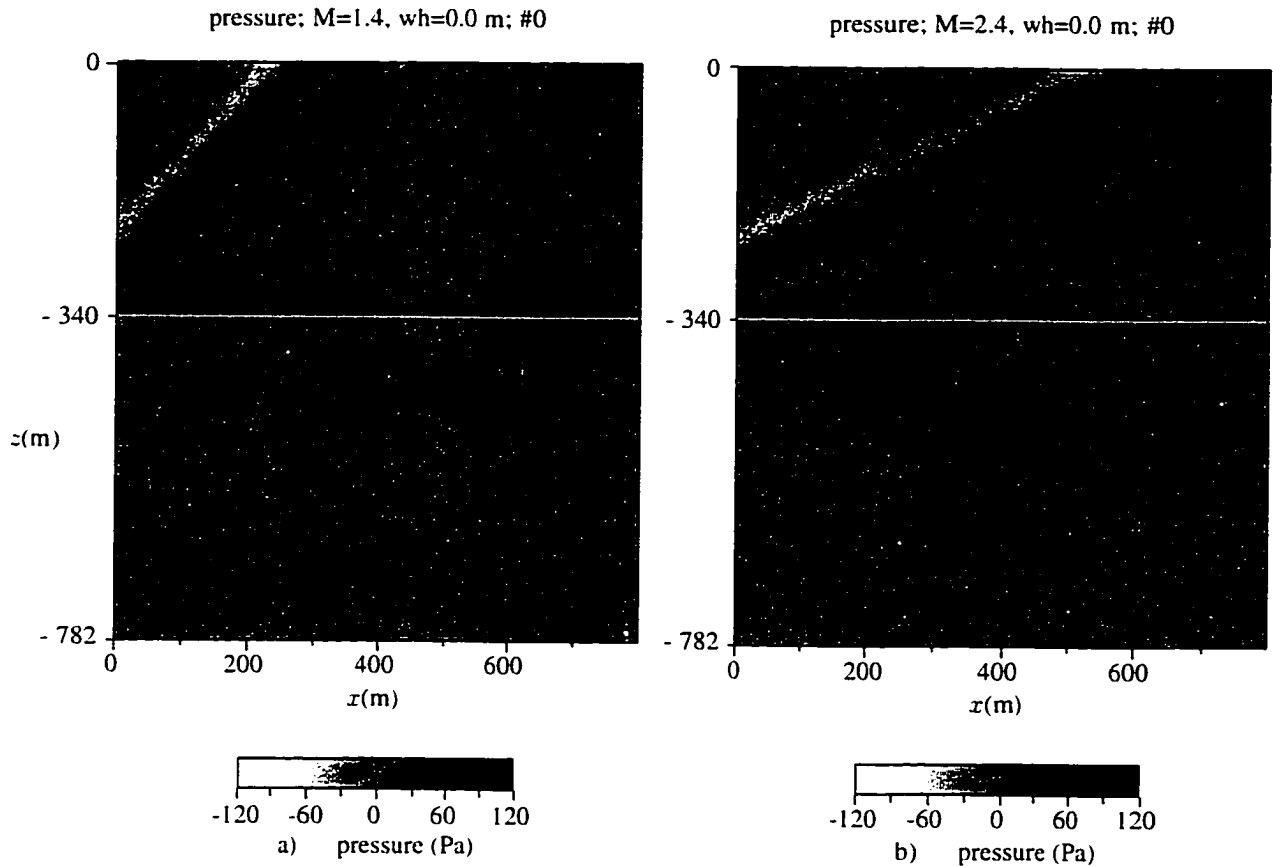
Figure 6.2: Initial pulse in full computational domain, wave height 0.0 m; (a) Mach 1.4; (b) Mach 2.4.

6.2. Program Runs

All runs of the simulation program used for this chapter and the remaining chapters were made on a Digital Equipment Corp. Alpha 3000/600 workstation, and each simulation took approximately 2.5 to 10 hours of run time (for the program in its final form), depending on the user load and the accuracy of the code. The user part of the CPU time (CPU time used by the process outside of system calls) for each 2nd-order accurate simulation takes on average about 2:35:47 (hours:minutes:seconds).

For the calculations in this chapter only one parameter is varied, the Mach number, since the peak pressure and duration of the sonic boom have already been set, and the ocean is homogeneous (no bubbles) with a flat surface (ocean wave height = 0.0 m). The Mach
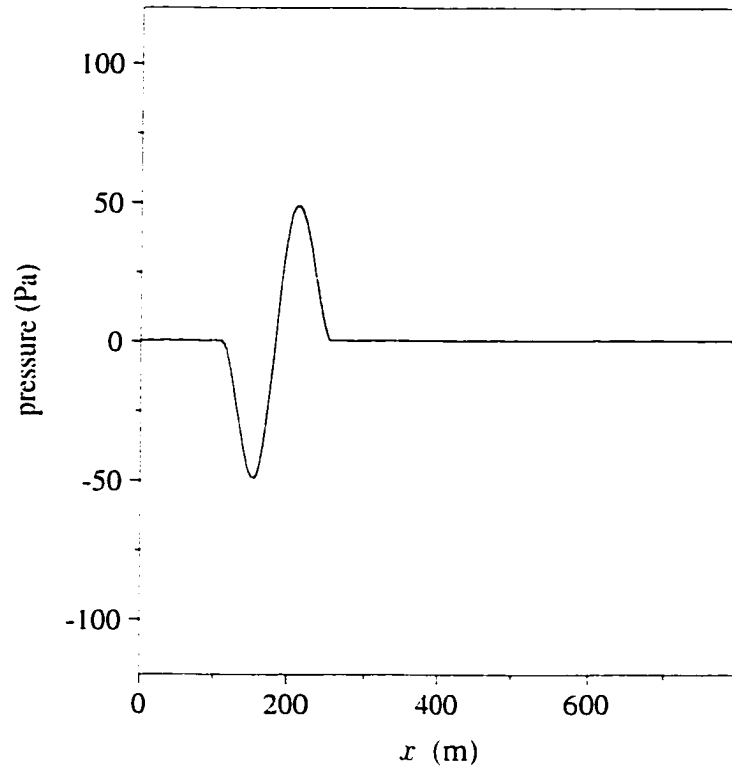
Figure 6.3: Initial pulse. horizontal slice of Mach 1.4 pressure field.

numbers used in these calculations are 1.4, 2.4. 3.0. 3.5. and 4.5. the order corresponding to decreasing angle of incidence.

6.3. Full-Field Pressure Results

An example of the pressure field over the entire computational domain at a time when the sonic boom is interacting with the ocean surface is seen in Fig. 6.4; this snapshot in time is taken from the Mach 2.4 simulation. The text on top of the graphic identifies the particular run: this will appear in all full domain plots in this dissertation. Here wh stands for wave height. and the number following the symbol # identifies the snapshot. During the calculations. the values for the full domain plot are written out every 40 time steps, and, as was previously described in Chapter 5, Sect. 5.2.5, $\Delta t \approx 1.1785 \times 10^{-4}$. Using Fig. 6.4 as an example, #160 $\Rightarrow t = \Delta t \times 40 \times 160 = 0.75$ seconds.

It should be noted that when viewing Fig. 6.4 and the remaining full-domain graphics in this dissertation, the plot is stretched in the region corresponding to the refined grid region in the computational domain. As previously mentioned, the refined region extends 3 m both above and below the mid-line of the air-water interface. The figures show every grid point as a pixel, so the 6 m of the refined region is shown as 24 m in height. In other words, the unrefined 1 m height computational grid blocks still correspond to 1 m in the figures, but the refined 0.25 m height grid blocks now also correspond to 1 m. Also, the graphics show dashed white lines which outline the refined grid region and a black box which outlines a region of the full computational domain from where all results will be extracted: the box would vary slightly with each different Mach number simulation, but the one shown gives a good example of where the significant information resides. The part of the domain outside the box is obviously affected by the chosen initial and boundary conditions, and the results there are not usable.

In Fig. 6.4 the sonic boom is shown to not only reflect from the surface of the ocean but also to penetrate into the ocean as an evanescent wave. Admittedly, it is difficult to look at this figure and say for sure that the underwater pressure field represents an evanescent wave, but it can be assumed that a sonic boom propagating into the ocean would look more like the incident and reflected waves and not what is seen under water here.

Many palettes are available to view the numerical data. Later in this dissertation a blue/red/yellow palette will be introduced (red for positive pressure, blue for negative pressure, and yellow for high amplitude pressure); it will be apparent then why that color palette is advantageous, but for this chapter a gray scale palette is sufficient. For the computational simulations, the inverted gray scale is preferred because of the clear, nondistracting display of the pressure field; however, to have a better understanding of the extent of the underwater pressure field, a banded gray scale is applied in this section of the chapter. As previously seen, the Fig. 6.4 graphic applies an inverted gray scale palette. Now that same graphic will be displayed using a banded gray scale palette; refer to Fig. 6.5. Here the evanescent field is somewhat more clear than in Fig. 6.4, bands of gray forming semi-circular shapes descending from each peak of the sonic boom waveform. To more fully visualize the underwater sound field it would also be highly beneficial to see the intensity field.

pressure; M=2.4; wh=0.0 m; #160
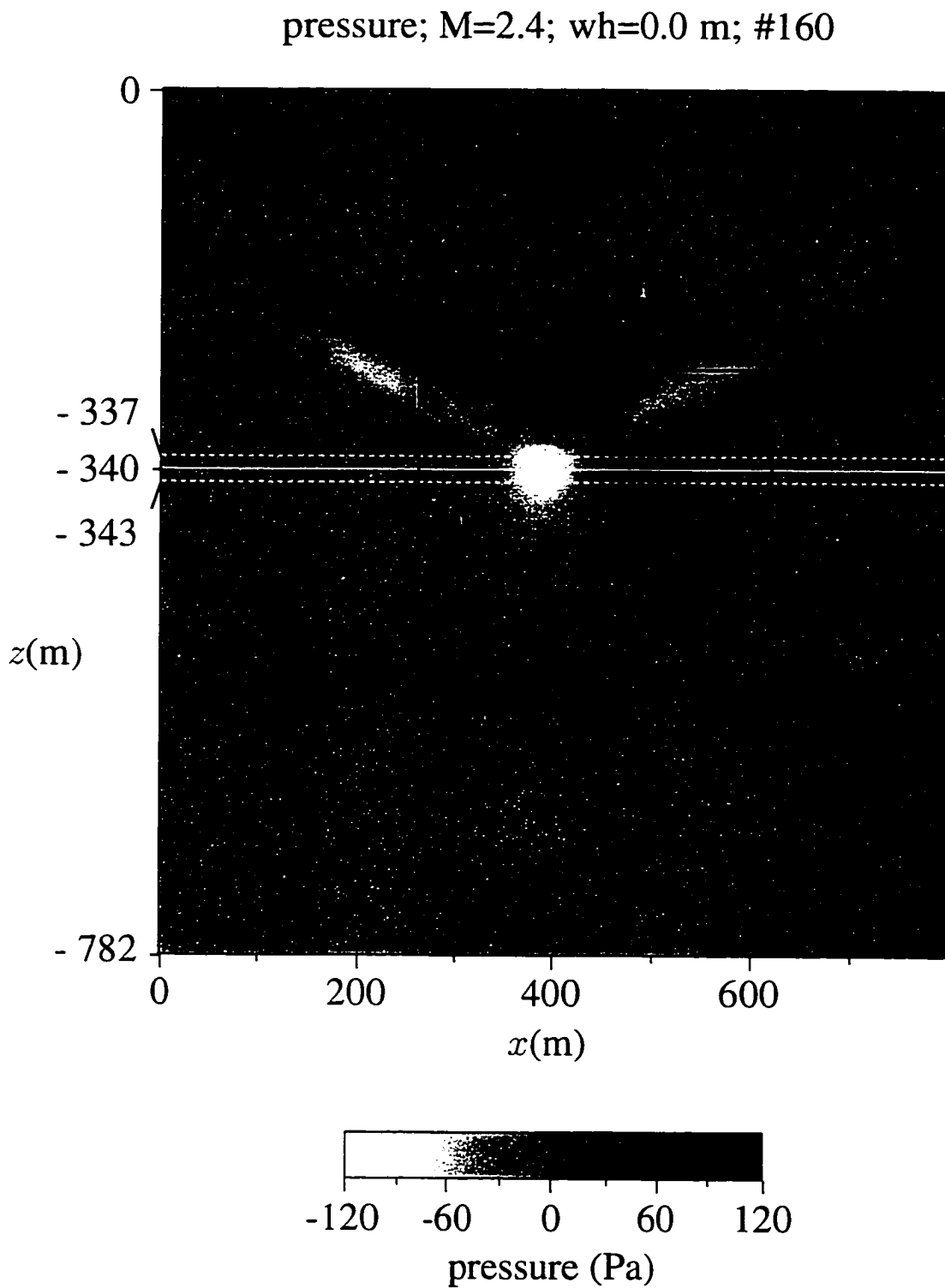
Figure 6.4: Mach number 2.4, wave height 0.0 m; incident wave interacting with the air-water interface. (Inverted gray scale)

pressure; M=2.4; wh=0.0 m; #160
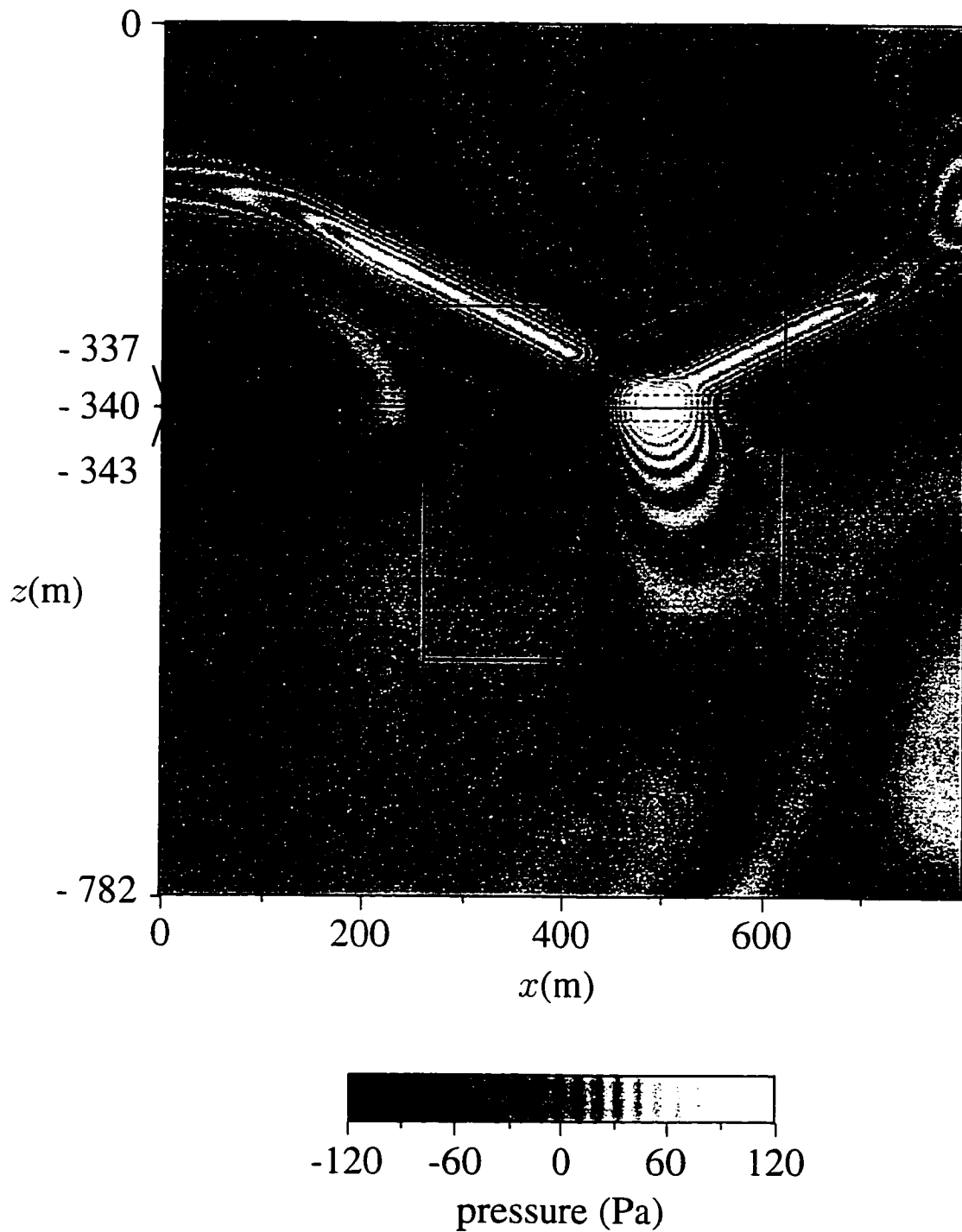
Figure 6.5: Mach number 2.4, wave height 0.0 m: incident wave interacting with the air-water interface. (Banded gray scale)

6.4. Incorporating Acoustic Intensity

6.4.1. Equations used to calculate intensity

Calculating the acoustic intensity in addition to the pressure enables an accurate interpretation of the computational simulation results. In order to visualize the power flow through a point the *instantaneous intensity* should be calculated. The instantaneous intensity $\vec{I}$ which is a function of time is expressed as

$$\vec{I}(r.t) = \text{Re}\{p(r.t)\} \cdot \text{Re}\{\vec{v}(r.t)\} \ . \tag{6.5}$$

where p is pressure. $\vec{v}$ is velocity, r is a spatial variable. and t is the time at which the instantaneous velocity is calculated.

Instead of keeping track of the velocity in the x and z directions throughout the calculations then calculating the intensity, it would be more efficient to calculate the intensity directly from the pressure. This can be achieved by using the formulation for a *sound intensity probe*, an experimental instrument that measures pressure at two points (two transducers) separated by a small distance. allowing the calculation of the intensity using the pressure information. The formulation applied to the intensity probe calculations is followed here where pressure values at two consecutive computational grid points separated by a small grid spacing is considered equivalent to the two transducer measurements separated by a small distance.

A brief derivation for intensity in terms of pressure. following that in Ref. 18. begins with Euler's equation. In its one-dimensional form it is written as

$$\frac{\partial p(t)}{\partial \xi} = -\rho_o \frac{\partial v(t)}{\partial t} \ . \tag{6.6}$$

Remember that ρ_o is the ambient density. Also ξ is an arbitrary spatial direction: the horizontal and vertical components of the intensity will have to be calculated separately on the rectangular computational grid. Solving for velocity. Eq. (6.6) becomes

$$v(t) = -\frac{1}{\rho_o} \int_{-\infty}^{t} \left(\frac{\partial p(\tau)}{\partial \xi} \right) d\tau \ . \tag{6.7}$$

where τ is here simply a dummy time variable. It should be noted that the integration involves all times prior to the one of interest. Realistically, sound field information becomes available at some initial time ($t = 0$), so $-\infty$ can be replaced by 0. Now using a finite difference approximation for the spatial derivative, $\frac{\partial p(\tau)}{\partial \xi} = \frac{p_2 - p_1}{d}$, Eq. (6.7) is written

$$v(t) = \frac{1}{\rho_o d} \int_0^t [p_1(\tau) - p_2(\tau)] \, d\tau \ . \tag{6.8}$$

where d is the distance separating p_1, the pressure at location ξ, and p_2, the pressure at location $\xi + d$. Equation (6.8) gives the velocity at the location $\xi + \frac{d}{2}$; when calculating the intensity it is then necessary to use the pressure also at the location $\xi + \frac{d}{2}$,

$$p\left(\xi + \frac{d}{2}\right) \approx \frac{p_1 + p_2}{2} \ . \tag{6.9}$$

Now substituting Eqs. (6.8) and (6.9) into Eq. (6.5), one component of the instantaneous intensity can be written

$$I_\xi(t) \approx \frac{1}{2\rho_o d} [p_1(t) + p_2(t)] \int_0^t [p_1(\tau) - p_2(\tau)] \, d\tau \ . \tag{6.10}$$

Equation (6.10) is written into the computer code with a summation approximating the integration as

$$I_\xi(t) \approx \frac{\Delta t}{2\rho_o d} [p_1(t) + p_2(t)] \sum_{\tau=0}^t [p_1(\tau) - p_2(\tau)] \ . \tag{6.11}$$

where the sum accumulates starting from the initial waveform. In Eq. (6.11) ξ is x or z, and d is Δx or Δz. The instantaneous intensity is then plotted using its horizontal component I_x and its vertical component I_z. The computer code for these intensity calculations is found in Appendix B as subroutine **intensity**.

6.4.2. Runs with intensity

Having the ability to calculate the intensity field, Fig. 6.4 can now be shown with the intensity vectors superimposed: refer to Fig. 6.6, where the inverted gray scale is applied so as not to distract from the intensity vectors. Here the speed of sound in the air is 343 m/s and the speed of sound in the ocean is 1500 m/s, representing realistic parameters for each medium (assuming a homogeneous ocean). The incoming waveform is represented by vectors pointing down and to the right and the reflected waveform is represented by vectors pointing up and to the right. Just under the ocean surface, it is seen that two sets of arrows, one for each peak of the sonic boom waveform, each form a scoop shape which decays with depth; this represents the evanescent wave caused by the impinging sonic boom. It should be noted that the intensity amplitude in the air is actually much greater than in the water; in order to even see any vectors in the ocean, the intensity vectors in that area of the domain are greatly magnified. Also, the density of the air intensity vectors is less than for the ocean vectors simply because it is more clear to see the incoming and reflected waves with fewer arrows.

As a comparison to Fig. 6.6 which illustrates the evanescent wave, Fig. 6.7 shows a propagating wave. Here the speed of sound in air is still 343 m/s, but the speed of sound in the liquid is unrealistically set to be 654 m/s throughout. According to Snell's Law, Eq. (2.2), for the Mach 2.4 case (incident angle is 24.6°) there should be a propagating wave traveling at an angle of 52.5° from the vertical into the water. Figure 6.7 depicts this scenario: the vectors in the propagating wave are all seen to be pointing down and to the right as is the case for the incident wave. Taking the horizontal and vertical components of the intensity vectors in the underwater propagating wave, the refracted angle θ_2 is calculated to be 52.6° ($\theta_2 = \tan^{-1}(\frac{I_x}{I_z})$), an angle which approximates that calculated using Snell's Law.

Another case where the incident wave should propagate into the water is when, using the realistic ocean sound speed (for a homogeneous ocean), the angle of incidence is less than the critical angle, 13.2°: this angle corresponds to a Mach number of approximately 4.4. Therefore, if the simulation is run with a Mach number of 4.5, a propagating wave should be seen underwater. Figure 6.8 shows the Mach 4.5 case; as in Fig. 6.7, the Mach 4.5 illustration depicts the propagating wave scenario. This propagating wave is seen more with

pressure and intensity; M=2.4; wh=0.0 m; #160
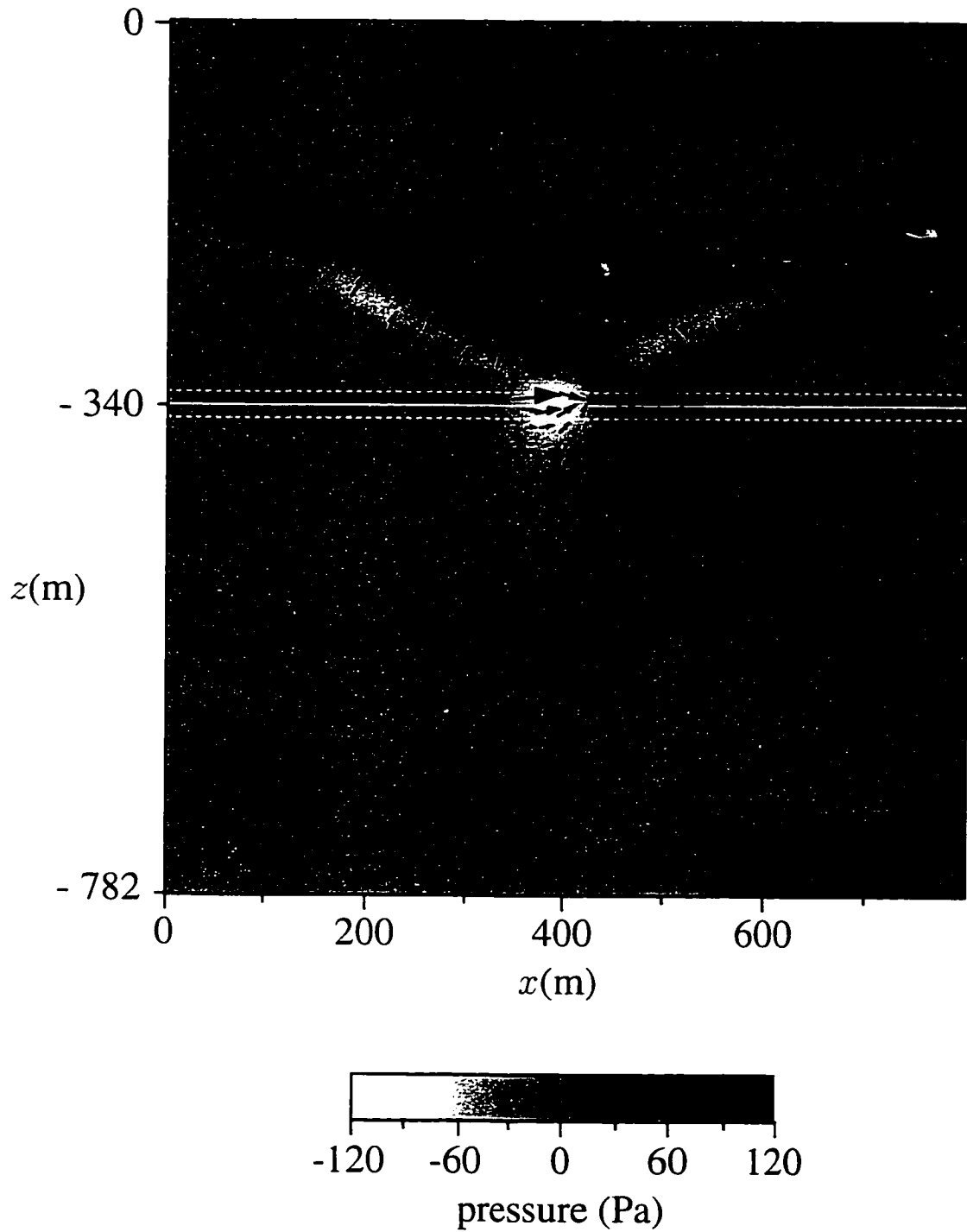
Figure 6.6: Mach number 2.4, wave height 0.0 m; incident wave interacting with the air-water interface; pressure and intensity fields for $c_{air} = 343$ m/s and $c_{ocean} = 1500$ m/s.

pressure and intensity; M=2.4; wh=0.0 m; #160; not real c
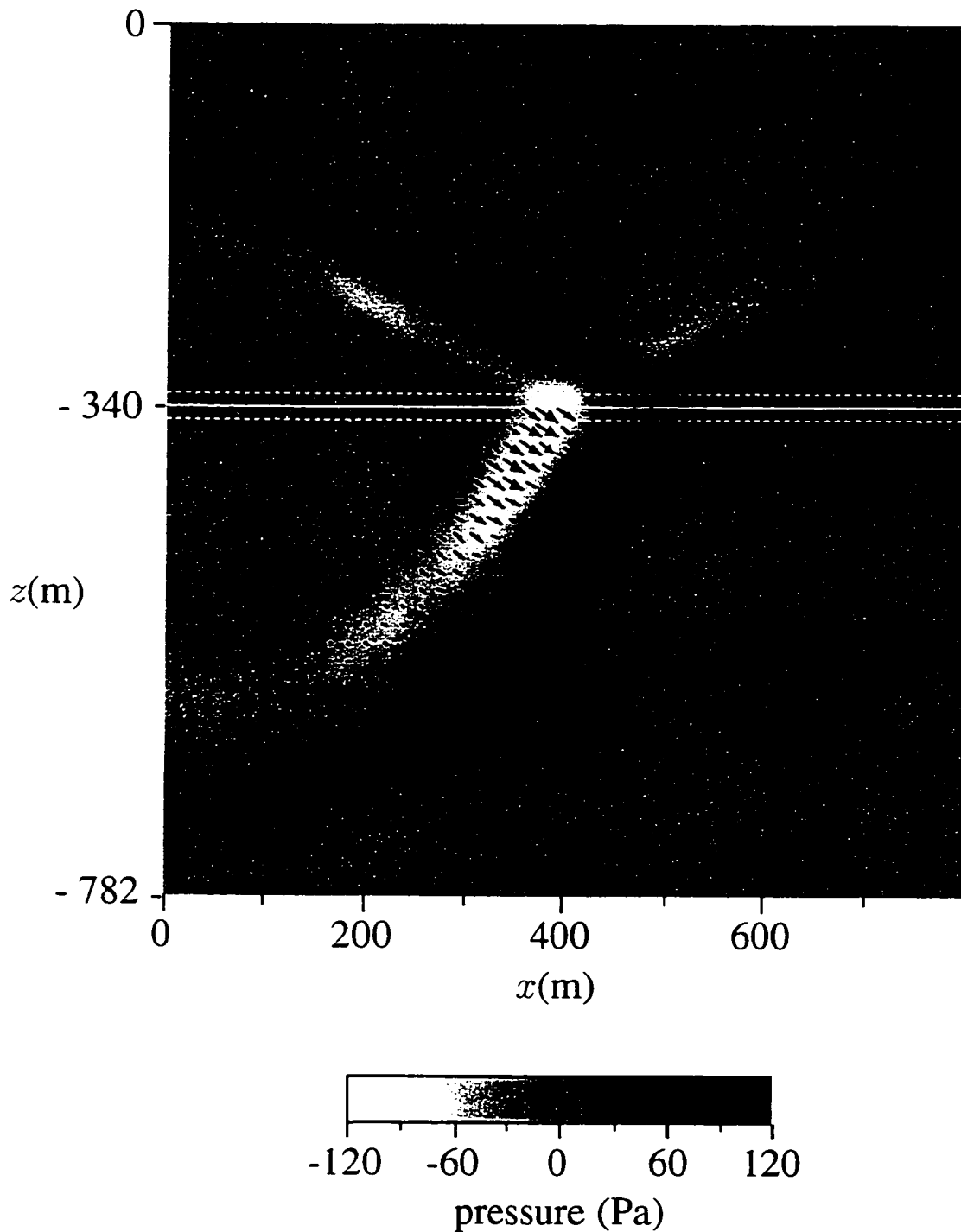
Figure 6.7: Mach number 2.4, wave height 0.0 m; incident wave interacting with the air-liquid interface; pressure and intensity fields for $c_{air} = 343$ m/s and $c_{liquid} = 654$ m/s.

pressure and intensity; M=4.5; wh=0.0 m; #120
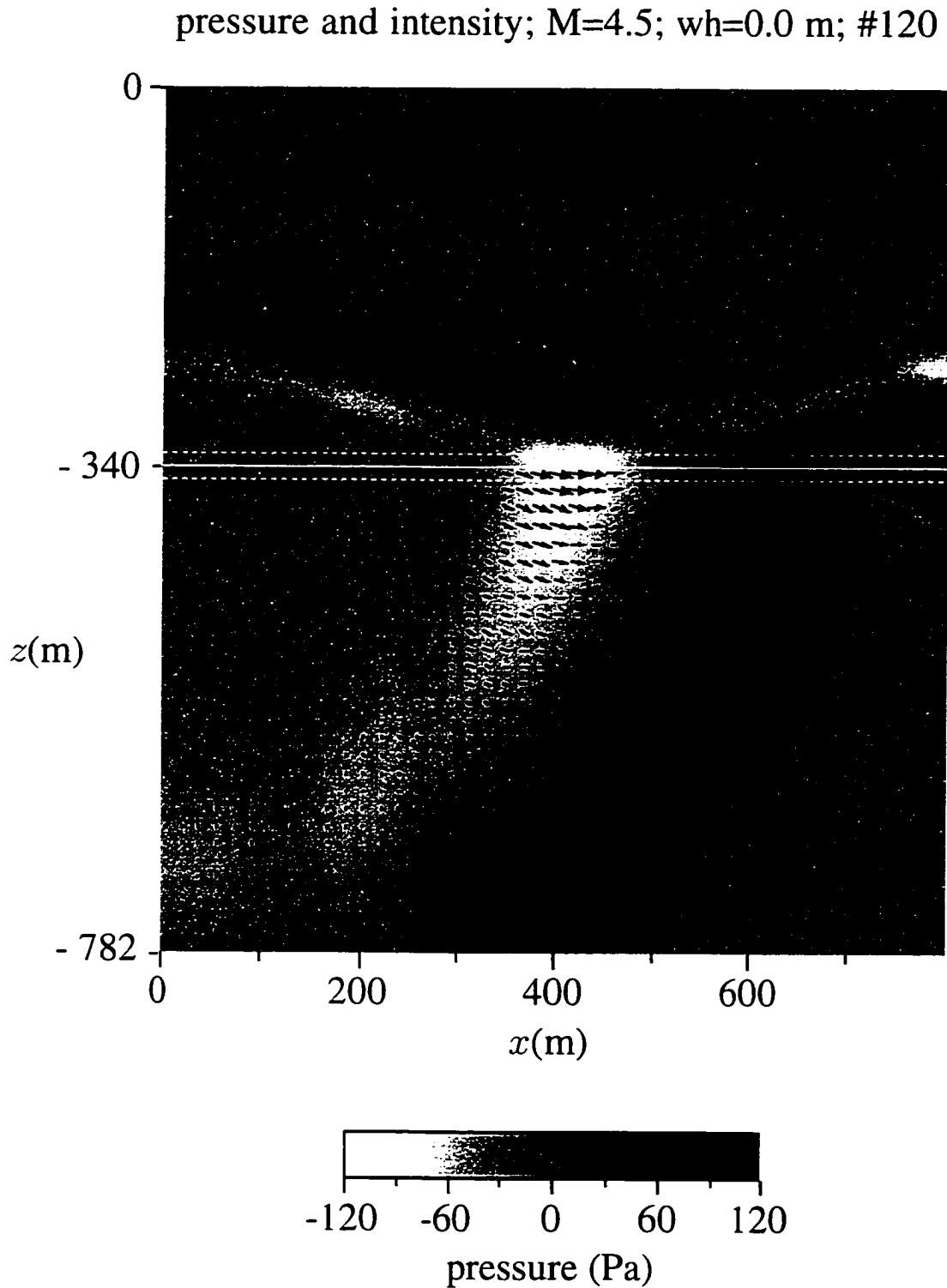
Figure 6.8: Mach number 4.5, wave height 0.0 m; incident wave interacting with the air-water interface; pressure and intensity fields for $c_{air} = 343$ m/s and $c_{ocean} = 1500$ m/s.

the positive peak of the sonic boom waveform: because the Mach 4.5 case would require a much wider domain in order to see the whole sonic boom waveform as a fully developed propagating wave. Fig. 6.8 (which uses the same size domain as the other simulations) illustrates the beginnings of a propagating wave, the negative part of the waveform not fully developed.

Not only do the figures with intensity vectors help in the understanding of the underwater sound field. but the cases used to illustrate a propagating wave serve as a reality check. The computer code correctly calculates a propagating wave where physics demands a propagating wave and an evanescent wave when appropriate.

6.5. Comparison of Results to Previous Theories

The analytical theories of Sawyers,[53] Cook.[12] and Sparrow[58] assume a flat water interface. The finite difference simulation results for the flat ocean surface can be qualitatively compared with these analytical theories. Although the finite difference simulations use a rounded sonic boom waveform and the analytical theories assume a perfectly N-shaped waveform. the finite difference results are similar to those predicted analytically.

6.5.1. Comparison to theories of Sawyers and Cook: decay in pressure

When the sonic boom interacts with the ocean surface. an evanescent wave is produced under water; this evanescent wave is analytically predicted by Sawyers[53] and Cook[12] as explained in Chapter 3. Figure 6.6 shows the evanescent wave, but it would also be useful to see the decay of the actual sonic boom waveform. Line graphs extracted at various depths below the surface illustrate the decaying waveform as described by both Sawyers and Cook. Figure 6.9 shows a horizontal line intersection of the full domain just under the ocean surface and at 10, 25, and 50 m under water. The pressure amplitude just under the ocean surface is approximately 100 Pa due to pressure doubling at the air-water interface. Figure 6.9 shows that the amplitude of the penetrating wave decays as a function of depth and is comparable to Fig. 3.1. the illustration of a decaying waveform calculated using Sawyers' theory.
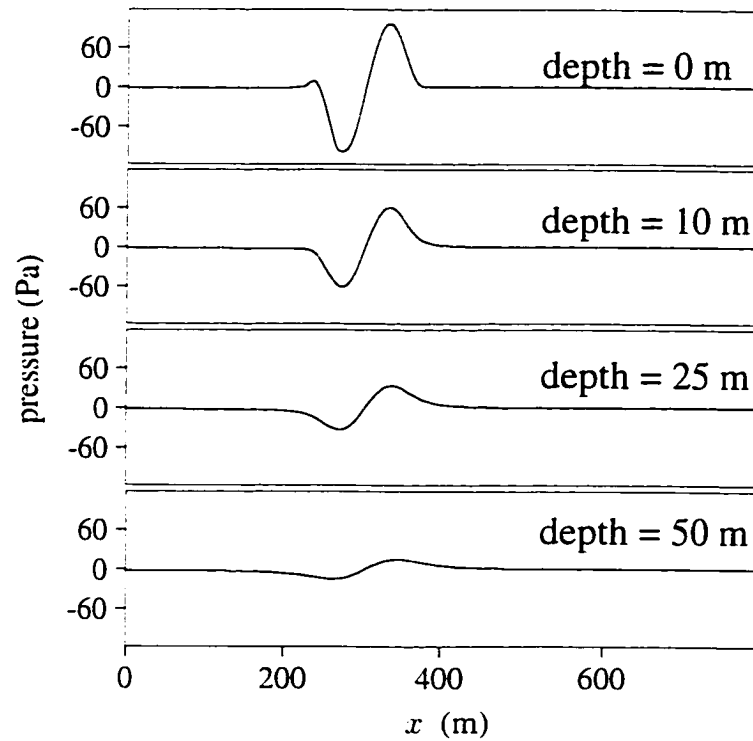
Figure 6.9: Mach number 1.4, wave height 0.0 m: horizontal line intersection of waveform just under the ocean surface and 10, 25, and 50 m under water.

Although the decaying waveform calculated using Sawyers' theory starts (at the surface) as a perfect N wave and the decaying waveform in the flat water simulations starts as a rounded sonic boom, a comparison can be made of the two decaying waveforms. Figure 6.10 shows each waveform on the same scale just under the ocean surface and at depths of 10, 25, and 50 m: Part (a) is the analytical N wave and Part (b) is the computationally calculated rounded sonic boom. It is seen that both waveforms decay at approximately the same rate, the N wave losing its sharpness and looking more like the rounded waveform. Figure 6.11 includes each 25 meter-depth waveform: the analytical waveform has been shifted to the left for a direct comparison. Figure 6.11 reiterates that the waveforms have decayed at approximately the same rate, the peaks quite close in amplitude. It is concluded that, in fact, the flat water simulations approximate well the analytical work of Sawyers and Cook

who predict that a sonic boom related to a Mach speed less than 4.4 impinging upon a flat-surfaced homogeneous ocean cause an underwater evanescent wave.

6.5.2. Comparison to theories of Sawyers and Cook: decay in sound level

The flat ocean surface results are now converted into sound levels and the decrease in decibels with depth are compared to the results presented in Chapter 4 and the results in Sparrow and Ferguson.[60] Each of these results are produced using a numerical method based on one of the known analytical theories.

6.5.2.1. Comparison to sound level decay table in Chapter 4

Chapter 4 presented several sound level calculations including the peak sound pressure level. Table 4.2 listed the decrease in sound level from the ocean surface as a function of depth. The pressure values used to produce these sound levels were numerically calculated using the theory of Sawyers,[53] as explained in Chapter 4. In order to compare the computational simulations of Chapter 6 with the values listed in that table, the peak pressure results calculated with the simulations had to be converted to sound level. After calculating the peak pressure sound level, a list similar to that in Table 4.2 of sound level decay as a function of depth is constructed.

Table 6.1 contains the decay in decibels as a function of depth calculated using both the computational model and the analytical model. (The model actually uses numerical methods based on analytical theories, but it will be referred to here simply as the analytical model.) Also, the difference between these two ($|$computational$|$ − $|$analytical$|$) is shown. Recall that for the analytical method of Chapter 4, a perfectly symmetric N wave is inserted; when calculating the peak pressure sound level, either the positive or negative peak can be chosen since they have the same magnitude, and hence one decibel value for each depth is found. The simulations in Chapter 6 involved a rounded N wave as the initial sonic boom waveform; just under the ocean surface and below, the peak pressures of the waveform do not have the same magnitude. To get one number for decay in decibels at a particular depth the sound levels associated with the positive and negative peaks are averaged, and the average decay is shown as a function of depth.
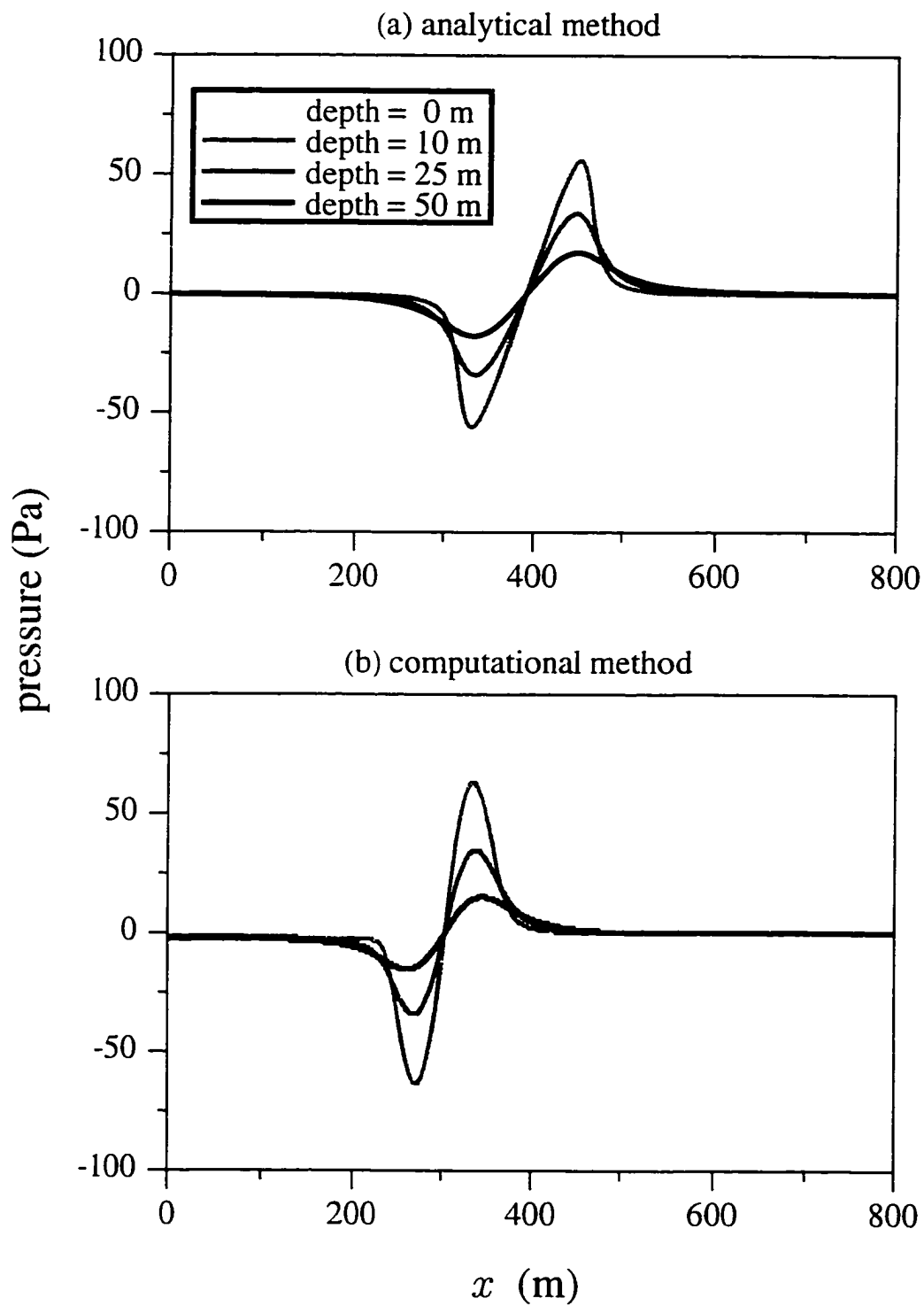
Figure 6.10: Sonic boom waveform just under the ocean surface and 10, 25, and 50 m under water. Part (a) analytical method; Part (b) computational method.
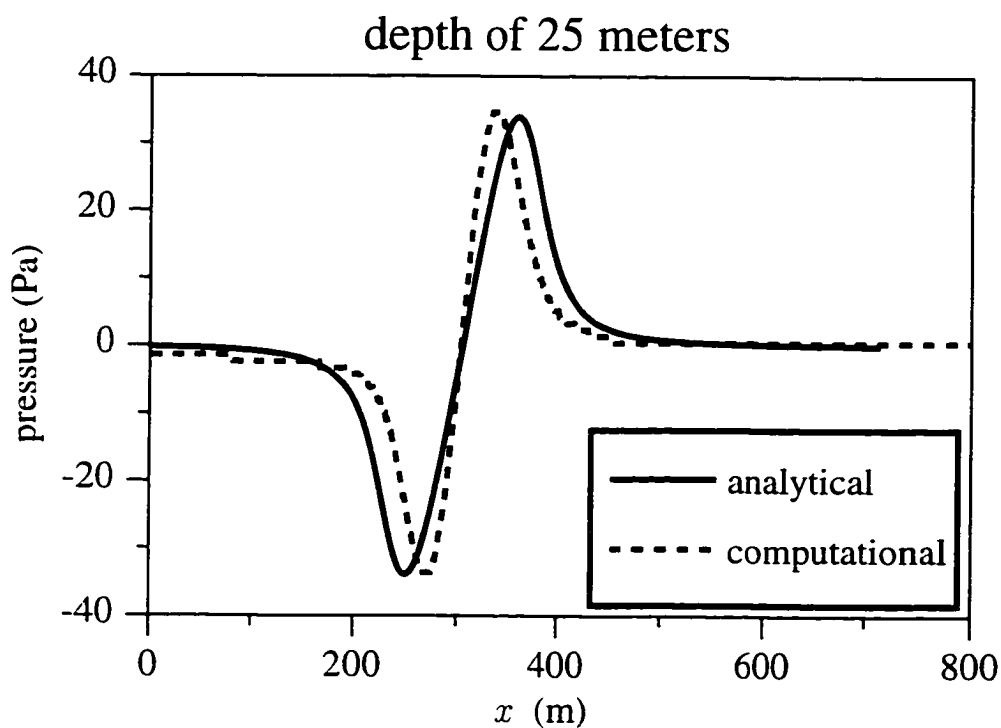
Figure 6.11: Sonic boom waveform at a depth of 25 m under water. Solid line, analytical method: dashed line, computational method.

Table 6.1: Decibel levels as a function of depth relative to those at the surface: a comparison of the computational model to the analytical model of Chapter 4. Flat homogeneous ocean.

depth (m)	computational method (dB)	analytical method (dB)	dB difference $|comp| - |analytical|$
0	0.0	0.0	0.0
4	−0.8	−2.0	−1.2
8	−1.6	−3.0	−1.4
16	−3.1	−4.6	−1.5
32	−6.1	−7.2	−1.1
64	−11.3	−11.6	−0.3
128	−19.4	−18.4	−1.0

The computational method is seen to decay more slowly near the surface and more quickly at greater depths than the analytical method. The largest difference in sound level, -1.5 dB, is at a depth of 16 m. This is not a great amount since the threshold of detecting an amplitude change for most people is 3 dB. The remaining differences are less than 1.5 dB in magnitude. So, although there is a notable difference, the underwater decay rate of the evanescent field due to a sonic boom is approximately the same for the computational simulations as for the analytical solutions.

6.5.2.2. Comparison to sound level decay table in Sparrow and Ferguson

A description of Sparrow and Ferguson's analytical method[60] is found in Sect. 3.2.2. The method is based on the theory of Cook.[12] In their work, a table was presented, labeled Table 3, similar to Table 4.2 in this research, except with fewer depths represented. Their theory also involved a perfectly symmetric N-shaped sonic boom with the same parameters as have been used for most of the simulations in this work. (As in the last section, the model actually uses numerical methods based on analytical theories, but again will be referred to as the analytical model.)

Table 6.2 contains the decay in decibels as a function of depth calculated using both the computational model and the analytical model. The values for the computational model are the same as those in Table 6.1. Again, the computational method is seen to decay more slowly near the surface and more quickly at greater depths than the analytical method. The largest difference in sound level, -1.6 dB, is at a depth of 16 m. The same conclusions are drawn as in the previous section.

Table 6.2: Decibel levels as a function of depth relative to those at the surface: a comparison of the computational model to the analytical model in Ref. 60. Flat homogeneous ocean.

depth (m)	computational method (dB)	analytical method (dB)	dB difference |comp| − |analytical|
0	0.0	0.0	0.0
4	−0.8	−2.0	−1.2
16	−3.1	−4.7	−1.6
64	−11.3	−11.6	−0.3

6.5.3. Comparison to Sparrow's conclusions

In order to see the relationship between Mach number and the underwater depth of the penetrating sound. it is necessary to extract information from the flat water Mach 1.4. 2.4. 3.0. and 3.5 runs. Table 6.3 contains the numerical data (real pressure values) found at a 100 m depth for increasing Mach numbers. It is seen that for Mach 1.4 the evanescent wave has decayed to about 5 percent of its value just under the ocean surface (100 Pa). However. for Mach 3.5 the peak pressures in the waveform at a depth of 100 m are still 40 percent of the value just under the surface. Hence. the numbers confirm Sparrow's prediction[58] which states that higher Mach number incident waves are associated with deeper penetration into the ocean.

Table 6.3: Real pressure values found at 100 m depth for increasing Mach numbers.

Mach number	positive peak (p, Pa)	negative peak (p, Pa)
1.4	4.19	-5.71
2.4	14.06	-15.98
3.0	21.71	-24.96
3.5	40.71	-38.20

An explanation for this calculated trend goes back to Chapter 2. Section 2.3.4. where the evanescent wave is first discussed. Included in the expression for the underwater pressure field are β_2 and $T_{1.2}$ which are dependent on the incident angle of the incoming waveform and hence on the Mach number. Changes in $T_{1.2}$ modify the pressure amplitude and changes in β_2 modify the exponential decay governed by $e^{-(\omega/c_2)(\beta_2)z}$. Since $\beta_2 = [(\frac{c_2}{c_1})^2 \sin^2 \theta_1 - 1]^{\frac{1}{2}}$, it is determined that as the Mach number increases. the incident angle θ_1 decreases (in the range $0°$ to $90°$); in turn. $\sin \theta_1$ decreases and β_2 decreases. This implies that the rate of decay decreases with increasing Mach number. or higher Mach numbers are associated with deeper penetration into the ocean.

Chapter 7.
Homogeneous Wavy Ocean: Concepts and Computational Simulations

It was shown in Chapter 6 that computational simulations are possible for a sonic boom impinging on a homogeneous ocean with a flat surface. The current chapter introduces the first realistic ocean feature, a wavy ocean surface. Since the goal here is to find the effects that just the curvature of the wavy ocean surface has on the incoming sonic boom, the ocean remains homogeneous for this chapter's simulations. Before describing the simulations, ocean wave concepts will be discussed, specifically the topics of wind-generated waves and sonic booms interacting with wind-generated waves. The computational simulations presented here will include ocean waves being represented by a simple curved surface and more complex curved surfaces, the surface profile shown for each case. Also, a wavelength comparison will be made between the ocean waves and the horizontal effective wavelength of the sonic boom waveform, and the effects due to the wavelength difference will be shown.

7.1. Ocean Wave Concepts

7.1.1. Wind-Generated waves

7.1.1.1. Wave generation

Although ocean waves can be caused by the passage of the moon or underwater earthquakes or landslides, among other things, the primary cause of waves is the wind.[3,22] As the wind blows over the surface of the ocean, the frictional drag of the moving air against the water creates ripples. Once a ripple has formed, there is a steep side against which the wind presses directly. The energy is then transferred from air to water more effectively and the small waves grow rapidly.[3]

As the wind speed increases. and if it is of any significant duration, larger waves begin to develop as the wave height builds up.[22] The ultimate state of wave growth depends primarily on three parameters basic to wave forecasting: 1.) the wind velocity, 2.) the duration of time the wind blows. and 3.) the fetch (the distance over open water across which the wind blows).[3,22] Eventually, the waves have absorbed as much energy as they can from the wind of that velocity; energy is being dissipated by the waves at the same rate as the waves receive energy from the wind. At this point the sea is said to be *fully developed*.[3,7] A fully developed sea at any given time is very complex and defies precise description.[22] Statistical methods can be used to describe the properties of these waves.

The *significant wave height* in a fully developed sea is the average height of the highest one-third of all waves occurring in a particular time period. The Beaufort Scale (found in Refs. 3. 7. and 22. among others) gives the relationship between sea state, significant wave height. and wind speed and is used by mariners for estimating the wind speed from the sea state. An example entry in the scale reads: at a sea state of 5. the wind can be described as a fresh breeze. the wind velocity is from 17–21 knots (kn) or 8.0–10.7 m/s. many white caps have formed. and the significant wave height is about 1.2–2.4 meters. The scale can be used in this work to help pick related wind speeds and wave heights and to help determine which wind speeds are associated with breaking waves: the breaking wave issue will be discussed more in the next chapter.

Wind waves leaving the generating area or continuing on after the generating wind has ceased are termed *swell*. Swell are more rounded and symmetrical than waves in the fully developed sea, their form approaching that of a true sine curve. They move in groups of similar period and height and can travel thousands of miles across deep water before being completely diminished by air resistance, turbulence. dispersion, and lateral spreading.[3,22] The conceptual development of wind-generated ocean waves is shown in Fig. 7.1: this figure is adapted from illustrations in Refs. 3 and 22.
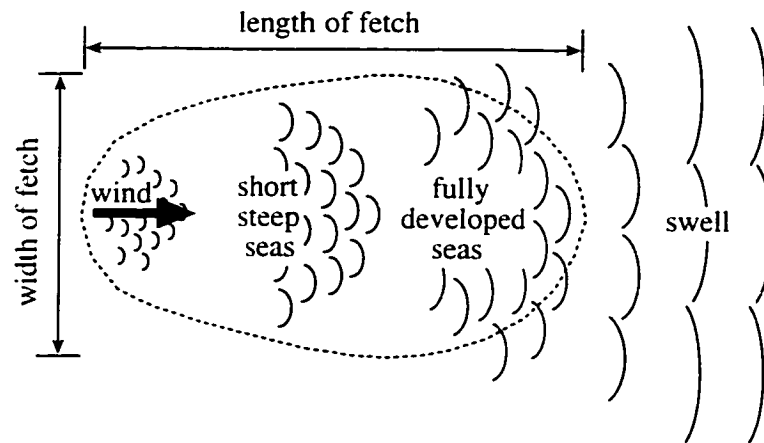
Figure 7.1: Conceptual development of wind-generated ocean waves. Top down view.

7.1.1.2. Analytical description of waves

Since swell cover such an extensive area, these are the type of ocean waves which are considered to be most important to the research. Swell can be approximated analytically by a sinusoidal profile. The expression for this sinusoidal profile is

$$\eta(x) = A\cos kx . \tag{7.1}$$

where x is the horizontal distance, η is the vertical displacement of the water surface from the mean water level, A is the amplitude of the ocean wave, $k = \frac{2\pi}{L}$ is the ocean wave number, and L is the wavelength of the ocean wave. The wave height H, significant to the simulations, is defined as the vertical distance between the crest and successive trough of the ocean wave; the amplitude A is half the wave height. Another parameter is the steepness which is the wave height over the wavelength (H/L); wave steepness observed at sea is normally between $1/17$ and $1/33$, the steeper waves corresponding to higher wind speeds. In this research, the steepness is fixed at $1/20$ (or $L = 20 \times H$), a common ocean engineering approximation.[22] The sinusoidal profile is seen in Fig. 7.2; also shown in the figure are the ocean wave parameters.

Although the sinusoidal surface is a fair approximation for describing swell, the steeper the waves become, the more the surface resembles a trochoidal curve.[7] Using a trochoid
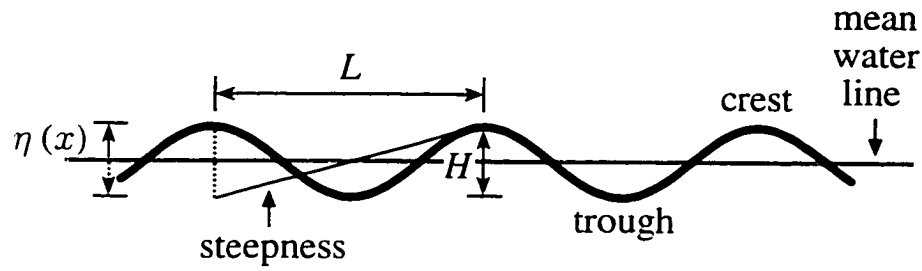
Figure 7.2: Sinusoidal ocean surface profile with wave parameters.

to represent the ocean swell was first established by Stokes[63] and improved by Rayleigh.[48] Stokes' waves form a trochoidal profile where the crests are steeper and the troughs are flatter: this representation more realistically describes observed ocean waves.[22] A third-order expression for a Stokes' wave is provided in Refs. 22 and 31:

$$\eta(x) = -A \cos kx + \frac{1}{2} kA^2 \cos 2kx - \frac{3}{8} k^2 A^3 \cos 3kx , \qquad (7.2)$$

where the variables are the same as in Eq. (7.1). Figure 7.3 shows the trochoidal profile.

Figure 7.3: Trochoidal ocean surface profile.

Besides the swell. profiles can be formulated for waves in a fully developed sea. The most realistic formulation involves statistically generated surfaces. Due to computational limitations these surfaces are not actually used in any of the final result simulations: numerical instabilities could not be eliminated for their general use, even after minimizing the number of frequency components. Refer to Appendix C for a formulation of a statistically generated surface profile.

Instead of implementing the statistically generated surface profile, two profiles more complex than a simple sinusoid or trochoid are applied. Each of the complex profiles contains not only a trochoidal wave with a frequency equivalent to that used for the swell but also a unique characteristic which represents a higher frequency component in the ocean waves. The first complex profile is calculated using the equation

$$\eta(x) = -\frac{A}{2}\cos kx + \frac{1}{2}kA^2\cos 2kx - \frac{3}{8}k^2A^3\cos 3kx + (0.6494)\, A\cos 2kx\ . \qquad (7.3)$$

Again, the variables match those in Eq. (7.1). This profile is a trochoidal wave with a sine wave of lower amplitude and half the wavelength superimposed. The shape of this surface is seen in Fig. 7.4. The second complex profile is calculated using the equation

$$\eta(x) = -\frac{A'}{2}\cos kx + \frac{1}{2}k\,(A')^2\cos 2kx - \frac{3}{8}k^2\,(A')^3\cos 3kx + (0.3)\,A'\cos 3kx\ . \qquad (7.4)$$

where $A' = (1.2481)A$. This profile is a trochoidal wave with a sine wave of lower amplitude and one-third the wavelength superimposed. The shape of the second complex surface is seen in Fig. 7.5. It should be noted that the amplitude adjustments in Eq. (7.3) and in Eq. (7.4) are necessary to achieve an ocean wave with height equal to that which is input.

Figure 7.4: First complex ocean surface profile: trochoid plus half wavelength sinusoid.

Figure 7.5: Second complex ocean surface profile: trochoid plus one-third wavelength sinusoid.

7.1.2. Sonic booms interacting with wind-generated waves

It is important to look at the alignment of the propagation directions of a sonic boom and wind-generated ocean waves. In studying this aspect. it is possible to determine whether or not two dimensions sufficiently represent the problem at hand or if three dimensions are necessary.

7.1.2.1. Sonic boom propagation aligned with ocean waves

It is first assumed that the horizontal component of the sonic boom propagation is aligned with the wind-generated ocean waves' propagation direction: this can be restated as the sonic boom propagation being perpendicular to each ocean wave front. Figure 7.6 illustrates this case.
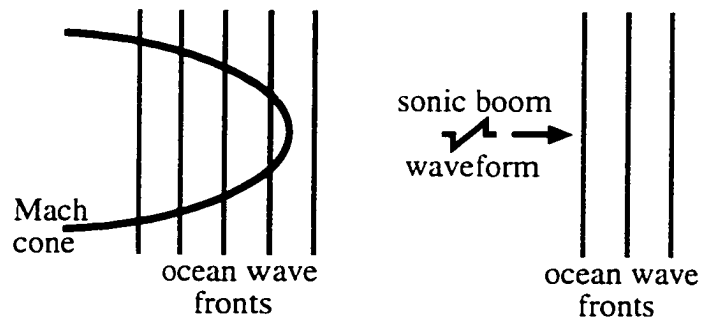
Figure 7.6: Sonic boom propagation aligned with ocean waves' propagation direction. Top down view.

Taking a horizontal slice through Fig. 7.6 gives an adequate two-dimensional representation of the aligned case. This is true since any reflections of the sonic boom would stay in the initial plane of propagation. In other words, using the law of mirrors. the angle of reflection can be determined simply by setting it equal to the angle of incidence. Refer back to Fig. 2.13 which shows the angle of incidence and the angle of reflection for a *flat* ocean surface.

Finding the angle of reflection from a *wavy* ocean surface is somewhat more complicated than for a flat surface. Figure 7.7 represents a horizontal cut through the middle of Fig. 7.6: as you can see from the figure. it is necessary to determine the tangent line at a point on the ocean surface in order to determine the true incident angle and thus the angle of reflection.
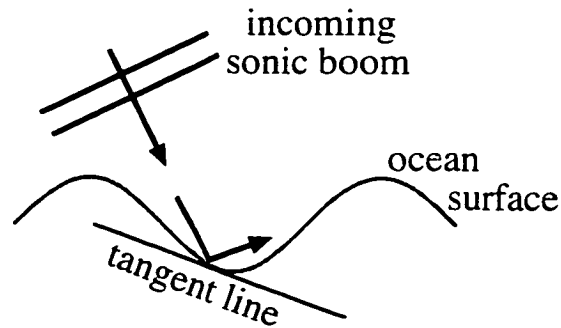
Figure 7.7: Two-dimensional representation of sonic boom interacting with a wavy ocean surface. Side view.
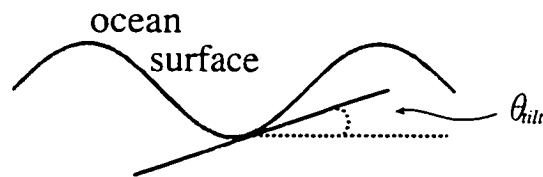
Figure 7.8: The angle which the tangent line sweeps from the horizontal.

The angle which the tangent line sweeps from the horizontal, as is seen in Fig. 7.8, can be found using

$$\theta_{tilt} = \tan^{-1} m \ . \tag{7.5}$$

where θ_{tilt} is the tilt angle and m is the slope of the tangent line. If the function of the ocean surface profile is known. the slope of the tangent line can be simply calculated by taking the spatial derivative at any point x on the surface. So. Eq. (7.5) becomes

$$\theta_{tilt} = \tan^{-1} \frac{\partial \eta}{\partial x} \ . \ \cdot$$
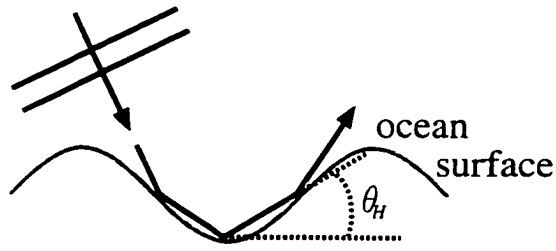
Figure 7.9: A single acoustic ray in air traced as it enters then exits the ocean wave trough.

where η is the function of the ocean surface profile. Then the true incident angle θ_I, the angle from the vertical to the line tangent to the surface curve at the point of reflection, can be written as

$$\theta_I = \theta_\iota - \theta_{tilt} \,. \tag{7.6}$$

where θ_ι is the incident angle measured from the vertical to a flat ocean surface.

Assuming a simple sinusoidal ocean surface profile, the tilt angle as a function of horizontal distance is found as

$$\theta_{tilt}(x) = \tan^{-1}\frac{\partial \sin\left(x\frac{180}{\pi}\right)}{\partial x} \,.$$

and the true incident angle in degrees can be calculated using

$$\theta_I(x) = \theta_\iota - \tan^{-1}\frac{\partial \sin\left(x\frac{180}{\pi}\right)}{\partial x} \,.$$

This sinusoidal ocean profile will be used in the following demonstration.

It is useful to trace rays going into the ocean wave trough in order to find out what happens to the acoustic energy. To illustrate the ray tracing, it is most useful to find the angle measured from the horizontal to the reflected ray. This angle θ_H can be written as

$$\theta_H(x) = 90° + \theta_{tilt} - \theta_I \,. \tag{7.7}$$

Figure 7.9 shows a single ray traced as it enters then exits the ocean wave trough. The ray reflects at three points: the angle θ_H is shown for the second reflection. Equation (7.7) is used to calculate each angle of reflection as seen from a horizontal intersection. Because of the large acoustic impedance mismatch between air and water it can be assumed that each reflection is essentially lossless.
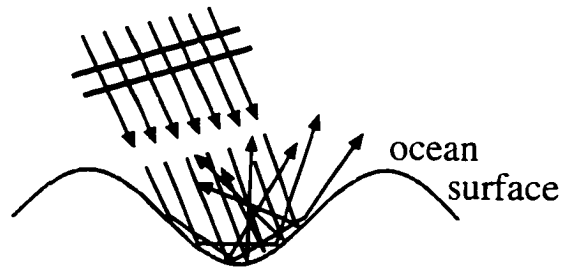
Figure 7.10: Multiple acoustic rays in air traced as they enter then exit the ocean wave trough.

A sonic boom wavefront hitting the ocean surface can be visualized in air using several rays. each representing a different point on the wavefront. Figure 7.10 shows multiple ray paths as they enter then exit the ocean wave trough.

As is seen in Fig. 7.10. the rays intersect to form an area of concentrated acoustic energy. It is this concentration which makes the aligned case the worst case scenario: worst in that it has the greatest build-up of acoustic energy.

7.1.2.2. Sonic boom propagation not aligned with ocean waves

It is now assumed that the horizontal component of the sonic boom propagation is not aligned with the wind-generated ocean waves' propagation direction: the sonic boom propagation is not perpendicular to the ocean wave fronts. Figure 7.11 illustrates this case. The ocean wave fronts shown in Fig. 7.11 are now shifted from the vertical by angle ψ.

The ocean surface profile in the misaligned case is the same as that in the aligned case. So. using the ray tracing illustration, the angle of reflection is calculated in the same manner. but now an out-of-plane sweep of angle ψ is included. Since this problem cannot be restricted to one plane. a three-dimensional representation is required. Figure 7.12 shows a top view of the ocean surface with a single ray traced: this is just an illustration and does not include accurate reflection points. What it does convey, however, is that the acoustic energy is not trapped in the one plane. as with the aligned case: there is therefore less concentration of energy in any single area.
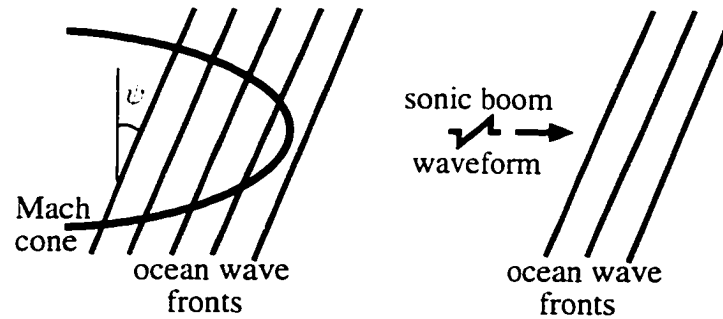
Figure 7.11: Sonic boom propagation not aligned with ocean waves' propagation direction. Top down view.
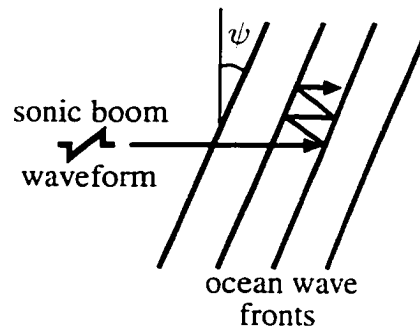
Figure 7.12: Top down view of the ocean surface with a single ray traced: this illustration does not include accurate reflection points.

After analyzing the cases of sonic boom propagation being aligned and not aligned with the ocean wave propagation direction, it is determined that the worst case scenario belongs to the aligned case. In support of this, it has been recently reported that the strongest effect occurs when the sonic boom propagation direction is normal to the ocean wavefronts;[8,9] this describes the aligned case. Since the interest lies in finding the strongest effect possible, the aligned case is the one chosen for modeling: a two-dimensional simulation is sufficient to study this aligned case. Going to a fully three-dimensional finite difference simulation is therefore unnecessary.

7.2. Homogeneous Ocean with a Simple Curved Surface: Computational Simulations

The first set of two-dimensional computational simulations for a wavy ocean surface includes the two simple surface profiles, sinusoidal and trochoidal; these profiles represent the ocean swell. A necessary addition to the computer code and the physical parameters used in these simulations will be described. followed by the enumeration of specific program runs and the presentation of their results.

7.2.1. Computer program

The same computer program that was used in Chapter 6 for the flat ocean surface results is used here. With the addition of the wavy ocean surface, however, numerical instabilities arise. Here the ocean surface curve is approximated by ragged grid blocks, its path never entering a grid block but rather outlining it. So instead of a nice. smooth surface. the profile resembles that shown in Fig. 7.13: this illustration does not give accurate dimensions. and the raggedness is greatly exaggerated. Because the surface is not smooth in the computational simulations. its "edges" sometimes cause the pressure calculations to become numerically unstable resulting in undesirable exponential growth in the solutions.
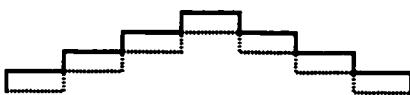
Figure 7.13: Illustration of grid-block-approximated ocean surface curvature.

Two steps are taken to control these instabilities: the first is the refined grid region surrounding the air-water interface already existing in the code. This decreases the vertical rise of the grid blocks, allowing a better approximation to a smooth surface. There are still, however, problems with the calculations. Depending on the ocean wave height, the crests and troughs of the ocean waves can create numerical instabilities. To avoid this

problem, a function or filter is inserted into the Fortran code which "smooths" any peak or dip in the surface with a singular maximum or minimum grid point. "Smoothing" the curve requires adjusting a singular extreme point to match the value of its neighboring grid points. Figure 7.14 illustrates the application of the "smoothing" filter. Once both of these steps have been taken, the computational simulations for a sonic boom interacting with a simple curved surface run without any numerical mishaps.
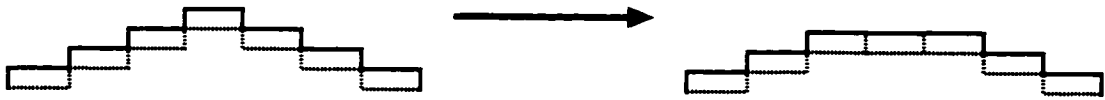
Figure 7.14: Illustration of the "smoothing" filter applied to the ocean surface curve.

7.2.1.1. Physical parameters

As in the flat ocean simulations, the ocean simulations with swell propagate a rounded sonic boom waveform. The initial waveform inserted in the computational domain is the same as in Chapter 6: the peak pressure of the sonic boom is 50 Pa and the duration 300 ms. The four Mach numbers used are 1.4, 2.4, 3.0, and 3.5.

The ocean surface profiles chosen to represent the swell are sinusoidal and trochoidal, each being a simple curved surface. Please refer to Sect. 7.1.1.2 for the equations used to calculate, and illustrations of, these surfaces; specifically Eq. (7.1) and Fig. 7.2 for the sinusoid and Eq. (7.2) and Fig. 7.3 for the trochoid. The wave heights chosen are associated with particular wind speeds. A plot in Ref. 3 shows two curves, one observed values and one theoretical values, for wind speed in knots (kn) vs. wave height in meters. Four different wind speeds and their corresponding wave heights are extracted from the observed wave height curve; the results are shown in Table 7.1. These wind conditions are described as calm (0 kn), gentle breeze (10 kn), fresh breeze (20 kn), and finally high wind (30 kn) in the Beaufort Scale. As was mentioned, this scale is found in many references; although the descriptions of the wind conditions and sea are similar, the wind speed and corresponding

wave height values differ from one reference to another. It was decided to use the values extracted from the plot in Ref. 3. given in Table 7.1. where most of the wave heights correspond to those in the Beaufort Scale. What is most important here is implementing several different wave heights to find their effect on an impinging sonic boom.

Table 7.1: Wind speeds and corresponding wave heights used in the curved interface simulations.

wind speed (knots)	ocean wave height (meters)
0	1.0
10	1.4
20	2.3
30	3.75

The air and the ocean are homogeneous for these simulations. In the air the speed of sound c_o is 343 m/s and the density ρ_o is 1.21 kg/m^3. In the ocean $c_o = 1500$ m/s and $\rho_o = 1000$ kg/m^3.

7.2.2. Program runs

The same computational parameters (grid spacing. number of grid points, ...) are used in the curved surface simulations as were use in the flat surface simulations in Chapter 6. For the calculations in this chapter three physical parameters are varied: the surface profile, the Mach number, and the ocean wave height. The Mach 1.4 and 2.4 cases are run for each of the four wave heights given in Table 7.1. Also run are the Mach 3.0 and 3.5 cases for the largest wave height, 3.75 m. with only the trochoidal interface profile. To find out how the curvature of a wavy ocean surface affects the impinging sonic boom waveform, the runs mentioned in Chapter 6 for a flat ocean surface for the Mach 1.4, 2.4, 3.0, and 3.5 cases are used for comparison. In all of this chapter's simulations, the ocean waves are static: this approximation is valid since the airplane speed far exceeds that of the ocean surface waves, so to the airplane the surface waves appear to be standing still.[9]

7.2.3. Effects caused by a simple curved ocean surface

An example of the pressure field over the entire computational domain at a time when the sonic boom is interacting with the wavy ocean surface (trochoidal profile) is seen in Fig. 7.15; the case shown is Mach 2.4, wave height 3.75 m at a time $t = 0.75$ seconds. As in Chapter 6, the dark band represents the positive peak of the sonic boom waveform and the light band the negative peak. The white curvy line is the ocean surface, and the white dashed lines enclose the refined grid region; as before, this graphic shows the refined region stretched vertically by a factor of four.

A great deal can be learned from these simple curved surface simulations. It is possible to look at the intensity vectors to visualize the power flow, to see how the curved interface warps the initial waveform, to animate this simulation and watch where high pressure values accumulate, and to extract underwater pressure values to see where and how much the sound field is modified and which parameters govern the modifications.

7.2.3.1. Results with intensity vectors and qualitative description of effects

It is useful to start by viewing the intensity vectors in the water; Figure 7.16 shows the same full-domain pressure field in Fig. 7.15 but now with the underwater intensity vectors superimposed. The vectors are shown only from the bottom of the troughs down and not in the peaks of the ocean waves; this is necessary in order to see any intensity vectors in the ocean. As was stated earlier, the intensity values in the air far exceed any in the ocean; when viewing a rectangular section (a limitation in the graphics), no air can be included or the ocean vectors will be so overpowered that they will appear as dots not arrows. Figure 7.16 reveals the underwater evanescent wave: this curved ocean surface, like the flat ocean surface, does not allow an incident wave to penetrate the surface as a propagating wave.

It is also useful to look at the intensity vectors in the air: Figure 7.17 shows the same full-domain pressure field in Fig. 7.15 but now with the air intensity vectors superimposed. Some of the ocean vectors are actually part of this graphic and appear as dots in the peaks of the ocean waves. What can be learned from this plot is that there is a build-up

pressure; M=2.4; wh=3.75 m; #160
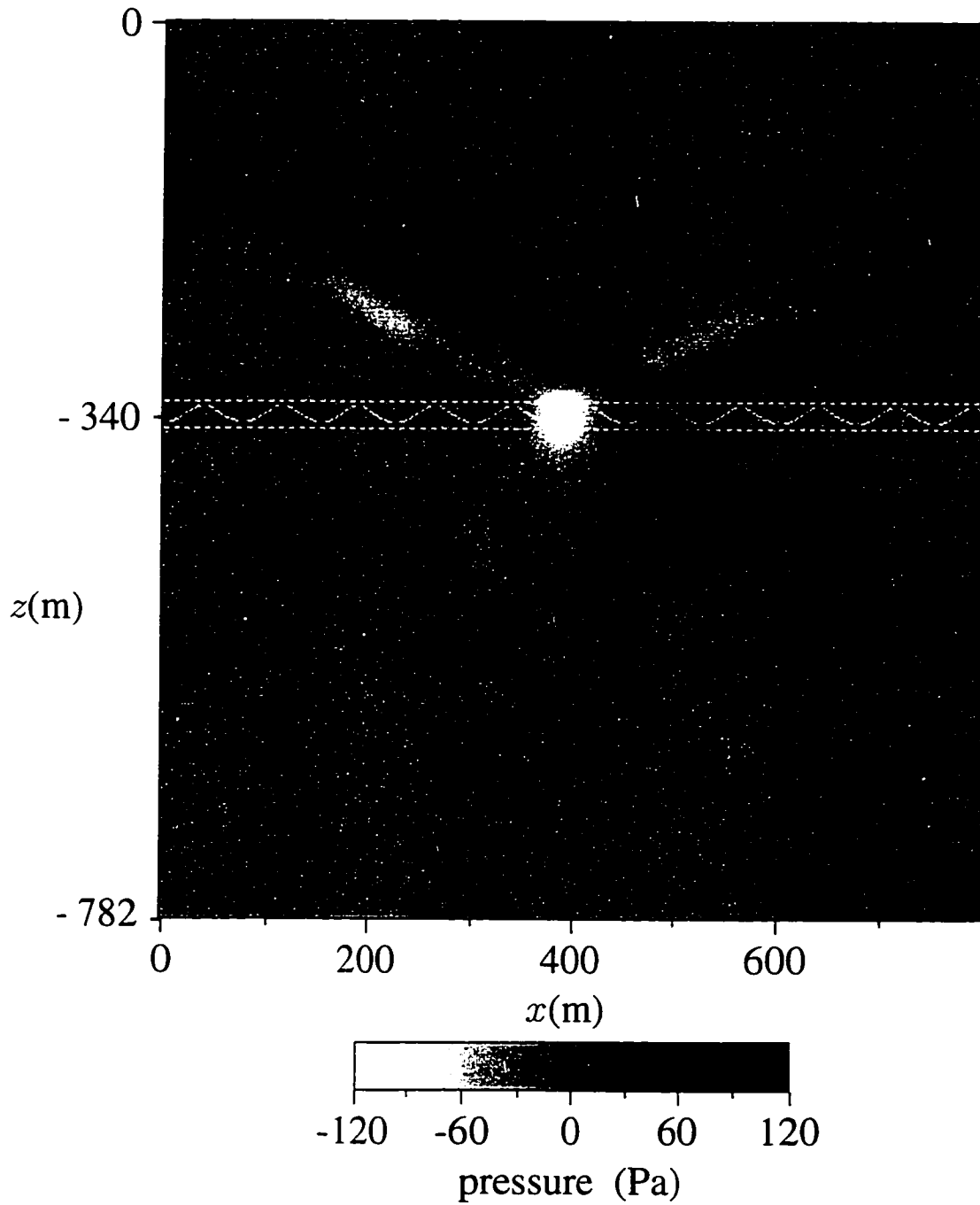
Figure 7.15: Mach number 2.4, wave height 3.75 m, trochoidal surface profile: incident wave interacting with the air-water interface. (Inverted gray scale)

Figure 7.16: Mach number 2.4, wave height 3.75 m, trochoidal surface profile, pressure and underwater intensity fields; incident wave interacting with the air-water interface.

pressure and intensity; M=2.4; wh=3.75 m; #160
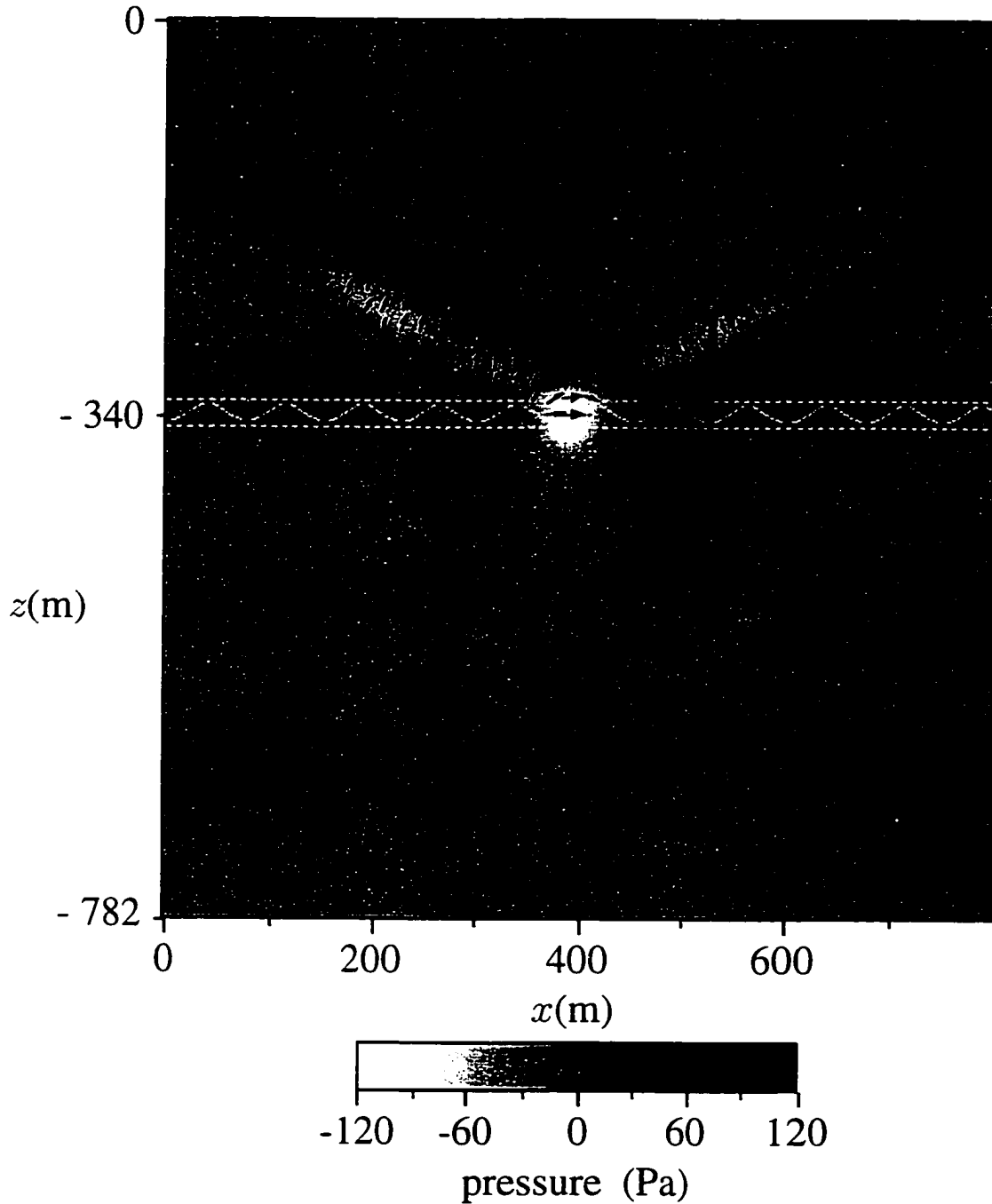
Figure 7.17: Mach number 2.4, wave height 3.75 m, trochoidal surface profile, pressure and air intensity fields; incident wave interacting with the air-water interface.

in the troughs represented by the large arrows superimposed on the light area where the negative peak of the incident wave interacts with the ocean surface. Also seen is a decrease in intensity as the waveform travels over an ocean peak: this is represented by the small arrows superimposed on the dark area where the positive peak of the incident wave interacts with the ocean surface.

The underwater pressure values, which will be more closely scrutinized in the next section, in general indicate that indeed there is a pressure build-up in the troughs which is matched by this higher pressure amplitude just under the surface. This is termed here a focusing effect. Also, the pressure values just under the ocean surface in the ocean wave peaks are actually minimized by a defocusing effect. Figure 7.18 illustrates where these focusing (which causes an increase in amplitude) and defocusing (which causes a decrease in amplitude) effects occur.
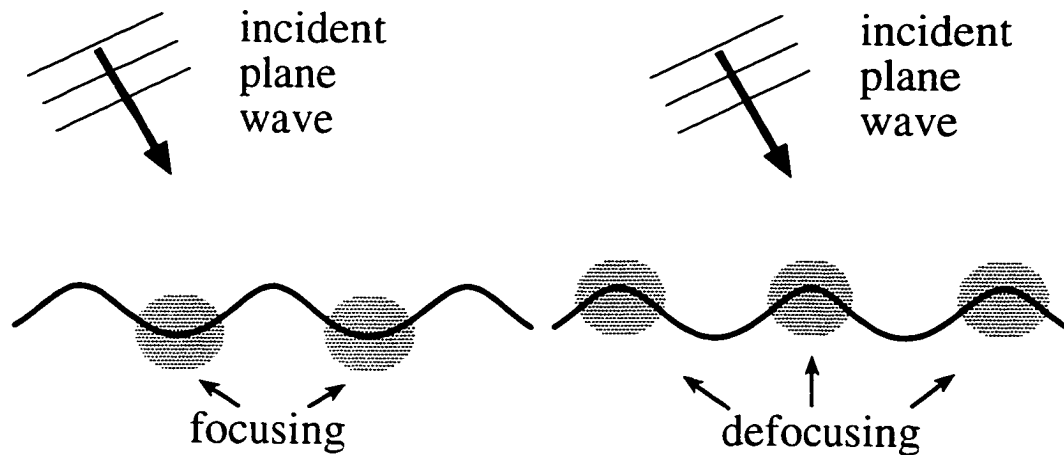
Figure 7.18: Focusing and defocusing of a wave incident on an ocean surface with swell. The shaded regions represent the regions of focusing and defocusing. This picture is not intended to illustrate a sonic boom interacting with the ocean surface, but rather just to show where focusing and defocusing occurs.

In addition to studying the snapshots in time, an animation of the Mach 2.4/wave height 3.75 m/trochoidal interface profile simulated data was created as a research tool.

Observing this animation, it was seen that the positive and negative peaks of the rounded sonic boom waveform just above the ocean surface increased (in magnitude) in the troughs of the ocean waves and decreased (in magnitude) over the ocean wave crests. The highest pressure values were located near the bottom of the trough and up the right-hand side slope of the trough. Recall that the airborne sonic boom is traveling from left to right over the ocean surface.

This animation was actually created in color, not in gray scale. A blue/red/yellow color palette represents the pressure values. Blue is for negative pressure, red for positive pressure, and yellow for high amplitude pressures, either positive or negative. Yellow bands are used to highlight the pressure amplitudes which exceed approximately 96.5 Pa. Three chronological snapshots of the animation are now shown to give the reader some indication of the previously described observations and the enhanced meaning of the pressure field by applying the blue/red/yellow color palette. Figure 7.19 is 0.75 seconds into the animation, the snapshot shown in the previous figures of this section: it shows the negative peak of the sonic boom waveform having a large pressure amplitude in a trough of the ocean waves and the positive peak not reaching any high amplitude pressure values when it is stretched over an ocean wave crest. Figure 7.20 is at time $t = 0.77$ s: here, the negative peak of the sonic boom waveform is leaving an ocean wave trough, and the positive peak is entering one, the negative peak decreasing in pressure amplitude and the positive peak increasing. Figure 7.21 is at time $t = 0.79$ s: this last graphic shows a high pressure amplitude for the positive peak which is fully in an ocean wave trough and the absence of high pressure amplitudes for the negative sonic boom waveform peak which is now stretched over an ocean wave crest.

7.2.3.2. Quantitative description of effects

The first quantitative look at the simulated data involves pressure values just under the ocean surface. Varying the Mach number, ocean wave height, and ocean surface profile, trends are noticed in the effects of the ocean curvature on the impinging waveform.

For each simulation, five to ten snapshots in time are evaluated, the number of times dependent on ocean wave height. Over the time it takes for the sonic boom to finish interacting with the ocean, the positive and negative peaks of the waveform have each

pressure; M=2.4; wh=3.75 m; #160
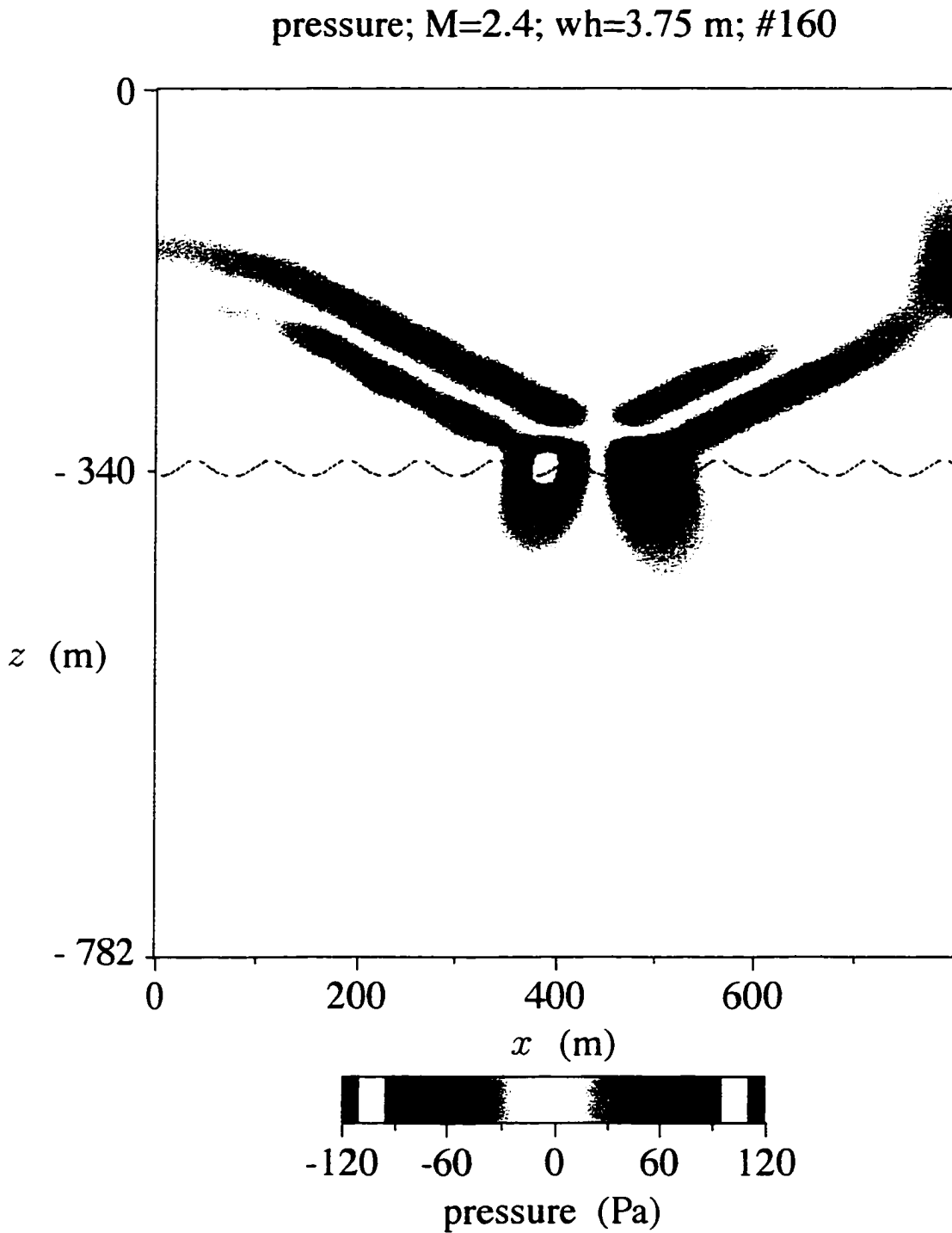
Figure 7.19: Mach number 2.4, wave height 3.75 m, trochoidal surface profile, pressure field, blue/red/yellow color palette; snapshot of animation at time $t = 0.75$ s.

pressure; M=2.4; wh=3.75 m; #164
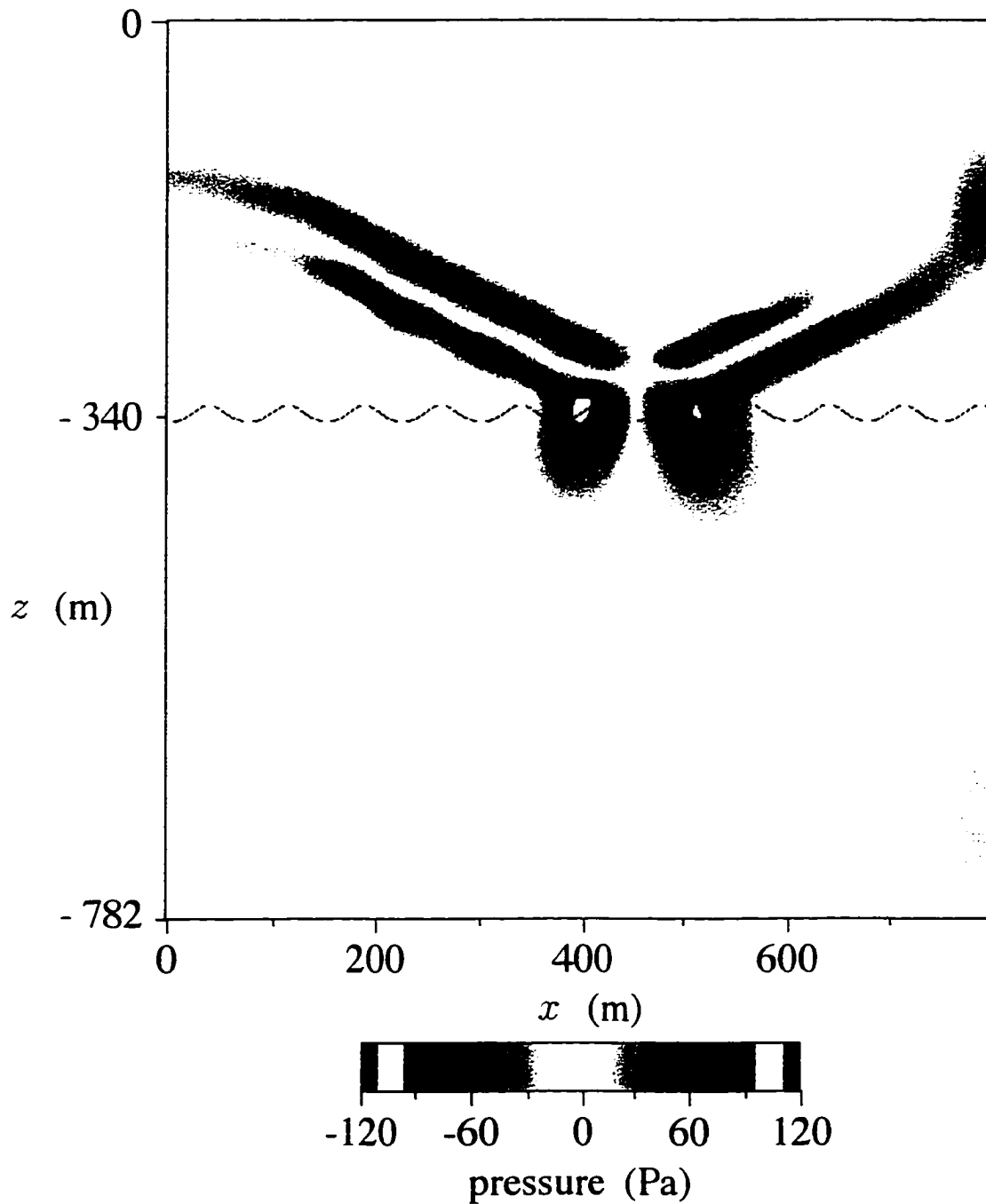
Figure 7.20: Mach number 2.4, wave height 3.75 m, trochoidal surface profile, pressure field, blue/red/yellow color palette; snapshot of animation at time $t = 0.77$ s.

pressure; M=2.4; wh=3.75 m; #168
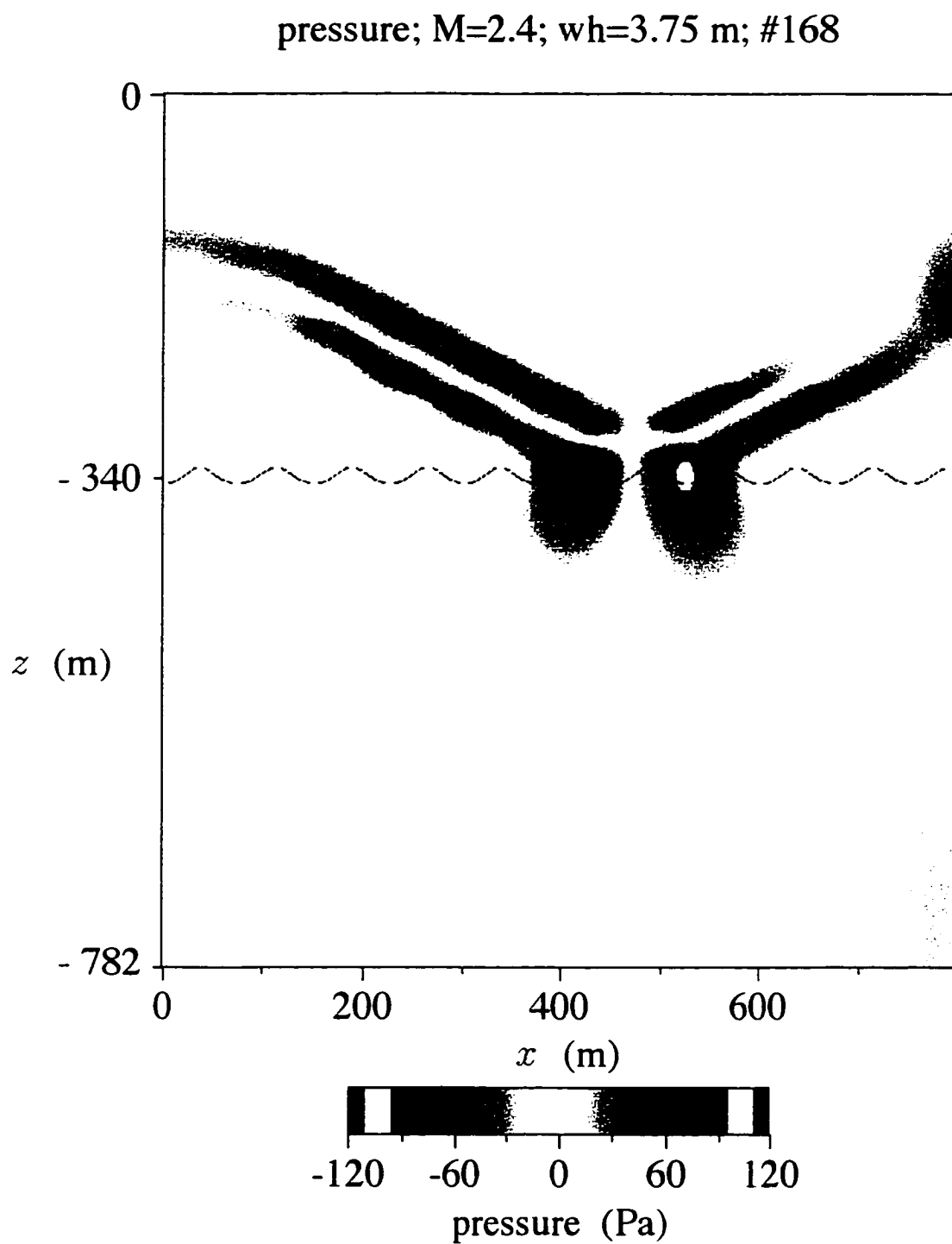
Figure 7.21: Mach number 2.4, wave height 3.75 m, trochoidal surface profile, pressure field, blue/red/yellow color palette; snapshot of animation at time $t = 0.79$ s.

interacted with several crests and troughs of the ocean surface swell. The times chosen for evaluation span a period which includes the time it takes for the sonic boom waveform to sweep across one wavelength of the surface swell: any time before or beyond this period does not need to be evaluated because the results are repetitive. The number of snapshots in time is chosen such that the sonic boom waveform shift is small enough to properly approximate the curvature of the surface—the larger the wave height. the steeper the curves, the greater the number of snapshots. What is meant by the sonic boom waveform shifts approximating the ocean surface curvature is shown in Fig. 7.22. Each dot represents the middle of the sonic boom waveform (zero pressure) as it sweeps over the ocean surface. If the dots are connected in either the large wave height or small wave height case, the resulting curve is a good approximation to the ocean surface.

10 snapshots in time   5 snapshots in time

large wave height   small wave height
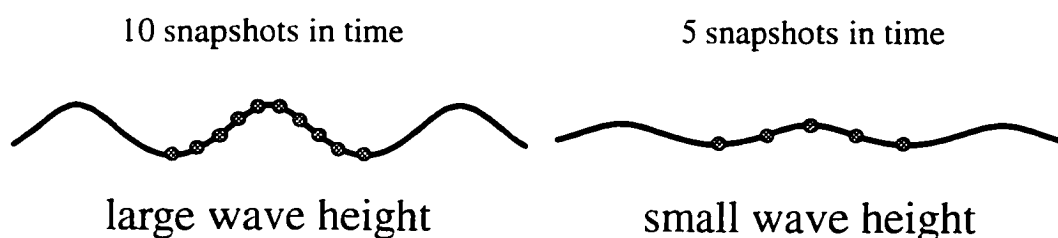
Figure 7.22: Dots represent the middle of the sonic boom waveform as it sweeps across the ocean surface: a large ocean wave height and small ocean wave height are represented.

Depending on where on the curved interface the sonic boom wave is incident, the peak pressure values just under the surface may be either increased or decreased by varying amounts. For each snapshot in time the most increased or decreased positive and negative pressure values are extracted from just under the wavy ocean surface. These values are then compared to the largest pressure amplitudes found just under a flat ocean surface. The percent change from a flat ocean surface due to the curvature on the wavy ocean surface is calculated as $\% = 100 \times (p_{wavy} - p_{flat})/p_{flat}$. Now that it is explained how the underwater pressure is evaluated, the effects found with each set of parameters are determined.

The effects caused by a sonic boom interacting with an ocean interface with swell can be divided into two main categories: 1.) how different wind wave heights strengthen the focusing and defocusing of the evanescent acoustic pressure and 2.) how different Mach numbers strengthen the focusing and defocusing.

For the first category. numerical results were extracted from the Mach 1.4 and Mach 2.4 runs for all wave heights. Tables 7.2 and 7.3 contain typical percent values reflecting the change in positive and negative peaks (in the pressure waveform just under the ocean surface) from the flat ocean result to each wavy ocean result. These numbers are taken from the sinusoidal interface calculations for Table 7.2 and the trochoidal interface calculations for Table 7.3.

Table 7.2: Percent change in peak pressure from the flat ocean results due to focusing and defocusing caused by ocean swell (sinusoidal waves): Mach 1.4 and 2.4. all ocean wave heights.

wind wave height (m) (crest to trough)	change in $+/-$ pressure peaks due to curvature effects	
	M=1.4	M=2.4
0	0 / 0	0 / 0
1.0	0.5% / 0.1%	0.6% / 0.6%
1.4	-0.5% / -0.2%	1.6% / -1.0%
2.3	-1.4% / 2.1% 1.6% / 1.5%	4.1% / 5.3% 1.4% / -3.8%
3.75	-2.8% /-2.8% 2.0% / 5.6%	-5.0% / 11.0% 8.0% / -9.6%

For Tables 7.2 and 7.3. each set of positive and negative peak percent changes is calculated from the numerical pressure values at a single time during the wave propagation simulation. For a particular set of parameters, snapshots in time are chosen out of the set of 5–10 snapshots, the time associated with the largest percent change. Because the percent increase or decrease varies considerably for the 2.3 m and 3.75 m wind wave heights, two typical results are shown for each, the other wave heights showing only one result.

Table 7.3: Percent change in peak pressure from the flat ocean results due to focusing and defocusing caused by ocean swell (trochoidal waves). Mach 1.4 and 2.4. all ocean wave heights.

wind wave height (m) (crest to trough)	change in +/− pressure peaks due to curvature effects	
	M=1.4	M=2.4
0	0 / 0	0 / 0
1.0	−0.4% / −0.1%	0.4% / 0.6%
1.4	0.8% / 0.5%	1.7% / −0.7%
2.3	1.5% / 2.4% −1.4% / 2.5%	−2.8% / −2.8% 4.2% / 5.3%
3.75	0.1% /−3.4% 1.3% / 5.0%	−5.4% / 11.7% 7.8% / −8.9%

Figure 7.23 also shows the results from the simple curvature simulations. The absolute value of the highest percent change due to the curvature is extracted from Tables 7.2 and 7.3 and is plotted as a function of wave height. This plot indicates that the sinusoidal results are very similar to the trochoidal results: later in this work. only the trochoidal surface will be applied because of this similarity.

The percent change values in Tables 7.2 and 7.3 and Fig. 7.23 indicate that for larger ocean wave heights the pressure values are more affected. either being decreased or increased. For a calm ocean (wave height being 1 m). the deviation in peak pressure from the perfectly flat ocean result is less than 1%. whereas for an ocean with significant swell (wave height being 3.75 m). changes can be as large as 5.6% for the Mach 1.4 case or 11.7% for the Mach 2.4 case. The evaluation of the most extreme case. Mach 2.4. ocean wave height 3.75 m. trochoidal surface profile. is useful in determining what exactly the 11.7% change means. As seen in Table 7.4. the 11.7% is an 11.4 Pa difference in peak pressure: this corresponds to only a 1.0 dB difference in sound pressure level. an amount not usually perceptible to humans.

The second category regards how different Mach numbers strengthen the focusing and defocusing of pressure values just under the surface. Because the sinusoidal and trochoidal
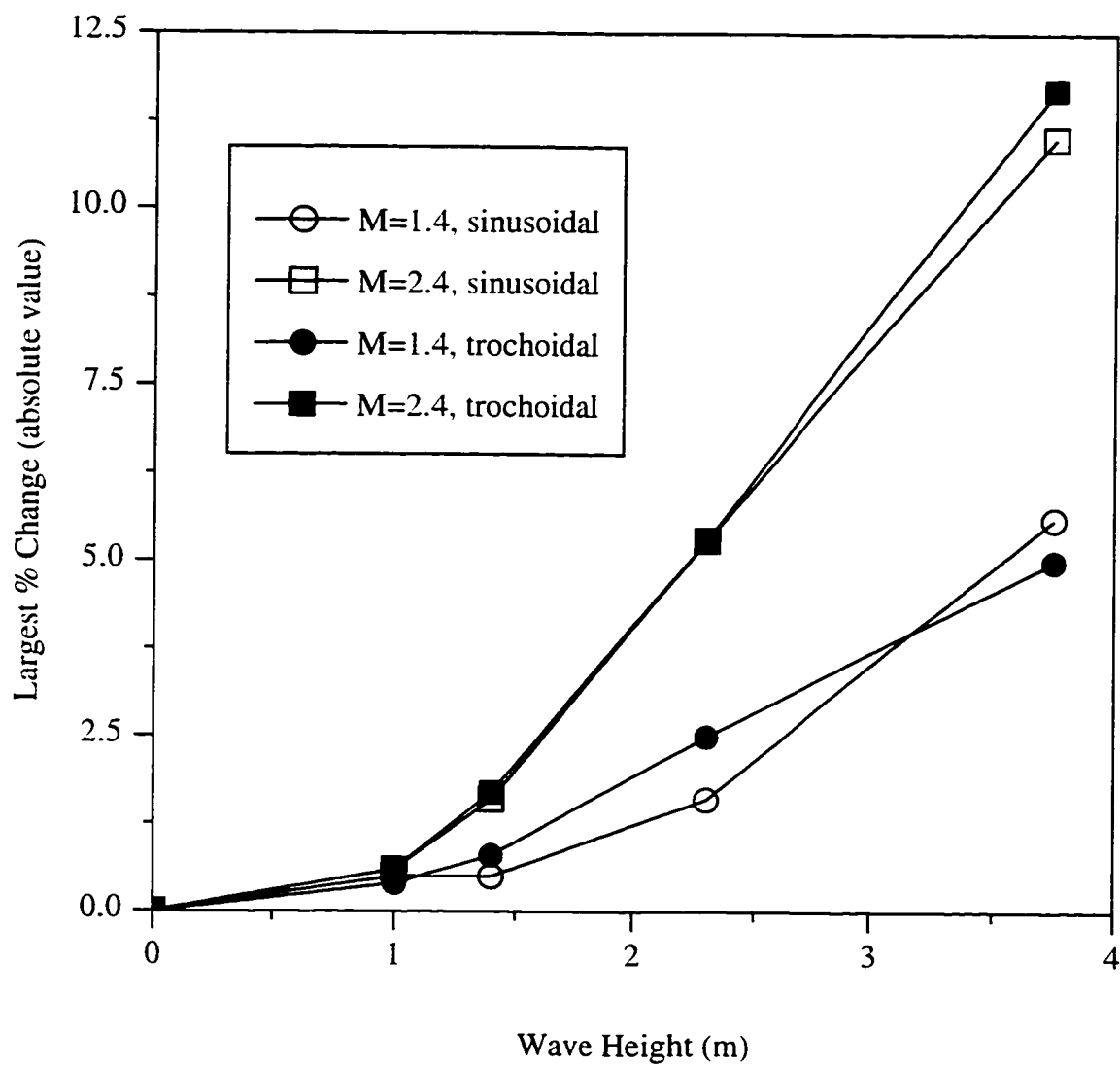
Figure 7.23: Largest amplitude percent change from a simple ocean swell to a flat surface as a function of wave height.

Table 7.4: Change in sound level (dB): most extreme change (−5.4%/11.7% from Table 7.3) due to curvature (trochoidal waves): Mach 2.4, ocean wave height of 3.75 m.

surface type	peak pressure (Pa)		peak dB re 1μPa	
	pos. peak	neg. peak	pos. peak	neg. peak
curved surface	92.42	-109.23	156.3	157.8
flat surface	97.73	-97.83	156.8	156.8
difference (curved - flat)	-5.31	-11.40	-0.5	1.0

interfaces yield very similar results, this second category describes the effects of the curvature using only the trochoidal surface profile. Runs include all Mach numbers but only the most extreme wave height. 3.75 m. Table 7.5 shows the percent changes in pressure values just under the ocean surface (from an incident plane wave interacting with a flat surface to one interacting with a wavy surface). Only cases with the greatest percent changes are shown. In viewing the data in Table 7.5. it can be seen that the highest percent changes increase with increasing Mach number.

Table 7.5: Percent change in peak pressure from the flat ocean results due to focusing and defocusing caused by ocean swell (trochoidal waves). All Mach numbers. ocean wave height of 3.75 m.

wind wave height (m) (crest to trough)	change in +/− pressure peaks due to curvature effects			
	M=1.4	M=2.4	M=3.0	M=3.5
0	0 / 0	0 / 0	0 / 0	0 / 0
3.75	1.3% /5.0%	−5.4% / 11.7%	5.1% / 15.4%	8.6% / 16.9%

The second quantitative look at the simulated data involves pressure values as a function of depth. The simulation used in this analysis includes the parameters Mach 2.4, ocean wave

height 3.75 m, and a trochoidal surface profile. A single snapshot in time is analyzed, this time showing the strongest effects of the curvature just under the surface (the snapshot which gives the 11.7% percent change from the flat water results as seen in Table 7.3). The peak positive and peak negative pressure values of the penetrating sonic boom waveform are extracted at depths of 4, 8, 16, 32, 64, and 128 m and just under the surface (0 m). These results are shown in Table 7.6 along with the percent change of the pressure values from those found in the Mach 2.4/flat ocean surface simulation. Just under the ocean surface the percent change from the flat ocean results is at most 11.7% and at 4 m, 12.8%; at depths of 64 or 128 m the change is 2% or less. These results indicate that the underwater pressure field closer to the surface is more affected by the ocean curvature than the pressure field at greater depths.

Table 7.6: Pressure values at several depths for the Mach 2.4, wave height 3.75 m simulation: differences from flat ocean surface results.

depth (m)	Mach 2.4 pos. peak. neg. peak pressure (Pa)	difference from flat surface +/− peaks	
		difference in Pa	% difference
0	92.42, -109.23	-5.31 / -11.40	−5.4% / 11.7%
4	83.68, -101.20	-5.86 / -11.49	−6.5% / 12.8%
8	77.10, -90.67	-4.57 / -8.67	−5.6% / 10.7%
16	65.71, -73.95	-2.54 / -5.35	−3.7% / 7.8%
32	47.60, -51.28	-0.90 / -2.21	−1.9% / 4.5%
64	26.34, -27.72	0.13 / -0.40	0.5% / 1.5%
128	9.06, -12.08	-0.14 / -0.24	−1.5% / 2.0%

7.2.4. Discussion of results

Results indicate that as a sonic boom interacts with the ocean surface with swell, its energy is being focused in the troughs and defocused over the crests. Two trends can be identified when extracting pressure values from just under the ocean surface. The first trend shows that the effects caused by the curvature are stronger for greater ocean wave heights, the largest percent change from the flat surface results being 11.7% in the Mach 2.4 case: this corresponds to a 1 dB increase. The second trend is that the highest percent changes increase with increasing Mach number. It should be noted that this trend was found when assuming the sonic boom has a duration of 300 ms. A section later in this chapter will address the issue of sonic boom duration and its corresponding effective wavelength compared to the ocean surface wavelength. When extracting pressure values at various depths, it is shown that the underwater pressure closer to the surface is more affected by the curvature than the pressure at greater depths. Excerpts from results in this section are also found in Rochat and Sparrow.[52]

7.3. Homogeneous Ocean with a Complex Curved Surface: Computational Simulations

The second set of computational simulations for a wavy ocean surface includes the two complex surface profiles, a trochoid plus a half-wavelength sinusoid (complex combination 1) and a trochoid plus a third-wavelength sinusoid (complex combination 2). These profiles are implemented to investigate the effects of waves with more complex structure: they are crude approximations to a fully developed sea. A necessary addition to the computer code and the physical parameters used in these simulations will be described. Also, the effects caused by the ocean curvature will be quantified and discussed.

7.3.1. Computer program

The same computer program that was used for the simple curved ocean surface is used here. However, the curvature of the complex combination 1 profile lead to instabilities independent of the ones found previously for the simple curved surfaces. As a result, an additional filter is applied to this profile, but not to the complex combination 2 profile. Again the instability is due to the grid-block approximation of the surface curve. Figure 7.24 illustrates the problem grid points and the application of this additional filter; again, this illustration does not give accurate dimensions, and the raggedness is greatly exaggerated. Running the program with the "smoothing" filter (described in the previous section) and this new filter allows for successful calculations for a sonic boom interacting with a complex curved surface.

Figure 7.24: Illustration of the additional filter applied to the complex ocean surface curve.

7.3.1.1. Physical parameters

As stated before, the wavy ocean simulations propagate a rounded sonic boom waveform. The initial waveform inserted in the computational domain is the same as in Chapter 6 and the previous section: the peak pressure of the sonic boom is 50 Pa and the duration 300 ms. The two Mach numbers used are 1.4 and 2.4.

The ocean surface profiles chosen to represent waves with more complex structure than swell are a trochoid plus a half-wavelength sinusoid (complex combination 1) and a trochoid plus a third-wavelength sinusoid (complex combination 2). Please refer to Sect. 7.1.1.2 for the equations used to calculate these surfaces and their illustrations; specifically Eq. (7.3)

and Fig. 7.4 for complex combination 1 and Eq. (7.4) and Fig. 7.5 for complex combination 2. The two ocean wave heights used in these simulations are 2.3 m and 3.75 m (see Table 7.1 for corresponding wind speeds): the smaller wave heights are not implemented because it was found with the simple curved surface results that the pressure change due to the curvature associated with these wave heights is trivial.

As in the simple curved surface simulations, the air and the ocean are homogeneous for the complex surface simulations. In the air the speed of sound c_o is 343 m/s and the density ρ_o is 1.21 kg/m^3. In the ocean $c_o = 1500$ m/s and $\rho_o = 1000$ kg/m^3.

7.3.2. Program runs

The same computational parameters (grid spacing, number of grid points, ...) are used in the curved surface simulations as were used in the flat surface simulations in Chapter 6 and for the simple curved surface simulations. In addition to the two complex ocean surface profiles, two physical parameters are varied: the Mach number and the ocean wave height. The Mach 1.4 and 2.4 cases are run for wave heights of 2.3 and 3.75 m. Again, to find out the ocean curvature effects, the runs mentioned in Chapter 6 for a flat ocean surface for the Mach 1.4 and 2.4 cases are used for comparison.

7.3.3. Effects caused by a complex curved ocean surface

7.3.3.1. Qualitative description of effects

A sonic boom interacting with the complex combination 1 ocean surface is seen in Fig. 7.25. As in previous graphics of the full computational domain, the black dashed lines enclose the refined grid region, the plot appearing stretched out in that area. Also, the black curvy line is the ocean surface. A blue/red/yellow color palette is chosen to represent the pressure values: blue represents negative pressure, red positive pressure, and yellow high pressure amplitudes. What is apparent in Fig. 7.25 is that, as in two of the color plots of the trochoidal ocean surface (Figs. 7.19 and 7.21) the highest pressure amplitudes are found in the troughs of the ocean waves. In this complex case, the trochoidal surface troughs are

divided into two smaller troughs by the lower amplitude sinusoid. The build-up effect of the pressure is divided by the sinusoid peak. creating two areas of high amplitude in the smaller troughs. Figure 7.25 shows the high amplitude pressure of the negative peak of the sonic boom in one of the smaller troughs.

A sonic boom interacting with the complex combination 2 ocean surface is seen in Fig. 7.26. Again. a blue/red/yellow color palette is chosen to represent the pressure values. As before. the highest pressure amplitudes are found in the troughs of the ocean waves. Figure 7.26 shows the positive peak of the sonic boom waveform having high amplitude pressure values in a trough of the ocean surface. In this complex case, the troughs created by the trochoid seems to govern the areas of high amplitude. while the small amplitude sinusoid only quantitatively affects the impinging waveform (described in next section).

7.3.3.2. Quantitative description of effects

A quantitative look at the simulated data examines pressure values just under the ocean surface. Numerical results were extracted from the Mach 1.4 and Mach 2.4 runs for the two larger wave heights. As for the simple curved surface results. several snapshots in time were evaluated, and the most significant results are presented here. Tables 7.7 and 7.8 contain typical percent values reflecting the change in positive and negative peaks (in the pressure waveform just under the ocean surface) from the flat ocean result to each wavy ocean result. These numbers are taken from the complex combination 1 interface calculations for Table 7.7 and the complex combination 2 interface calculations for Table 7.8.

For both complex surface combinations. the higher wave height causes greater changes in pressure: this is consistent with the trend found for the simple curved surfaces. However, it is obvious from looking at Tables 7.7 and 7.8 that the second trend observed for the simple curved surface (higher Mach numbers result in higher pressure changes due to curvature) is not true for the complex surfaces. The greatest percent change for complex combination 1 is the same for the Mach 1.4 and 2.4 cases, 11.2%. For complex combination 2, the greatest percent change for Mach 1.4 is 8.5% and for Mach 2.4, 8.3%.

pressure; M=2.4; complex combo #1; #160
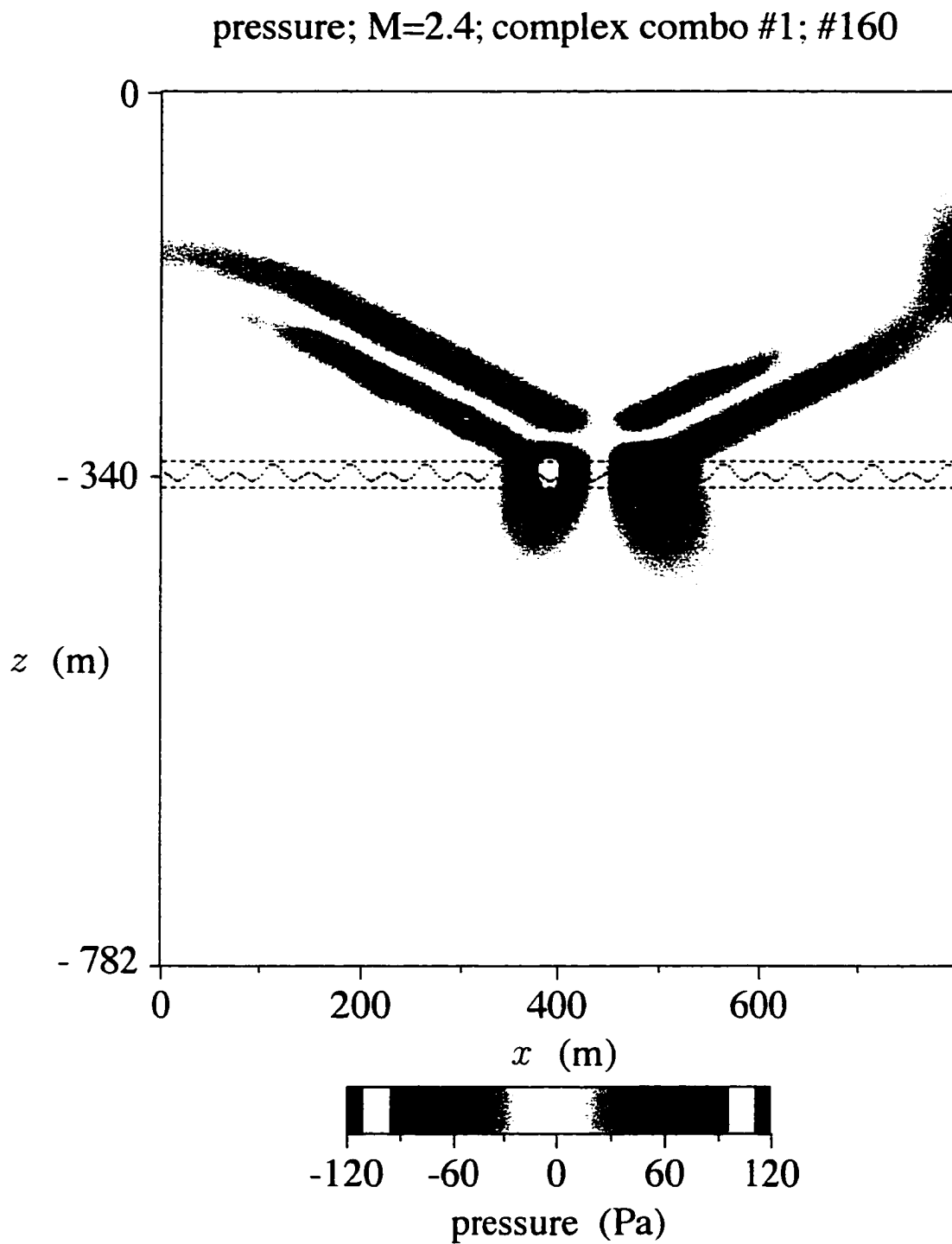
Figure 7.25: Mach number 2.4, wave height 3.75 m, complex combination 1 surface profile, pressure field; incident wave interacting with the air-water interface.
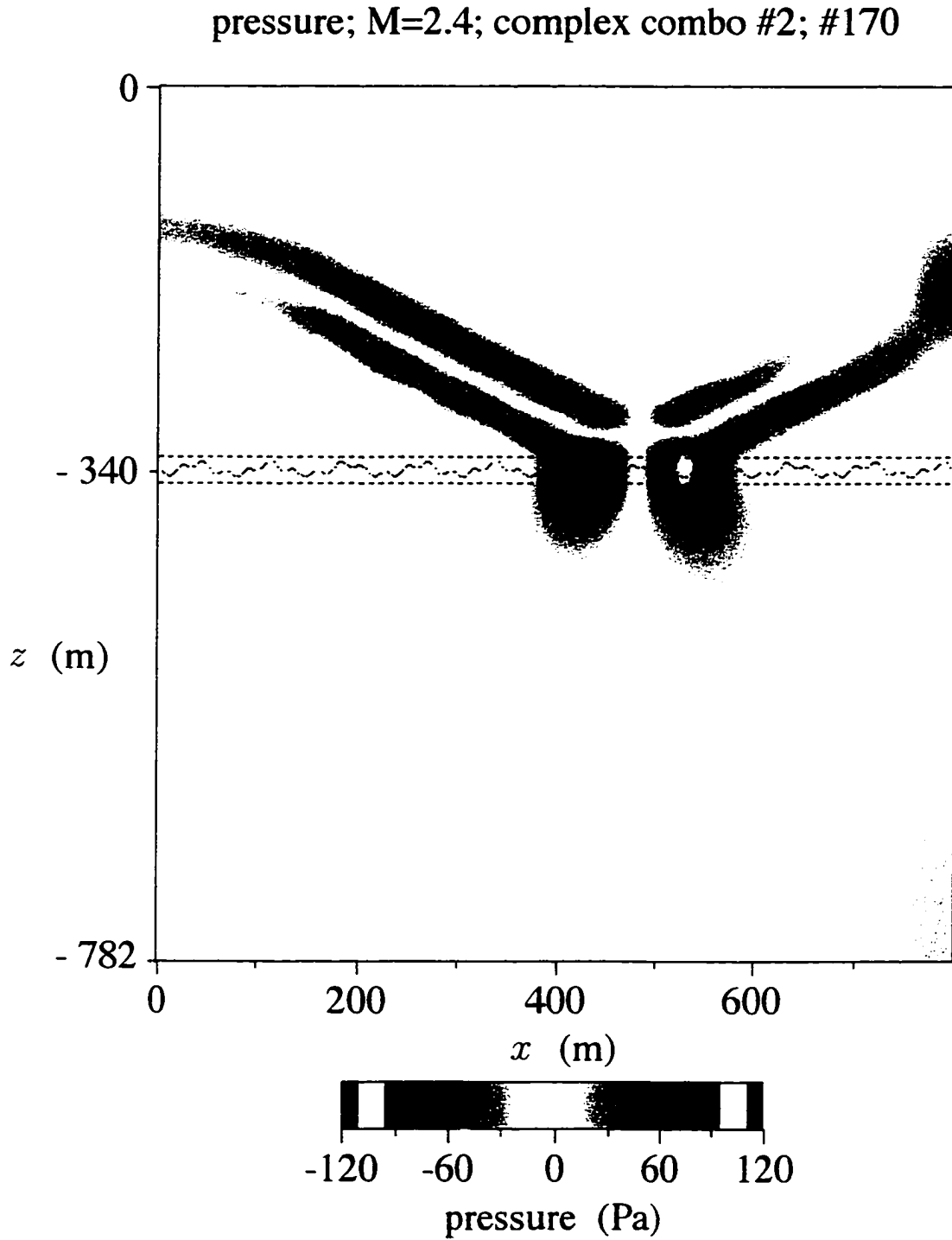
Figure 7.26: Mach number 2.4, wave height 3.75 m, complex combination 2 surface profile, pressure field; incident wave interacting with the air-water interface.

Table 7.7: Percent change in peak pressure from flat ocean results due to focusing and defocusing caused by a complex surface (combination 1): Mach 1.4 and 2.4, ocean wave heights: 2.3 and 3.75 m.

wind wave height (m) (crest to trough)	change in +/− pressure peaks due to curvature effects	
	M=1.4	M=2.4
0	0 / 0	0 / 0
2.3	1.9% / 6.5% −1.6% / 7.9%	3.5% / 1.8% 1.9% / 3.3%
3.75	−0.3% / 11.2% 4.1% / −0.3%	5.6% / −1.1% −5.0% / 11.2%

Table 7.8: Percent change in peak pressure from flat ocean results due to focusing and defocusing caused by a complex surface (combination 2): Mach 1.4 and 2.4, ocean wave heights: 2.3 and 3.75 m.

wind wave height (m) (crest to trough)	change in +/− pressure peaks due to curvature effects	
	M=1.4	M=2.4
0	0 / 0	0 / 0
2.3	0.3% / 7.5% 1.3% / 0.7%	2.3% / 2.8% 0.7% / 3.4%
3.75	0.7% / −6.4% 1.3% / 8.5%	−5.4% / 8.3% 5.4% / −5.9%

7.3.4. Discussion of results

As was found for the simple ocean swell, results indicate that as the sonic boom interacts with the complex wavy ocean surface, its energy is being focused in the troughs and defocused over the crests. Effects caused by the curvature are stronger for greater ocean wave heights, consistent with the trend found for the simple curved surface. The largest percent change in peak pressure was 11.2%; this corresponds to less than 1 dB. Results indicate, however, that the complex numerical data do not follow the other simple ocean swell trend, that the highest percent changes increase with increasing Mach number. Inspired

by this inconsistency, research was conducted to determine the cause; the following section reveals the results of this investigation.

7.4. Wavelength Comparisons: Computational Simulations

7.4.1. Wavelengths

As was explained in Chapter 2, the term *effective wavelength* describes the physical length associated with the sonic boom duration. For all previous runs the sonic boom duration T was 300 ms; this is associated with a sonic boom effective wavelength λ_{boom} of 102.9 m in air ($\lambda_{boom} = c_{air}T$). The results for the simple and complex ocean profiles indicate that the positive and negative peaks of the sonic boom waveform interact with the ocean waves separately; while one peak is in a trough of the ocean surface the other peak is either on the far side of the neighboring peak or in the next trough. Hence, looking at just one peak of the sonic boom is warranted. The wavelength associated with half the duration of the sonic boom is $\lambda_{\frac{1}{2}boom} \approx 51.5$ m. Since the ocean surface is horizontal in the computational domain, the horizontal component of $\lambda_{\frac{1}{2}boom}$, $\lambda_{\frac{1}{2}boom,horiz}$, is applied when comparing the sonic boom effective wavelength to the ocean surface wavelength. The length of the horizontal component is Mach number dependent. Figure 7.27 illustrates the sonic boom effective wavelength variables. Table 7.9 gives effective wavelengths for the full and half sonic boom and the horizontal components of the full boom and half sonic boom for the Mach 1.4 and Mach 2.4 cases.

For a simple trochoidal ocean surface profile the wavelength λ_{ocean} is taken to be twenty times the wave height. Table 7.10 gives the wavelengths for the ocean wave heights of 2.3 m and 3.75 m. Also shown in Table 7.10 are the largest and smallest wavelength components used for each of the complex ocean surface profiles and each of the wave heights. Recall that the complex combination 1 surface is a trochoidal wave just like the simple surface one but with a lower-amplitude, half-wavelength sinusoid superimposed. Also the complex
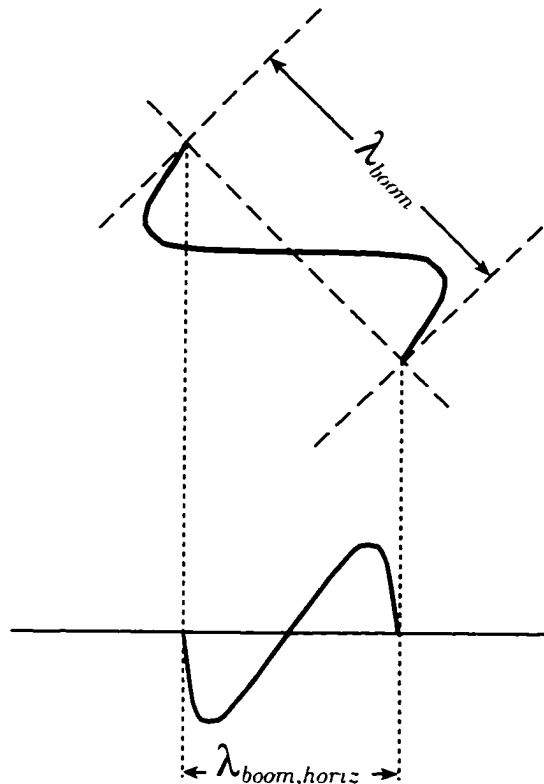
Figure 7.27: Illustration of the sonic boom effective wavelength variables.

Table 7.9: Sonic boom effective wavelengths.

	effective wavelength	horizontal component of effective wavelength	
		M=1.4	M=2.4
full sonic boom	$\lambda_{boom} = 102.9$ m	$\lambda_{boom,horiz} = 73.5$ m	$\lambda_{boom,horiz} = 42.8$ m
half sonic boom	$\lambda_{\frac{1}{2}boom} = 51.5$ m	$\lambda_{\frac{1}{2}boom,horiz} = 36.8$ m	$\lambda_{\frac{1}{2}boom,horiz} = 21.4$ m

combination 2 surface is the trochoid with a lower-amplitude, third-wavelength sinusoid superimposed.

As a way of comparing the sonic boom effective wavelength with the ocean wavelength the following ratio is defined: λ-Ratio $= \lambda_{\frac{1}{2}boom,horiz}/\lambda_{ocean}$. Given in Table 7.11 are values

Table 7.10: Wavy ocean wavelengths λ_{ocean}. H is the ocean wave height.

ocean surface profile	largest wavelength component		smallest wavelength component	
	$H = 2.3$ m	$H = 3.75$ m	$H = 2.3$ m	$H = 3.75$ m
trochoidal	46.0 m	75.0 m	46.0 m	75.0 m
complex combo 1	46.0 m	75.0 m	23.0 m	37.5 m
complex combo 2	46.0 m	75.0 m	15.3 m	25.0 m

for λ-Ratio for the trochoidal and complex surfaces, the smallest wavelength component used for λ_{ocean}. The values for $\lambda_{\frac{1}{2}boom.horiz}$ assume that the sonic boom duration is 300 ms. The trend found for the simple surface profiles was that higher Mach numbers are associated with greater effects caused by the curvature of the wavy surface. As seen in Table 7.11, the values of λ-Ratio where that trend was seen are all less than one. It was thought that perhaps the trend holds only for λ-Ratio < 1 and not for λ-Ratio ≈ 1 or λ-Ratio > 1. A systematic approach was developed in order analyze the situation.

Table 7.11: λ-Ratio $= \lambda_{\frac{1}{2}boom.horiz}/\lambda_{ocean}$ for trochoidal and complex surfaces: the smallest wavelength component of the complex ocean surface is used for λ_{ocean}.

ocean surface profile	λ-Ratio			
	M = 1.4		M = 2.4	
	$H = 2.3$ m	$H = 3.75$ m	$H = 2.3$ m	$H = 3.75$ m
trochoidal	0.80	0.49	0.47	0.29
complex combo 1	1.60	0.98	0.93	0.57
complex combo 2	2.41	1.47	1.40	0.86

7.4.2. Computer program, physical parameters, and program runs

The computer programs used for the simulations in this section are those for a tro-choidal ocean profile. Again. a rounded sonic boom with a peak pressure of 50 Pa is the initial waveform. For these runs. however, the duration of the sonic boom varies. Varying the duration allows a systematic evaluation of the wavelength ratio λ-Ratio and the corre-sponding effects. The durations are chosen such that λ-Ratio ranges from 0.125 to 1.875 in steps of 0.125: this adds up to 15 ratios or durations. Due to the size of the computational domain the runs are limited: not all cases (combinations of Mach 1.4 and 2.4 and ocean wave heights 2.3 m and 3.75 m) were run to completion. The only wave height applied is 2.3 m since the duration of the sonic boom would have to be quite long in order to accommodate the higher wavelength ratio values for the ocean wave height of 3.75 m. The Mach 1.4. wave height 2.3 m case is run for all wavelength ratios. The Mach 2.4. wave height 2.3 m case is run for all wavelength ratios less than or equal to one: the higher wavelength ratios cannot be applied because of computational limits. Table 7.12 gives the values of λ-Ratio and the corresponding durations. It should be noted that not all of these sonic boom durations rep-resent realistic sonic booms: this exercise is purely for the understanding of the wavelength variations and the corresponding effect.

7.4.3. Results

As before. for every case run, several snapshots in time are evaluated. The largest modification of the pressure values just under the surface are then turned into percent change from the flat ocean results. Figure 7.28 shows the results of this wavelength comparison study. It is a plot of the largest amplitude percent change from a flat surface caused by a simple ocean swell for the corresponding wavelength ratio λ-Ratio $= \lambda_{\frac{1}{2}boom.horiz}/\lambda_{ocean}$. The line with dots represents the Mach 1.4 results, and the line with squares represents the Mach 2.4 results: these two curves roughly follow the same path. indicating that for a wavy surface the percent change is governed by the relation between the sonic boom and ocean wavelengths rather than the Mach number. Overall the curves show that the effects of the

Table 7.12: Values of λ-Ratio and sonic boom durations used in program runs.

λ-Ratio	duration of full sonic boom (seconds)	
	M=1.4, $H = 2.3$ m	M=2.4, $H = 2.3$ m
0.125	0.047	0.080
0.250	0.094	0.161
0.375	0.141	0.241
0.500	0.188	0.322
0.625	0.235	0.402
0.750	0.282	0.483
0.875	0.329	0.563
1.000	0.376	0.644
1.125	0.422	—
1.250	0.469	—
1.375	0.516	—
1.500	0.563	—
1.625	0.610	—
1.750	0.675	—
1.875	0.704	—

ocean waviness are more significant when the ocean surface wavelength is larger than the horizontal component of the sonic boom effective wavelength.

Cheng, et al.[8,9] assert that waviness influence will be significant only for swell with wavelengths comparable to or much larger than the sonic boom effective wavelength. The arrow in Fig. 7.28 indicates where the sonic boom effective wavelength (not the horizontal component) is comparable to the ocean wavelength. It is at this point on the plot where the largest percent change is approximately 7%. Sonic boom effective wavelengths larger than the ocean wavelength would be associated with percent changes larger than 7%. So the plot in Fig. 7.28 supports Cheng, et al.'s assertion.

Now it is necessary to go back to the trochoidal ocean profile results and the complex profile results and see where specific values of λ-Ratio fall on the plot and if the percent change predicted by Fig. 7.28 matches with those previously presented in the results tables.
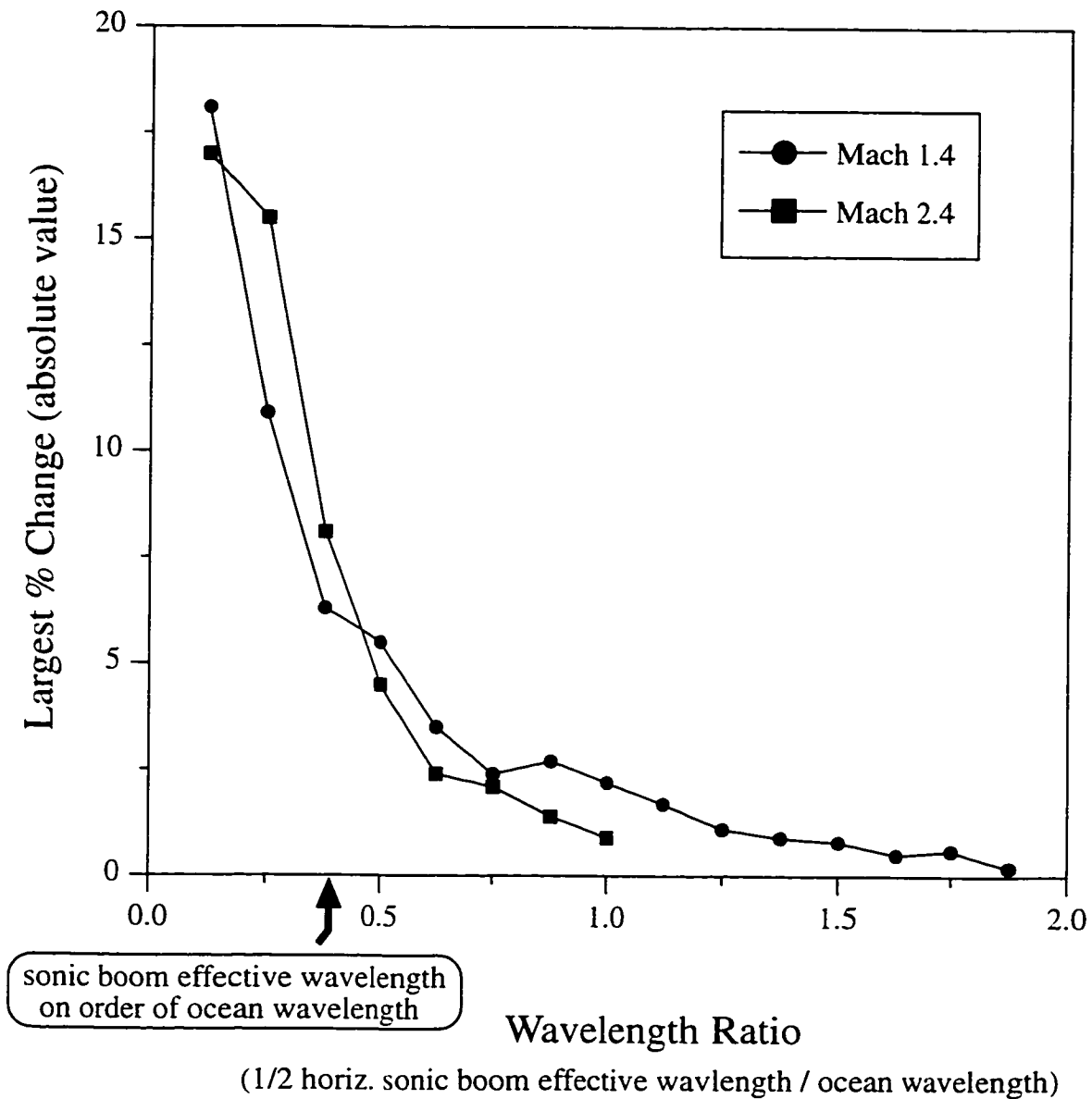
Figure 7.28: Largest amplitude percent change from a simple ocean swell to a flat surface for the corresponding wavelength ratio, λ-Ratio $= \lambda_{\frac{1}{2}boom.horiz}/\lambda_{ocean}$. Mach 1.4: dots, Mach 2.4: squares.

Table 7.13 presents λ-Ratio. the absolute value of the largest percent change found in the simulation, the largest percent change approximated using Fig. 7.28, and the absolute value of the difference between the results obtained in the simulations and those predicted using Fig. 7.28; each of these is presented for the trochoidal surface profile, various Mach numbers, and various wave heights. It should be noted that the largest percent changes extracted from the plot in Fig. 7.28 are approximated by taking the average of the two curves. It is seen from the differences that the approximations using the plot are quite accurate, the largest difference being 1.4%. This indicates that, in fact, when a sonic boom impinges upon an ocean with a simple surface profile, the amount that waveform is affected is governed by the relation between the sonic boom effective wavelength and the ocean surface wavelength.

When studying the trochoidal surface profile back in Sect. 7.2 with wave height 3.75 m and a 300 ms duration sonic boom and applying different Mach numbers, the boom duration was kept constant and the Mach number was varied, unlike the simulations that were just described. By increasing the Mach number, λ-Ratio decreases, and Fig. 7.28 implies that the largest percent change increases. This explains the trend that the higher Mach numbers have a greater effect due to curvature.

Now the complex surface results are presented for the wavelength comparisons. Table 7.14 presents the same information as in Table 7.13. but now λ-Ratio is shown for the largest and smallest components of the complex ocean surface profile; this results in two predictions and two differences between predictions and the simulations. It is obvious from Table 7.14 that the smallest ocean wavelength component does not govern the effects of the curvature nor does the largest ocean wavelength component. Each is off by more than 6% in some cases. It follows that the plot in Fig. 7.28 cannot be used to predict how much the curvature of the complex ocean surface affects the impinging sonic boom.

Table 7.13: Largest amplitude percent changes for simulations and predictions from plot in Fig. 7.28; also the difference between these two results: TROCHOIDAL surface.

Mach number	ocean wave height	λ-Ratio	simulation % change	Fig. 7.28 predicted % change	magnitude of difference in percent
1.4	1.0	1.84	0.4	0.5	0.1
	1.4	1.31	0.8	1.0	0.2
	2.3	0.80	2.5	2.0	0.5
	3.75	0.49	5.0	5.0	0.0
2.4	1.0	1.07	0.6	1.8	1.2
	1.4	0.76	1.7	2.4	0.7
	2.3	0.47	5.3	5.1	0.2
	3.75	0.29	11.7	11.0	0.7
3.0	3.75	0.23	15.4	14.8	0.6
3.5	3.75	0.20	16.9	15.5	1.4

Table 7.14: Largest amplitude percent changes for simulations and predictions from plot in Fig. 7.28; also the difference between these two results: COMPLEX surfaces.

Mach number	ocean wave height	λ-Ratio largest ocean component (loc)	λ-Ratio smallest ocean component (soc)	simulation % change	Fig. 7.28 predicted % change (loc)	Fig. 7.28 predicted % change (soc)	magnitude of difference in percent (loc)	magnitude of difference in percent (soc)
combo 1								
1.4	2.3	0.80	1.60	7.9	2.0	1.0	5.9	6.9
	3.75	0.49	0.98	11.2	5.0	2.0	6.2	9.2
2.4	2.3	0.47	0.93	3.5	5.1	2.0	1.6	1.5
	3.75	0.29	0.57	11.2	11.0	5.0	0.2	6.2
combo 2								
1.4	2.3	0.80	2.41	7.5	2.0	—	5.5	—
	3.75	0.49	1.47	8.5	5.0	1.3	3.5	7.2
2.4	2.3	0.47	1.40	2.8	5.1	1.3	2.3	1.5
	3.75	0.29	0.86	8.3	11.1	2.5	2.8	5.8

7.4.4. Discussion of results

In conducting these wavelength comparison studies, a tool was developed to help predict the magnitude of the effect that simple ocean curvature has on an impinging waveform. For a particular Mach number and ocean wave height, the plot in Fig. 7.28 can be used to estimate the percent change of the acoustic pressure just under the surface from a flat ocean to that with simple ocean swell. For complex surfaces, however, the plot cannot be applied. So although the complex surface studies prompted the wavelength comparison studies, the resulting prediction plot cannot help in determining the underwater pressure. The wavelength comparison studies do not explain the unpredictability of the curvature effects felt by a sonic boom interacting with a complex ocean surface. It is concluded that each complex surface case must be treated individually as in Sect. 7.3.

7.5. Summary of Overall Results for a Wavy Ocean Surface

Qualitatively, it is seen that as a sonic boom impinges upon a wavy ocean surface, its energy is focused in the ocean wave troughs and defocused over the ocean wave crests. The amplitude of the underwater pressure just under the troughs is higher, and in the crests is lower, than would be for the case of the pressure values just under a flat ocean surface. Quantitatively, looking at a sonic boom with a peak pressure of 50 Pa and duration of 300 ms, this pressure increase/decrease is at most 11.7% for a simple ocean surface and 11.2% for a complex ocean surface: in either case, the change in peak dB level is 1 or less. Two trends can be identified when extracting pressure values from just under a simple ocean swell: 1.) increasing wind wave heights strengthens the focusing and defocusing of the evanescent acoustic pressure and 2.) increasing the Mach number strengthens the focusing and defocusing due to curvature. The first trend is confirmed by the complex ocean surface simulations, but the second is not. Studying the relation between the sonic boom effective wavelength and the ocean wavelength reveals a useful tool for predicting the curvature effect for various Mach numbers and ocean wave heights for an ocean with a simple surface profile: these studies also explain the second trend. However, this prediction tool cannot be applied to complex surfaces: each complex case must be treated independently. Lastly,

when extracting pressure values as a function of depth for a simple ocean surface simulation, it is shown that the effects of curvature on the underwater pressure are stronger near the ocean surface than at greater depths.

Chapter 8.
Inhomogeneous Ocean with Flat Surface: Concepts and Computational Simulations

Chapter 7 introduced the first realistic ocean feature, waviness on the ocean surface, to the computational simulation. This chapter introduces the second realistic ocean feature, bubbles near the surface. In order to isolate the effects of this feature, the ocean surface is flat for the simulations including the bubbles. According to Snell's Law, the speed of sound in water must be less than about 823 m/s for a flat ocean surface to allow propagation of a sonic boom waveform for the Mach 2.4 case. In a flat homogeneous ocean, the speed of sound is approximately 1500 m/s, so propagation is impossible. However, the presence of bubbles in an ocean changes its sound speed. In an extreme case,[5] the speed of sound in the bubbly water could be as low as 500 m/s; for the Mach 2.4 case, there should then be propagation into this ocean. It is therefore important to add the feature of ocean bubbles to the overall analysis. After a review of ocean bubble concepts, the results of the computational simulations including a simple flat model of the bubble layers and a bubble plume model will be revealed.

8.1. Ocean Bubbles

8.1.1. Bubble generation

Bubbles in the ocean are formed when wind-generated waves break. The Beaufort Scale (mentioned in Chapter 7, found in many references including Refs. 3, 7, and 22, among others) reveals that waves start to break when the wind speed is from 7–10 knots (kn) (3.4–5.4 m/s); it is at this sea state that scattered whitecaps appear.[3,7] Large numbers of bubbles are entrained during the breaking wave process; air is trapped or enveloped as the seawater surrounds it.[36] Figure 8.1 illustrates a breaking wave causing bubble entrainment; this is not an accurate illustration, rather just a pictorial explanation of entrainment. These

bubbles are actually seen in groups under the ocean surface as clouds or plumes, their shape and depth dependent on the turbulent motions in the water.[64] Figure 8.2 is taken from Thorpe:[64] it shows actual bubble plumes as a function of time, measurements taken in fresh water by sonar. These bubble plumes are presented simply to show their shapes: the number of meters in depth that the bubble plumes reach is dependent on the wind speed and will be explained later in this chapter.

time 1   time 2   time 3

Figure 8.1: Breaking ocean wave causing bubble entrainment.

ocean surface
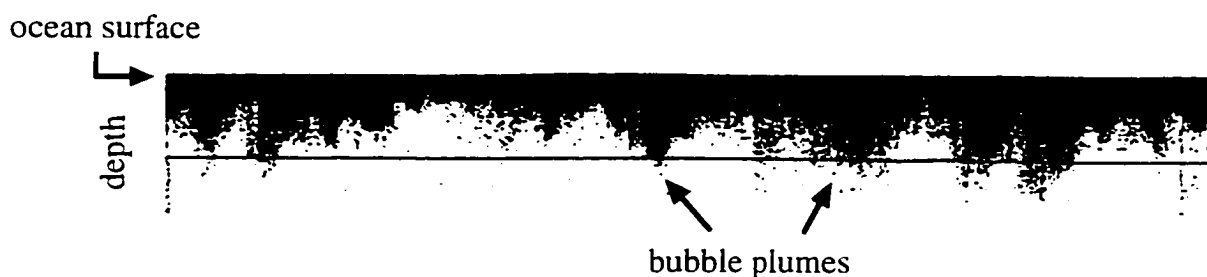
bubble plumes

Figure 8.2: Actual bubble plumes as a function of time: taken from Thorpe.[64] Illustrated to show bubble plume shapes.

Even though ocean bubbles dissipate after formation. their existence should not always be considered transient. Studying the bubble plume depth, the maximum lifetime of a bubble plume, and the frequency of wave breaking, it is determined that if the wind speed exceeds 12.6 kn (6.5 m/s), the bubbles will persist from one wave break to the next.[36] The

Beaufort Scale categorizes this wind speed as a "moderate breeze," where whitecaps are fairly frequent. So while at lower wind speeds bubbles will not exist or may be formed in discrete patches, the bubble plumes will overlap at higher ones forming a continuous bubble layer. Recall that the wind speeds applied in Chapter 7 were 10, 20, and 30 knots.

8.1.2. Bubble plume parameters

In general, the speed of sound in a bubble plume depends on the size and number of bubbles and on the frequency of the sound.[36] However, if the insonification frequency is much lower than any of the bubbles' resonance frequencies, then the plume can be viewed as a uniform body with an effective acoustic impedance.[36,13] The assumption just stated will be applied: what follows is the justification for its application.

Although an N-shaped sonic boom has significant high frequency content, it can be found that most of its energy is at frequencies below 400 Hz, most significant below 40 Hz (as stated in Chapter 2). In order to use the assumption, it must be shown that 400 Hz is much lower than any of the single bubbles' resonance frequencies. The resonance frequency of a freely oscillating bubble is called the Minnaert resonance frequency and can be expressed as[36]

$$ f_{bub} = \frac{1}{2\pi R_o} \sqrt{\frac{3\gamma p_o}{\rho_o}} \,. \tag{8.1} $$

where f_{bub} is the frequency in hertz, R_o is the bubble radius, γ is the ratio of specific heats for the liquid, p_o is the ambient pressure in the liquid, and ρ_o is the liquid density. For seawater, $\gamma = 1.01$ and $\rho_o \approx 1000$ kg/m³.

For R_o and p_o, it is best to examine the most extreme cases to find the lowest possible resonance frequency. The smallest value of ambient pressure is just under the surface; here, $p_o = 1.01 \times 10^5$ Pa. (The pressure can be calculated as a function of depth using $p = p_o + \rho_o g h$, where p_o is the pressure at the surface, ρ_o is the liquid density, g is the gravitational constant, and h is the distance from the surface.) Extreme bubble radii have been observed to be approximately 300 μm for the largest and 20 μm for the smallest.[64] Using Eq. (8.1), the resonance frequency of a 300 μm radius bubble near the ocean surface is calculated to be approximately 9.3 kHz. A more commonly observed bubble, say one with an 80 μm radius, has a resonance frequency near the surface of 34.8 kHz. Even the lowest

bubble resonance (9.3 kHz) is much greater than the majority of the sonic boom frequency content ($<$ 400 Hz).

It is therefore possible, assuming that the significant frequency content in a sonic boom is much lower than any bubble's resonance frequency, to look at the portion of the ocean containing bubbles as an area of effective acoustic impedance rather than modeling individual bubbles, its speed of sound and density must be calculated as a function of depth. The speed of sound profile used for an ocean with bubbles resembles the exponential law as a function of depth.[36,19] A plot of this profile can be seen in Fig. 8.3. The values reflect an extreme case[5] where the speed of sound at the surface is 500 m/s. The values decay to 1500 m/s at the maximum depth of the bubble plume.
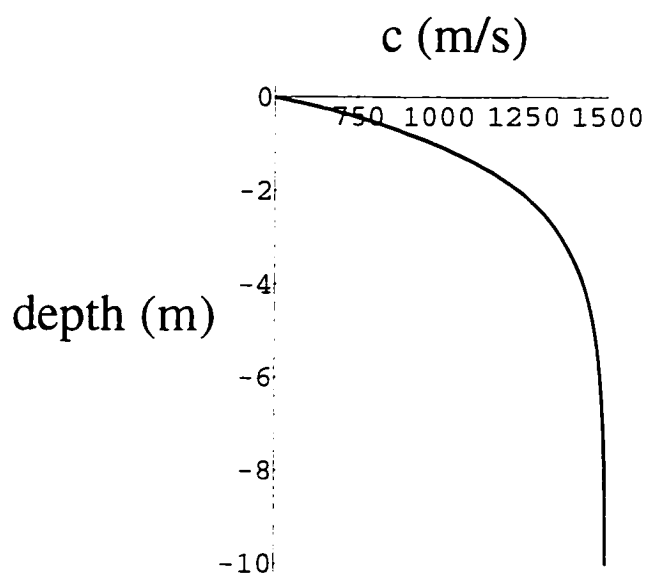
Figure 8.3: Speed of sound profile near the surface of an ocean with bubbles. The case shown is for a 30 knot wind.

The average depth of a bubble plume can be determined by[64]

$$depth_{Ave} \approx 0.4\,(wind\ speed - 2.5)\ .$$

$$(8.2)$$

The wind speed here is in meters/second and the $depth_{Ave}$ in meters. The maximum depth of a bubble plume is approximately twice the average depth.

$$depth_{Max} \approx B \times depth_{Ave} . \tag{8.3}$$

where B is approximately 2. actually about 2.3 for a 20 knot wind and 1.9 for a 30 knot wind. Equation (8.3) is formulated extracting values from plots in Farmer and Vagle[19] and Thorpe.[65] It can be seen from Eqs. (8.2) and (8.3) that as the wind speed increases. the maximum depth of the bubble plume increases. For example, for a wind speed of 20 knots (10.28 m/s), the maximum depth of the bubbles is approximated to be 7 m and for 30 knots (15.42 m/s) it is 10 m.

The density in bubbly water can be calculated as[13]

$$\rho_o = (1 - \alpha)\rho_l + \alpha\rho_g . \tag{8.4}$$

where ρ_l is the density in the pure liquid. ρ_g is the density of the gas inside of the bubbles, and α is the void fraction. the fraction of unit volume of the mixture occupied by the gas. The void fraction on average is found to be much less than 1%;[5,36] so. although the speed of sound is significantly changed due to the bubbles. the density change is negligible.

8.1.3. Analytical description of bubbles

The analytical description of ocean bubbles is here separated into two models. one simple and one more complicated. Instead of a continuous change in speed of sound with depth, the part of the ocean containing the bubbles is divided into sections or layers; as depth increases. each layer has an increasing sound speed. but the density remains a constant (the density of the water without the bubbles). A simple model is constructed with several flat bubble layers. and a more complicated model consists of one bubble plume also with several bubble layers.

A simple model is desirable in order to find the effects of the bubbles (on the sonic boom) just from the different impedance layers and not from the shape of the bubble plumes. Flat bubble layers. each with a different sound speed. span the width of the computational domain. Each layer has a specified height or thickness. the sound speed calculated at the layer's average depth using an exponential approximation similar to the one in Fig. 8.3.

The equation used for approximating the speed of sound in the bubble layers is

$$c_{bub} = c_{water} - \left(c_{water} - c_{surface}\right) e^{(\text{depth/efold})} . \tag{8.5}$$

where c_{water} is the speed of sound in the ocean without bubbles. $c_{surface}$ is the speed of sound in the bubble layers right near the surface (here taken to be 500 m/s), depth is the depth from the surface (a negative value), and efold determines the shape of the exponential decay. Values for efold are here case specific. for a 20 knot wind efold = 1.05, and for a 30 knot wind efold = 1.5. These efold values were chosen to approximate the shape of the sound speed profiles found in Ref. 19. The layers are narrower near the surface where the sound speed is rapidly changing. The number of layers depends on the maximum depth of the bubbles which depends on the wind speed. Appendix B shows the input file **bublayers** which contains all parameters for the flat bubble layers: the case shown is for a 20 knot wind. Values for the number of layers, each layer's thickness, and the speed of sound and density for each layer will be presented for each wind speed later in this chapter.

A more realistic model of bubbles in the ocean would include bubble plumes. If sound energy does penetrate a bubble plume, there is the possibility that it may become trapped. bouncing around and forming a concentrated noise source.

The more complicated model applied here is a single bubble plume. In order to model a bubble plume it is necessary to determine its approximate shape. This is somewhat of an ambiguous task since the shape can never be represented by a smooth. known function. The shape of a bubble plume depends on the heat flux through the water surface. columnar plumes appearing when the air is colder than the water. billow-like plumes when the water is colder than the air.[64] References 33 and 37 indicate that a semi-circular plume located immediately below the free surface is a good geometric model of observed plumes; this model seems to resemble the billow-like plumes more than the columnar plumes. For the current work, a function was chosen that represents a somewhat billow-like, somewhat columnar type plume.

A single bubble plume is modeled using a sine squared function. This function was chosen for its nice smooth profile. an important feature when approximating its curve with computational grid blocks. The model, seen in Fig. 8.4. is comprised of bubble layers. The speed of sound for each bubble layer is calculated as if the layers were flat, as in the previous
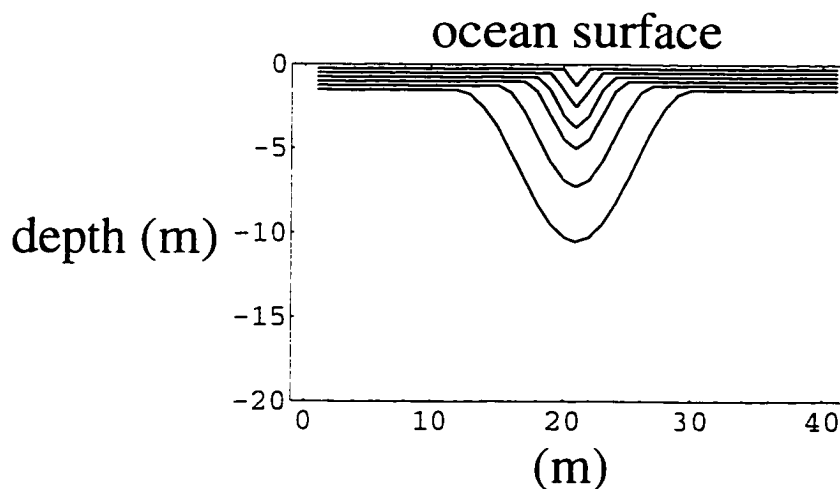
Figure 8.4: Profile of bubble plume near the surface of the ocean. The case shown is for a 30 knot wind.

model. However. there are fewer layers used in the bubble plume model than in the flat bubble layer model.

8.2. Flat Bubble Layer Model: Computational Simulations

8.2.1. Computer program

The computer program used for the computational simulations in Chapters 6 and 7 is also used here. In addition. flat bubble layers are included in the code. The program allows the choice of bubbles or no bubbles; if bubbles are chosen, an input file containing all the information for the bubble layers is then read.

In addition to using the air-water interface finite difference scheme at the flat ocean surface. it is also applied between each bubble layer. Moreover, the refined grid region is now stretched to include all bubble layers. It should be noted that since the domain is still 800 grid points in the z direction, this implies that the domain has physically shrunk.

8.2.2. Program runs

Two runs of the program including the flat bubble layer model were made in order to help assess the impact bubbles have on an impinging sonic boom waveform. For the sonic boom, the peak pressure is 50 Pa and the duration 300 ms; the only Mach number used is 2.4. Calculations were performed for two different wind speeds, 20 knots (10.3 m/s) and 30 knots (15.4 m/s). In each case the ocean surface is flat: while unrealistic for these wind speeds, this separates the bubble effects from the wind wave effects. Table 8.1 presents the bubble layer parameters used for the 20 knot and 30 knot flat bubble layer model runs. As seen in Table 8.1, each layer has an associated thickness. The maximum depth, the total of all thicknesses added together, is determined using Eqs. (8.2) and (8.3). The speed of sound is calculated using Eq. (8.5), and the density is the same in all cases since it was determined that the density change due to bubbles is negligible for this work.

Table 8.1: Flat bubble layer model parameters for two wind speed cases: c_o is the speed of sound. and ρ_o is the density.

layer number	bubble layers for 20 kn wind case			bubble layers for 30 kn wind case		
	thickness (m)	c_o (m/s)	ρ_o (kg/m^3)	thickness (m)	c_o (m/s)	ρ_o (kg/m^3)
1	0.5	712	1000	0.5	654	1000
2	0.5	1010	1000	0.5	893	1000
3	0.5	1196	1000	0.5	1065	1000
4	0.5	1311	1000	0.5	1189	1000
5	0.5	1383	1000	0.5	1277	1000
6	0.5	1427	1000	0.5	1340	1000
7	1.0	1464	1000	1.0	1403	1000
8	1.0	1486	1000	1.0	1450	1000
9	2.0	1497	1000	2.0	1482	1000
10	—	—	—	3.0	1497	1000

8.2.3. Effects caused by flat bubble layers in ocean

The simulation that will now be analyzed is for the 30 knot case. It was reasoned that the more extreme case would be more likely to show effects due to the bubbles if there are any. A plot of the full computational domain appears in Fig. 8.5. The solid white lines represent the ocean surface and the bottom of each bubble layer; as in previous full-domain pictures, the dashed white lines enclose the refined grid region. This picture shows a time near the completion of the simulation, $t = 0.75$ s. After observation, it appears that there is no propagation of the sonic boom into the ocean; the pressure field and especially the intensity field indicate that the sonic boom is penetrating the ocean as an evanescent wave. As in previous simulation results, the grid appears stretched in the refined grid region; this region is larger than before since the refinement surrounds all of the bubbles layers, not just 3 m above and below the air-water interface.

Now that it is seen that flat layers of bubbles do not allow propagation into the ocean, the question remains as to whether or not they affect the evanescent field. It is appropriate to compare a run with the ocean bubbles to one without in order to find the answer. A vertical slice of data is extracted from the two-dimensional computational domain, Fig. 8.5, at two horizontal locations: these two locations are chosen to be the "hottest," greatest pressure value, and "coldest," lowest pressure value, just under the ocean surface. Figure 8.6 shows two line graphs, each a vertical extraction at the greatest positive peak pressure, for the runs with and without ocean bubbles. The vertical axis is pressure and the horizontal axis is the vertical grid location: the interface is at -340 m, pointed to with a large black arrow. It can be seen that the two curves overlap, indicating that the ocean bubbles do not have an effect on the evanescent field. Figure 8.7, vertical line extractions at the greatest negative peak pressure, repeats this indication.

Further investigation, however, reveals that there are slight changes in the underwater pressure field. Table 8.2 presents percent changes from the flat homogeneous ocean results (found in Chapter 6) caused by the flat ocean bubble layers. The results at several depths are shown since the bubble layers extend several meters into the ocean, and any potential effects of the bubbles may be apparent there instead of just near the surface. Table 8.2 indicates that most of the changes in peak pressure are less than 1%. The maximum

pressure and intensity; M=2.4; flat bub. layers; 30 kn wind; #160
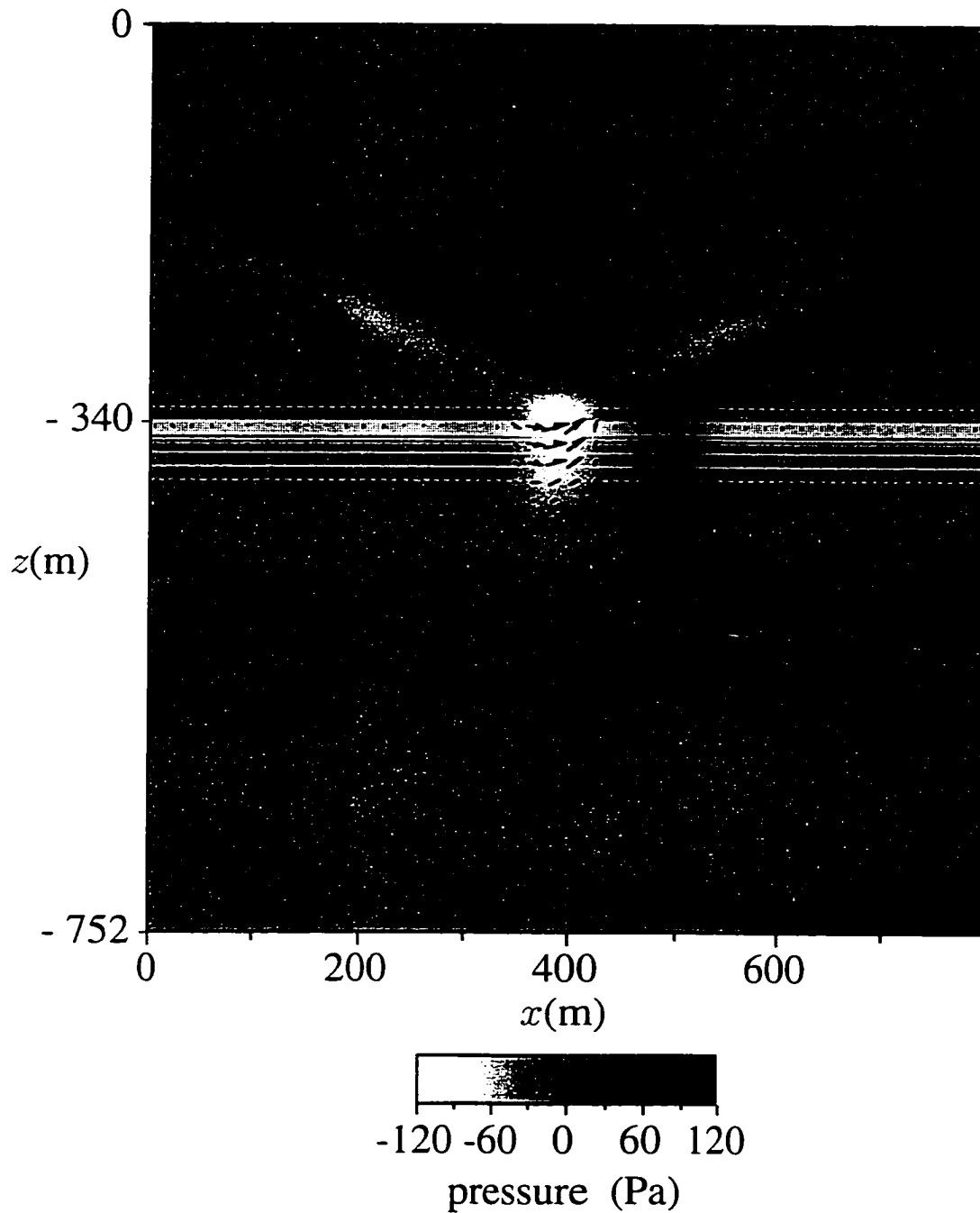
Figure 8.5: Mach number 2.4, flat ocean surface, flat bubble layers, pressure and intensity fields; incident wave interacting with the air-water interface; $t = 0.75$ s.

percent change is at the 128 m depth, a magnitude of 4.57% for the 20 knot wind case and 5.33% for the 30 knot wind case: the changes in pressure (Pa) are also given for these cases just to show really how little the pressure is affected. Since the pressure values are relatively small at these depths these percent changes correspond to only 0.5 dB and 0.4 dB changes, respectively. Even though the effects are shown to be minimal, it should be noted that the percent changes are slightly greater for the 30 knot wind case than the 20 knot wind case.

Table 8.2: Percent change in peak pressure from ocean without bubbles caused by the flat bubble layers; Mach 2.4, 20 and 30 knot wind cases.

depth (m)	% difference caused by flat bubble layers +/− peaks	
	20 kn case	30 kn case
0	0.03% / 0.02%	0.04% / 0.03%
4	0.06% / 0.07%	0.09% / 0.12%
8	0.05% / 0.07%	0.09% / 0.15%
16	0.01% / 0.10%	0.03% / 0.20%
32	−0.06% / 0.20%	−0.10% / 0.41%
64	−0.46% / 0.55%	−0.72% / 1.10%
128	−4.57% / 1.94% (−0.42 Pa / −0.23 Pa)	−5.33% / 3.29% (−0.49 Pa / −0.39 Pa)

8.2.4. Discussion of results

Computational simulations including flat bubble layers and a flat ocean surface indicate that the sonic boom noise penetrates the ocean surface only as an evanescent wave. As stated before, for the Mach 2.4 case the sound should penetrate the ocean surface as a propagating wave if the sound speed is less than 823 m/s. Table 8.1 shows that the top layer of bubbles for both the 20 knot and 30 knot cases has a sound speed less than 823 m/s; the underwater sound field is probably unaffected by this layer because its thickness is only 0.5 m. Since the sonic boom effective wavelength is quite large in comparison ($\approx$ 103 m in air for the

Figure 8.6: Vertical intersection of the greatest positive peak pressure of the sonic boom waveform. The solid gray line represents the numerical data for the ocean without bubbles; the dashed black line represents the numerical data for the ocean with bubbles.

Ocean with and without bubbles
(negative peak vertical line extraction)
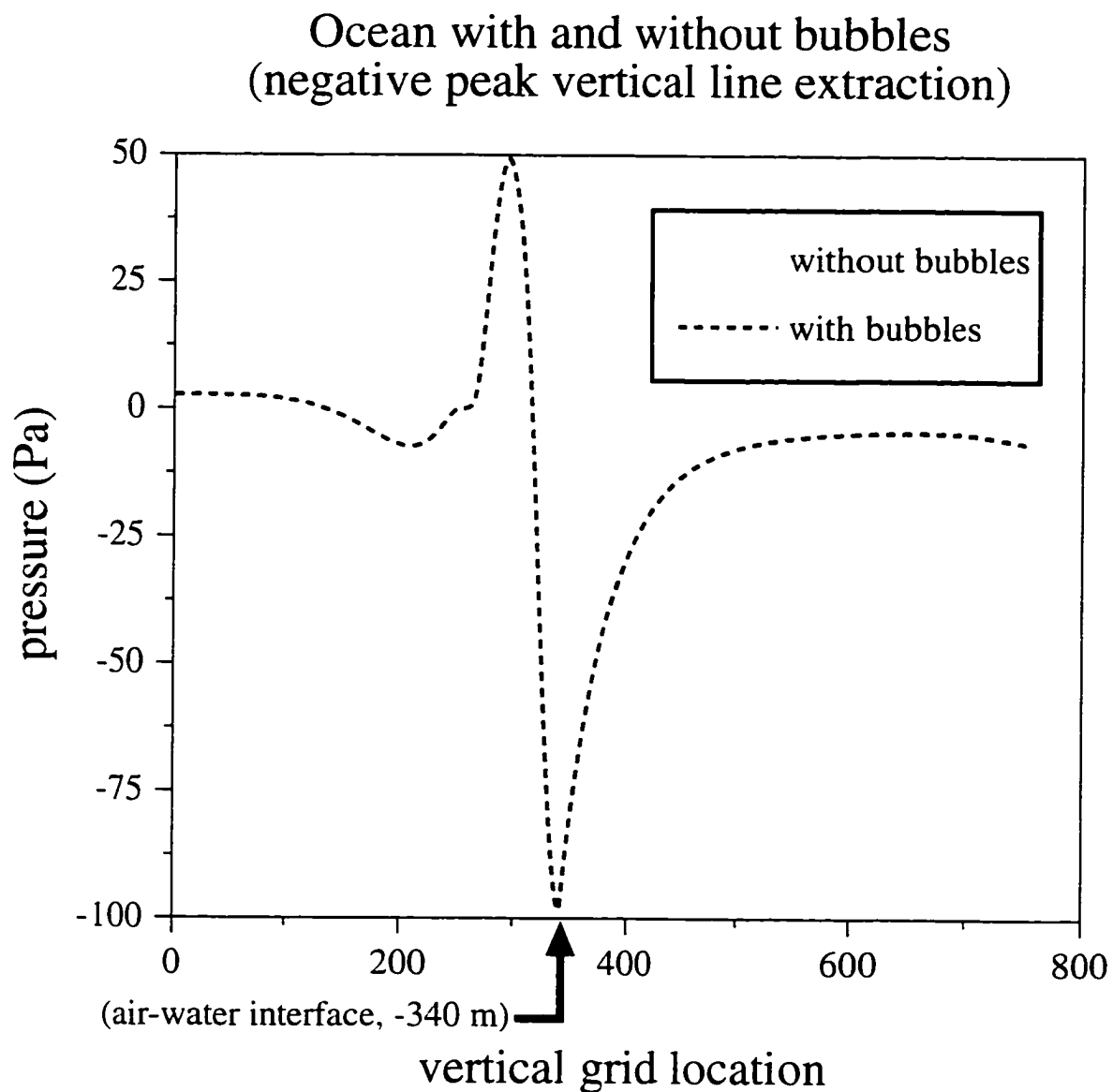
Figure 8.7: Vertical intersection of the greatest negative peak pressure of the sonic boom waveform. The solid gray line represents the numerical data for the ocean without bubbles; the dashed black line represents the numerical data for the ocean with bubbles.

300 ms duration sonic boom), the top bubble layer is most likely invisible to the incoming waveform. The underwater pressure field is only minimally affected by the bubble layers, the deeper bubble layers for the 30 knot wind case showing slightly greater changes than the 20 knot wind case. Studying changes in sound at several depths for each of the two runs, the largest effect on sound level is found to be less than 1 dB. This indicates that the effects caused by the flat bubble layers are negligible.

8.3. Bubble Plume Model: Computational Simulations

8.3.1. Computer program

Again the computer program used for the computational simulations in Chapters 6 and 7 is also used here. In addition, the layered bubble plume is included in the code. Similar to the flat bubble layer model, the bubble plume model applies the interface finite difference scheme at the ocean surface and between each of the bubble layers. The refined grid region envelopes the entire bubble plume.

8.3.2. Program runs

Two runs of the program including the bubble plume model were made in order to help assess the impact bubbles have on an impinging sonic boom waveform. For the sonic boom, the peak pressure is again 50 Pa and the duration 300 ms: the Mach numbers applied are 1.4 and 2.4. For this bubble plume model, a computer simulation of the more extreme wind speed, 30 knots, was performed. As in the flat bubble layer model runs, the ocean surface is flat. Table 8.3 presents the bubble layer parameters used for the 30 knot runs. Each bubble layer has an associated thickness at the location of the bubble plume. As before, the maximum depth, the total of all thicknesses added together, is approximated using Eqs. (8.2) and (8.3): using these equations, the maximum depth of the bubble plume should be about 10 m. For the bubble plume model, the maximum depth is actually 10.5 meters; there is a good reason for this. The bubble layers in the bubble plume model are closely packed everywhere except at the bubble plume location, as seen in Fig. 8.4. These

compacted layers are limited in closeness by the vertical grid spacing, 0.25 m in this refined region. Since each layer has at least a 1 m thickness (a limitation for this model) in the plume and there are six layers. the total depth would be the sum of the 6 thicknesses plus 0.25 times the number of layers: total depth $= 9 + 1.5 = 10.5$ m. The speed of sound is calculated using Eq. (8.5). and the density is the same in all cases.

Table 8.3: Bubble plume model parameters for 30 knot wind: c_o is the speed of sound. and ρ_o is the density.

layer number	bubble layers for 30 kn wind case		
	thickness (m)	c_o (m/s)	ρ_o (kg/m^3)
1	1.0	841	1000
2	1.0	1214	1000
3	1.0	1375	1000
4	1.0	1466	1000
5	2.0	1483	1000
6	3.0	1497	1000

8.3.3. Effects caused by bubble plume in ocean

The full two-dimensional computational domain for the Mach 2.4 case is shown in Fig. 8.8: this figure includes the intensity vectors in order to see if the bubble plume is warping the shape of the evanescent field. The thick white horizontal line is actually the collection of lines representing the ocean surface and the bottom of each bubble layer: although difficult to see. these lines separate near the middle of the domain to form the bubble plume (indicated by an arrow). As before. the dashed white lines enclose the refined grid region. Even though Fig. 8.8 shows just a snapshot in time. it is clear from tracking the waveform over time that the sonic boom penetrates as an evanescent wave. the field relatively unaffected as it passes over the bubble plume.

pressure and intensity; M=2.4; bubble plume; 30 kn wind; #160
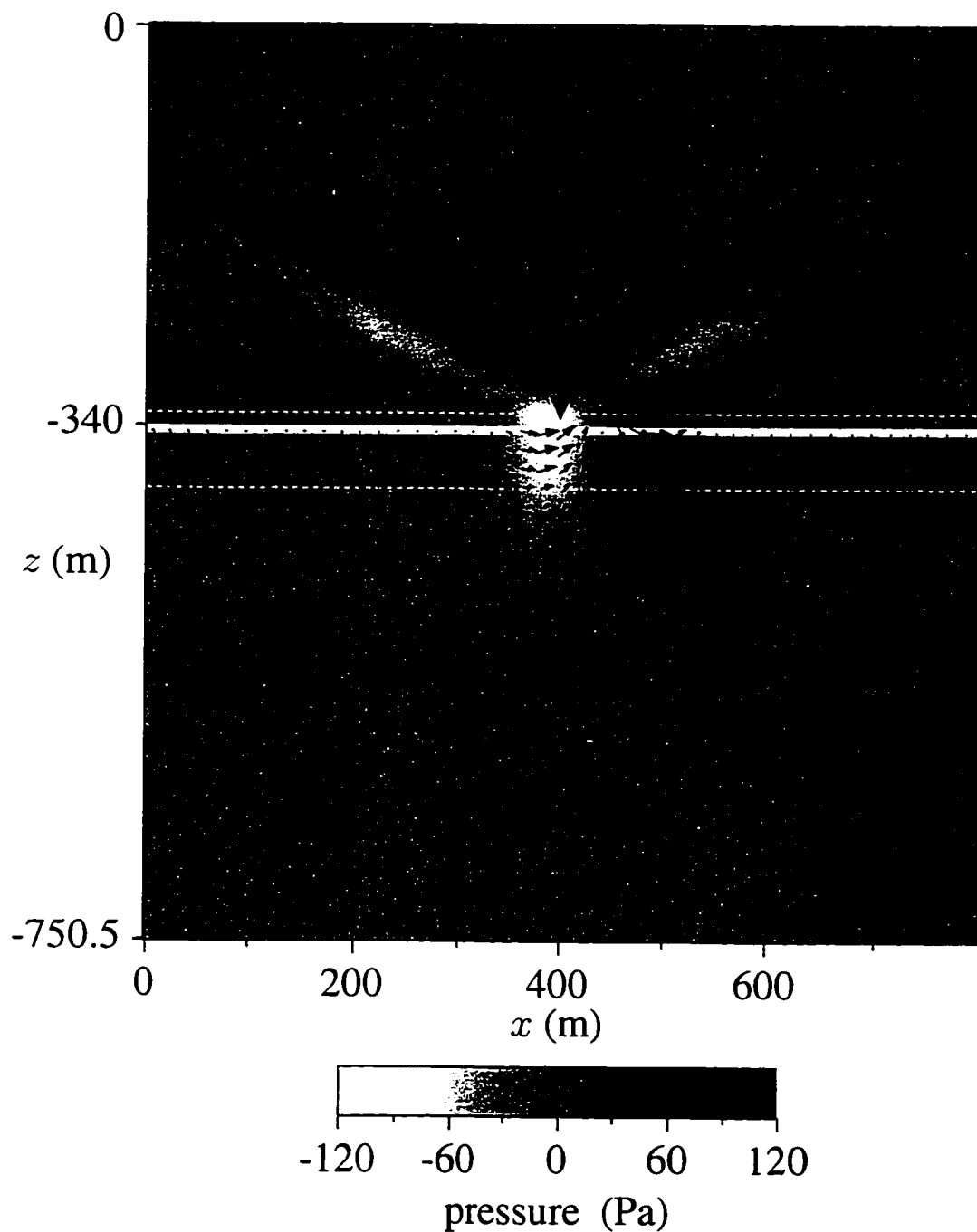
Figure 8.8: Mach number 2.4, flat ocean surface, one bubble plume, pressure and intensity fields; incident wave interacting with the air-water interface; $t = 0.75$ s.

The smaller Mach number calculation was performed in order to determine whether or not a sonic boom with a shorter horizontal effective wavelength would more easily penetrate the narrow opening of the bubble plume. A run was performed with a Mach 1.4 sonic boom. The results still indicate that the evanescent field is relatively unaffected by the bubble plume. Table 8.4 presents percent changes from the flat homogeneous ocean results (found in Chapter 6) caused by the bubble plume. The results at several depths are shown. The results indicate that most of the changes in peak pressure are less than 1%. The maximum percent change. the same as for the flat bubble layer model, is at the 128 m depth. a magnitude of 6.02% for the Mach 1.4 case and 5.54% for the Mach 2.4 case; the changes in pressure (Pa) are also given for these cases just to show how little the pressure is affected. These percent changes correspond to only 0.5 dB changes.

Table 8.4: Percent change in peak pressure from ocean without bubbles caused by the bubble plume; Mach 1.4 and 2.4. 30 knot wind case.

depth (m)	% difference caused by bubble plume +/− peaks	
	M = 1.4	M = 2.4
0	0.01% / 0.00%	0.02% / 0.01%
4	0.00% / 0.01%	−0.01% / 0.03%
8	−0.01% / 0.01%	−0.01% / 0.06%
16	−0.02% / −0.02%	−0.06% / −0.15%
32	−0.07% / −0.04%	−0.21% / −0.37%
64	−0.57% / −0.47%	−0.80% / 1.10%
128	−6.02% / −3.64% (−0.13 Pa / 0.16 Pa)	−5.54% / 3.46% (−0.51 Pa / −0.41 Pa)

8.3.4. Discussion of results

Computational simulations including the bubble plume model and a flat ocean surface indicate that the sonic boom noise penetrates the ocean surface only as an evanescent wave. The underwater pressure field is only minimally affected by the bubble plume both for the Mach 1.4 and Mach 2.4 cases. Examining several depths for the two runs, the largest change in sound level is less than 1 dB. This indicates that the effects caused by the bubble plume are negligible.

8.4. Summary of Overall Results for an Ocean with Bubbles

This chapter focused on the effect of ocean bubbles on an impinging sonic boom waveform. Accessing a wide array of literature, models for the ocean bubbles were formed, the simplest model being flat bubble layers and a more complicated model being a single bubble plume. These models became features in the existing finite difference code. Simulations were performed for each of the models: results indicated that ocean bubbles do not allow the sonic boom to penetrate the ocean as a propagating wave but rather just an evanescent wave. The bubbles only negligibly affect the pressure values in the evanescent field, the magnitude of the maximum change being less than 1 dB in sound pressure level.

Chapter 9.
Verification of Results

The method applied for computationally producing the results found in Chapters 6, 7, and 8 can be verified through numerical accuracy and differences in the initial waveform. Chapter 6 provided some verification by making comparisons to known analytical theories and corresponding analytical/numerical models where trends in the underwater pressure field were duplicated; the current chapter provides further validation for the computational method and the results produced. Since the finite difference scheme used over the majority of the grid (the air/water scheme) was 2nd-order accurate, a 4th-order accurate in space scheme is applied to the computational domain in this chapter to check the accuracy of the solution. Also, the initial waveform used in the program was a rounded sonic boom; here a more N-shaped sonic boom replaces the rounded one to see if this makes much of a difference in the calculated underwater pressure field.

9.1. Fourth-Order Finite Difference Simulations

To check the accuracy of the 2nd-order finite difference code, the results are compared to that of a 4th-order accurate in space finite difference code. This section will present the derivation for a 4th-order scheme for both a uniform and nonuniform grid. This scheme is then implemented in the air/water wave propagation simulations, and the 4th-order accurate results are then compared to the 2nd-order accurate results.

9.1.1. Derivation of fourth-order finite difference method

The 4th-order centered finite difference approximation for a *uniform* grid starts with four Talyor series expansions.

$$p\left(x+2\Delta x\right)=p\left(x\right)+2\Delta x p_x\left(x\right)+\frac{\left(2\Delta x\right)^2}{2!}p_{xx}\left(x\right)+\frac{\left(2\Delta x\right)^3}{3!}p_{xxx}\left(x\right)+\frac{\left(2\Delta x\right)^4}{4!}p_{x(iv)}\left(x\right)$$

$$+\frac{\left(2\Delta x\right)^5}{5!}p_{x(v)}\left(x\right)+\frac{\left(2\Delta x\right)^6}{6!}p_{x(vi)}\left(x\right)+\ldots .\qquad(9.1a)$$

$$p\left(x+\Delta x\right)=p\left(x\right)+\Delta x p_x\left(x\right)+\frac{\Delta x^2}{2!}p_{xx}\left(x\right)+\frac{\Delta x^3}{3!}p_{xxx}\left(x\right)+\frac{\Delta x^4}{4!}p_{x(iv)}\left(x\right)$$

$$+\frac{\Delta x^5}{5!}p_{x(v)}\left(x\right)+\frac{\Delta x^6}{6!}p_{x(vi)}\left(x\right)+\ldots .\qquad(9.1b)$$

$$p\left(x-\Delta x\right)=p\left(x\right)-\Delta x p_x\left(x\right)+\frac{\Delta x^2}{2!}p_{xx}\left(x\right)-\frac{\Delta x^3}{3!}p_{xxx}\left(x\right)+\frac{\Delta x^4}{4!}p_{x(iv)}\left(x\right)$$

$$-\frac{\Delta x^5}{5!}p_{x(v)}\left(x\right)+\frac{\Delta x^6}{6!}p_{x(vi)}\left(x\right)-\ldots .\qquad(9.1c)$$

$$p\left(x-2\Delta x\right)=p\left(x\right)-2\Delta x p_x\left(x\right)+\frac{\left(2\Delta x\right)^2}{2!}p_{xx}\left(x\right)-\frac{\left(2\Delta x\right)^3}{3!}p_{xxx}\left(x\right)+\frac{\left(2\Delta x\right)^4}{4!}p_{x(iv)}\left(x\right)$$

$$-\frac{\left(2\Delta x\right)^5}{5!}p_{x(v)}\left(x\right)+\frac{\left(2\Delta x\right)^6}{6!}p_{x(vi)}\left(x\right)-\ldots .\qquad(9.1d)$$

In addition to suppressing the arguments. the following variable representations will be applied throughout the remainder of the derivation:

$$p\left(x\pm 2\Delta x\right)\equiv p_{i\pm 2}$$

$$p\left(x\pm\Delta x\right)\equiv p_{i\pm 1}\qquad(9.2)$$

$$p\left(x\right)\equiv p_i .$$

Adding together Eqs. (9.1a) and (9.1d) gives

$$p_{i+2}+p_{i-2}=2p_i+4\Delta x^2 p_{xx}+\frac{16\Delta x^4}{12}p_{x(iv)}+\frac{128\Delta x^6}{6!}p_{x(vi)}+\ldots ,\qquad(9.3)$$

and adding together Eqs. (9.1b) and (9.1c) gives

$$p_{i+1}+p_{i-1}=2p_i+\Delta x^2 p_{xx}+\frac{\Delta x^4}{12}p_{x(iv)}+\frac{2\Delta x^6}{6!}p_{x(vi)}+\ldots .\qquad(9.4)$$

Multiplying Eq. (9.4) by -16 then adding the result to Eq. (9.3), the formulation becomes

$$p_{i+2}-16p_{i+1}-16p_{i-1}+p_{i-2}=-30p_i-12\Delta x^2 p_{xx}+\frac{96\Delta x^6}{6!}p_{x(vi)}+\ldots .\qquad(9.5)$$

Solving for the second derivative. Eq. (9.5) is written

$$p_{xx} = \frac{1}{12\Delta x^2}\left(-p_{i+2} + 16p_{i+1} - 30p_i + 16p_{i-1} - p_{i-2}\right) + \frac{\Delta x^4}{90}p_{x(vi)} + \dots \quad . \tag{9.6}$$

This approximation matches that in Ref. 25. So the 4th-order centered finite difference formulation for a uniform grid is

$$p_{xx} = \frac{1}{12\Delta x^2}\left(-p_{i+2} + 16p_{i+1} - 30p_i + 16p_{i-1} - p_{i-2}\right) + \mathcal{O}\left(\Delta x^4\right) \quad . \tag{9.7}$$

This 4th-order derivative approximation is applied only to the spatial derivative in the x direction.

The 4th-order centered finite difference approximation for a *nonuniform* grid also starts with four Talyor series expansions.

$$p\left[z + (\Delta z_{j+2} + \Delta z_{j+1})\right] = p(z) + (\Delta z_{j+2} + \Delta z_{j+1})\, p_z(z) + \frac{(\Delta z_{j+2} + \Delta z_{j+1})^2}{2!}p_{zz}(z)$$
$$+ \frac{(\Delta z_{j+2} + \Delta z_{j+1})^3}{3!}p_{zzz}(z) + \frac{(\Delta z_{j+2} + \Delta z_{j+1})^4}{4!}p_{z(iv)}(z)$$
$$+ \frac{(\Delta z_{j+2} + \Delta z_{j+1})^5}{5!}p_{z(v)}(z) + \dots \quad . \tag{9.8a}$$

$$p\left[z + \Delta z_{j+1}\right] = p(z) + \Delta z_{j+1}p_z(z) + \frac{(\Delta z_{j+1})^2}{2!}p_{zz}(z) + \frac{(\Delta z_{j+1})^3}{3!}p_{zzz}(z)$$
$$+ \frac{(\Delta z_{j+1})^4}{4!}p_{z(iv)}(z) + \frac{(\Delta z_{j+1})^5}{5!}p_{z(v)}(z) + \dots \quad , \tag{9.8b}$$

$$p\left[z - \Delta z_j\right] = p(z) - \Delta z_j p_z(z) + \frac{(\Delta z_j)^2}{2!}p_{zz}(z) - \frac{(\Delta z_j)^3}{3!}p_{zzz}(z)$$
$$+ \frac{(\Delta z_j)^4}{4!}p_{z(iv)}(z) - \frac{(\Delta z_j)^5}{5!}p_{z(v)}(z) + \dots \quad . \tag{9.8c}$$

$$p\left[z - (\Delta z_j + \Delta z_{j-1})\right] = p(z) - (\Delta z_j + \Delta z_{j-1})\, p_z(z) + \frac{(\Delta z_j + \Delta z_{j-1})^2}{2!}p_{zz}(z)$$
$$- \frac{(\Delta z_j + \Delta z_{j-1})^3}{3!}p_{zzz}(z) + \frac{(\Delta z_j + \Delta z_{j-1})^4}{4!}p_{z(iv)}(z)$$
$$- \frac{(\Delta z_j + \Delta z_{j-1})^5}{5!}p_{z(v)}(z) + \dots \quad . \tag{9.8d}$$

Recall that $\Delta z_j = z_j - z_{j-1}$. z_j being the physical location at index j on the grid: knowing the grid location is necessary for Δz since the refined grid region has smaller spacing than the rest of the computational grid. In addition to suppressing the arguments, the following

variable representations will be applied throughout the remainder of the derivation:

$$p\left[z + (\Delta z_{j+2} + \Delta z_{j+1})\right] \equiv p_{j+2} \qquad a \equiv \Delta z_{j+2} = z_{j+2} - z_{j+1}$$

$$p\left[z + \Delta z_{j+1}\right] \equiv p_{j+1} \qquad b \equiv \Delta z_{j+1} = z_{j+1} - z_j$$

$$p\left[z - \Delta z_j\right] \equiv p_{j-1} \qquad c \equiv \Delta z_j = z_j - z_{j-1} \qquad (9.9)$$

$$p\left[z - (\Delta z_j + \Delta z_{j-1})\right] \equiv p_{j-2} \qquad d \equiv \Delta z_{j-1} = z_{j-1} - z_{j-2}$$

$$p(z) \equiv p_j \ .$$

Multiplying Eq. (9.8a) by $(c + d)$ and Eq. (9.8d) by $(a + b)$ then adding the two results together and multiplying by $\frac{1}{(a+b)(c+d)}$ gives

$$\frac{p_{j+2}}{(a+b)} + \frac{p_{j-2}}{(c+d)} = \frac{a+b+c+d}{(a+b)(c+d)}p_j + \frac{(a+b)+(c+d)}{2!}p_{zz} + \frac{(a+b)^2 - (c+d)^2}{3!}p_{zzz}$$

$$+ \frac{(a+b)^3 + (c+d)^3}{4!}p_{z(iv)} + \frac{(a+b)^4 - (c+d)^4}{5!}p_{z(v)} + \cdots \ .$$

$$(9.10)$$

Multiplying Eq. (9.8b) by c and Eq. (9.8c) by b then adding these two results together and multiplying by $\frac{1}{bc}$ gives

$$\frac{p_{j+1}}{b} + \frac{p_{j-1}}{c} = \frac{b+c}{bc}p_j + \frac{b+c}{2!}p_{zz} + \frac{b^2 - c^2}{3!}p_{zzz} + \frac{b^3 + c^3}{4!}p_{z(iv)} + \frac{b^4 - c^4}{5!}p_{z(v)} + \cdots \ .$$

$$(9.11)$$

Now left with two equations, Eqs. (9.10) and (9.11), it is possible to combine them to get one equation, the nonuniform derivative. Multiplying Eq. (9.10) by $\frac{-4}{a+b+c+d}$ and Eq. (9.11) by $\frac{16}{b+c}$, performing minor manipulations, adding these two results together, then solving for p_{zz}, the expression for the 4th-order centered finite difference approximation of the second derivative can be stated as

$$p_{zz} = \frac{2}{3(a+b+c+d)}\left[\frac{-p_{j+2} + p_j}{a+b} + \frac{-p_{j-2} + p_j}{c+d}\right] + \frac{8}{3(b+c)}\left[\frac{p_{j+1} - p_j}{b} + \frac{p_{j-1} - p_j}{c}\right]$$

$$+ \left[\frac{1}{9}\frac{(a+b)^2 - (c+d)^2}{a+b+c+d} - \frac{4}{9}(b^2 - c^2)\right]p_{zzz} \ .$$

$$(9.12)$$

It is possible to check this scheme by letting $a = b = c = d = \Delta z$; Eq. (9.12) then collapses down to the uniform grid derivative, Eq. (9.7), except the difference in variables (z instead of x). This 4th-order approximation for a nonuniform grid is applied only to the z derivative.

Now the total 4th-order air/water scheme (4th-order in space, 2nd-order in time) is presented:

$$p_{i,j}^{n+1} = 2p_{i,j}^n - p_{i,j}^{n-1} + c_o^2(\Delta t)^2\left[p_{xx} + p_{zz}\right] \ . \qquad (9.13)$$

where

$$p_{xx} = \frac{1}{12\Delta x^2}\left(-p_{i+2,j} + 16p_{i+1,j} - 30p_{i,j} + 16p_{i-1,j} - p_{i-2,j}\right) .$$

$$p_{zz} = \frac{2}{3\left(\Delta z_{j+2} + \Delta z_{j+1} + \Delta z_j + \Delta z_{j-1}\right)}\left[\frac{-p_{i,j+2} + p_{i,j}}{\Delta z_{j+2} + \Delta z_{j+1}} + \frac{-p_{i,j-2} + p_{i,j}}{\Delta z_j + \Delta z_{j-1}}\right]$$
$$+ \frac{8}{3\left(\Delta z_{j+1} + \Delta z_j\right)}\left[\frac{p_{i,j+1} - p_{i,j}}{\Delta z_{j+1}} + \frac{p_{i,j-1} - p_{i,j}}{\Delta z_j}\right] .$$

(9.14)

9.1.2. Implementation into computer program and runs performed

Equations (9.13) and (9.14) form the finite difference scheme applied in the air and water sections of the computational domain, the majority of the grid. The finite difference scheme applied at the interface is still Eq. (5.25), the scheme derived in Chapter 5. The new 4th-order scheme is coded in Fortran. the new subroutine sochdiff2d can be seen in Appendix D. Besides this subroutine the rest of the code is similar to that with 2nd-order accuracy.

Several runs were performed in order to compare the 2nd-order accurate and 4th-order accurate results. First, the air-water interface was placed near the bottom of the rectangular domain so that the sonic boom could propagate through a homogeneous medium without interruption. Two runs were executed in this state. the first with the 2nd-order code and the second with the 4th-order code; in both cases, the Mach number was 2.4. the peak pressure of the sonic boom was 50 Pa. and the duration of the sonic boom was 300 ms. The goal here was to find out if the waveform was dissipating at different rates for the different accuracies. Then the interface was placed back in it's original position, 340 m from the top of the domain. Two more runs were performed, both 4th-order accurate but now with a flat ocean surface and a trochoidal surface profile of wave height 3.75 m. This second set of runs was necessary to see if results changed with the interface involved.

It should be noted that the 2nd- and 4th-order runs were all performed on the same grid. Although one benefit of 4th-order codes is their ability to use a coarser grid compared to 2nd-order codes, the intent here was to check the accuracy of the existing 2nd-order code. not to decrease the run times.

9.1.3. Results

In order to compare the different dissipation rates of the 2nd-order accurate propagation and the 4th-order accurate propagation, the two runs with the interface moved near the bottom of the domain are examined. Results indicate that over a 100 m distance the 2nd- and 4th-order accurate schemes have each decreased in peak pressure by at most 0.008 Pa. This implies that not only are they dissipating at the same rate, but the rate of dissipation is negligible.

Next, with the interface back in place, the 4th-order accurate flat water run is compared to the 2nd-order accurate flat water run, described in Chapter 6, and the 4th-order accurate 3.75 m wave height trochoidal profile run is compared to that of second order in Chapter 7. For the first set of runs, simulations involving the flat ocean surface, values are extracted just under the ocean surface and at a depth of 100 m. For the second set, simulations involving the wavy ocean surface, pressure values are extracted only from just under the surface. Table 9.1 presents the largest percent changes between pressure values. The percent changes were calculated as $100 \times (p_{4tho} - p_{2ndo})/p_{2ndo}$. All of the differences were less than 2%, but the changes are slightly higher for the run involving ocean surface curvature.

Table 9.1: Percent change in peak pressure comparing the 2nd- and 4th-order accurate wave propagation schemes: Mach 2.4, flat ocean surface and wave height of 3.75 m.

depth (m)	% change in +/− pressure peaks	
	$H = 0.0$ m	$H = 3.75$ m
0	0.7% / 0.8%	1.8% /−1.9%
100	1.1% /−0.9%	—

Since the waves dissipate at the same rate through a homogeneous domain, any percent difference between the 2nd- and 4th-order accurate simulations involving the two media must occur at the air-water interface. Perhaps the way the two different order accurate

schemes interact with the separate interface scheme is the cause of the small discrepancies, and the curvature just enhances the effect. In any case the accuracy difference in sound level is far less than 1 dB.

One more thing should be noted about the 2nd- and 4th-order accurate codes running on the same spatial grid (800 × 800 grid points). The user CPU time for the 2nd-order accurate code was on average 2:35:47 (hours:minutes:seconds), whereas for the 4th-order accurate code it was 3:28:49; again, all runs were performed on a Digital Equipment Corp. Alpha 3000/600 workstation. This is a difference of 0:53:02 CPU time, a time savings enough to make the 2nd-order code more attractive than the 4th, especially since the difference in accuracy seems to be negligible.

9.2. A More N-Shaped Waveform

The initial waveform for the simulations in Chapters 6–8 was a rounded sonic boom; as was seen in Chapter 2, actual sonic booms vary from an almost perfect N shape to a rounded N. A rounded sonic boom was chosen for the simulations because it lacks the discontinuities found in the shocks of a perfect N wave. It is here the goal to find differences, if any, by replacing the rounded sonic boom waveform with one that is slightly more N shaped. In doing this, it could show the code's flexibility in the shape of the initial waveform and whether or not the results with the rounded waveform are a good approximation to those that would be found using a different sonic boom shape.

9.2.1. New waveform

The more N-shaped sonic boom is calculated by multiplying a straight-line function. Eqs. (6.1) and (6.2), by an appropriate window. The desired waveform is one in which there are still no sharp discontinuities but the waveform appears more N-like than the rounded N wave. The window applied, constructed specifically to get the desired waveform, is

$$
\text{window}\,(x,z) = \begin{cases} 1 & |\zeta| \le 0.6\,(c\tau) \\ 0.152 + 1.37\cos\left[\frac{1.5(\zeta)}{c\tau}\right] & 0.75\,(c\tau) \ge |\zeta| > 0.6\,(c\tau) \\ 3.9\cos^2\left[\frac{1.5(\zeta)}{c\tau}\right] & c\tau \ge |\zeta| > 0.75\,(c\tau) \\ 0 & |\zeta| > c\tau \end{cases} \quad , \qquad (9.15)
$$

Where the variables are the same as in Chapter 6: ζ is the position on the computational grid, c is the speed of sound in air, and τ is half the duration of the sonic boom. This window appears in Appendix E as the Fortran function w2. This function is applied instead of the w function in the main code in Appendix B. The initial pressure p_{init} is then

$$p_{init}\,(x, z) = 1.5\,(\text{line}) \times 1.107\,(\text{window}) \ . \tag{9.16}$$

The straight line, personalized window, and the resulting more N-shaped wave appear in Fig. 9.1.
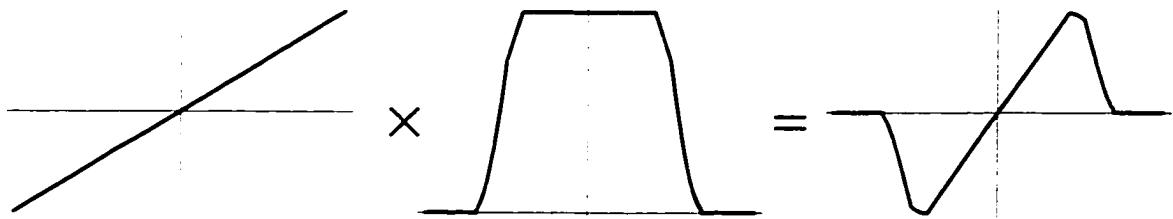
Figure 9.1: A more N-shaped sonic boom waveform than the rounded waveform in Fig. 6.1 and the functions used to construct it. These are plots of pressure as a function of space.

9.2.2. Computer program runs

Two runs were performed using this new initial waveform: the Mach number was 2.4, the peak pressure of the sonic boom was 50 Pa, the duration of the sonic boom was 300 ms, and a flat ocean surface and one with a trochoidal profile of wave height 3.75 m were applied. The flat surface run is compared to a simulation in Chapter 6, a rounded sonic boom interacting with a flat surface. The wavy ocean surface run is compared to one made in Chapter 7, a rounded sonic boom interacting with a wavy ocean surface with a trochoidal surface profile and a wave height of 3.75 m.

9.2.3. Results

The first set of results is shown in Table 9.2: these are the results for the flat ocean surface. Here the positive and negative peak pressures are compared for the run with the more N-shaped waveform to the one with the rounded sonic boom. In addition to examining the pressure values just under the surface, results are extracted from depths of 4, 16, and 64 m. The difference in pressure values are represented in three forms—pascal difference, percent difference, and sound level (peak dB) difference—in order to properly understand the percent changes at the greatest depth. Table 9.2 indicates that the greatest percent difference, 9.2%, is at 64 m, but this corresponds to only a 2.51 Pa difference or 0.8 dB (peak pressure level) difference, not a significant amount. It should be noted that the peak pressures of the initial waveforms are also slightly different: neither waveform has a peak pressure of exactly 50 Pa, as this is just the approximation. The difference in sound level of the initial waveform is actually about 0.1 dB; applying this to Table 9.2 would decrease the dB difference. Regardless of whether or not the small error is involved, it is determined that less than 1 dB in the peak pressure sound level is insignificant. This implies that the rounded N wave model is actually a good representation of at least one other sonic boom shape, quite likely other shapes as well.

Table 9.2: Changes in peak pressure (pascal, percent, and decibel) comparing the more N-shaped sonic boom runs to the rounded sonic boom runs: Mach 2.4, flat ocean surface.

depth (m)	difference between more N and rounded N (more N - rounded N) +/− peak pressures		
	Pa difference	% difference	dB difference
0	4.67 / −2.01	4.8% / 2.1%	0.4 / 0.2
4	2.04 / −0.69	2.3% / 0.8%	0.2 / 0.1
16	0.23 / −0.57	0.3% / 0.7%	0.0 / 0.1
64	1.58 / −2.51	6.0% / 9.2%	0.5 / 0.8

The second set of results is shown in Table 9.3; these are the results for the wavy ocean surface, wave height 3.75 m. The wavy surface results were here calculated to determine if the underwater pressure field with the focusing/defocusing at the surface is altered much by the shape of the initial waveform. Only the negative peak pressures are compared for this second set of runs. Since the curvature of the ocean surface focuses/defocuses the incoming waveform, several snapshots in time were examined in order to find just one snapshot for the more N-shaped wave simulation and one snapshot for the rounded N-wave simulation where either the positive peak pressure or negative peak pressure of the sonic boom waveform matched closely in value just under the surface. A fairly close match was made with the negative peak pressure; the positive peak pressures in this case were not at all matching, and this is why only the negative peak pressure results are shown in Table 9.3. Once the values were matched at the surface, several depths for each case could be examined.

As before, the difference in pressure values are represented in three forms: pascal difference, percent difference, and sound level (peak dB) difference. Table 9.3 indicates that the greatest percent difference, 8.5%, is at 64 m, but this corresponds to only a 2.35 Pa difference or 0.7 dB (peak pressure) difference, not a significant amount. Again, it is determined that less than 1 dB in the peak pressure sound level is insignificant; the rounded N wave model seems to be a good representation for other sonic boom shapes.

Table 9.3: Changes in negative peak pressure (Pa, %, dB) comparing the more N-shaped sonic boom runs to the rounded sonic boom runs; Mach 2.4, 3.75 m wave height, trochoidal profile.

depth (m)	difference between more N and rounded N (more N - rounded N) neg. peak pressure only		
	Pa difference	% difference	dB difference
0	−0.27	0.2%	0.0
4	−0.60	0.6%	0.1
16	1.92	−2.6%	−0.2
64	−2.35	8.5%	0.7

Tables 9.2 and 9.3 reveal that the underwater pressure fields obtained using a rounded sonic boom as the initial waveform are a good approximation to results that are obtained using a more N-shaped waveform. The simulation code is flexible in allowing different initial waveforms. Moreover, a rounded sonic boom seems to adequately represent at least one other waveform in simulating a sonic boom interacting with the ocean.

9.3. Summary of Verifications

By implementing a 4th-order accurate in space finite difference scheme in the computer code, the simulation results indicated that the 2nd-order accurate code is sufficient and preferred for the present studies. The 4th-order accurate solutions were less than 1 dB (peak pressure level) different from the 2nd-order accurate solutions, and they took 34% more user CPU time on the same grid. The runs made with a more N-shaped sonic boom waveform revealed that the rounded N wave simulation results were a good approximation to at least one other shaped waveform; the calculated percent changes and trends found in this work would most likely apply to other initial sonic booms as well.

Chapter 10.
Discussion and Conclusions

10.1. Summary

This dissertation has introduced concepts, provided literature reviews, given derivations, and described computational simulations of sonic booms interacting with a realistic ocean surface. The research for this work was motivated by a proposed high speed civil transport (HSCT) expected to fly supersonically over the world's oceans early next century. The sonic booms generated by these aircraft are an environmental concern, the noise potentially affecting marine mammals. The underwater sound caused by an HSCT-generated sonic boom was here calculated in order to quantify the sound levels which may be heard by marine mammals. As previous work has addressed the issue of sonic booms impinging upon a homogeneous ocean with a flat surface, this research accounts for the curvature of a wavy ocean surface and bubbles near the surface and the effects on the underwater sound that these realistic ocean features cause.

Chapter 2 reviewed sonic boom concepts, information extracted from books and articles that cover this topic. It was explained how a sonic boom is formed, starting with the initial disturbance at the airplane then the waveform's propagation through the atmosphere. A description of the sonic boom ground intercept and its associated noise was given along with illustrations of the waveform. Also, the underwater pressure field associated with a sonic boom impinging upon the water's surface was examined; conditions were given for the impinging waveform to penetrate the surface as a propagating wave versus an evanescent wave. For the proposed HSCT, it was concluded that the underwater pressure field was evanescent.

Literature concerning sound penetration from air into water was reviewed in Chapter 3. Initially, articles involving a general sound source in the air and its reception in the water were summarized. This was followed by reviews of articles that deal specifically with the sonic booms as the air source; historical applications involving analytical theories

and experiments and recent applications involving analytical and numerical theories and experiments were reported in these reviews.

Chapter 4 addressed the issue of underwater sonic boom noise. This chapter started by explaining the motivation to understand the underwater sound, summarizing the environmental concerns involving marine mammals. A description followed of the sound levels and a review of the recommendations for which sound metric to apply when describing a sonic boom. Applying several noise metrics, most being weighted for the hearing of humans rather than calculating a purely physical quantity, the underwater sound levels were numerically calculated using known analytical equations for the pressure values. As a result, the evanescent wave due to a sonic boom impacting a *flat* surface was analyzed. The fall-off of decibel levels for a broad range of metrics were presented as a function of depth, the results indicating that the high frequency content in the sonic boom decays rapidly. At the surface the sound level is approximately 94–131 dB re 20 μPa (120–157 dB re 1 μPa) where the initial sonic boom in the air has a 50 Pa peak pressure and a 300 ms duration; the exact decibel value depends on the metric applied. The sound level decreases over 40 dB for weighted metrics and about 13–18 dB for unweighted metrics at a depth of 128 m. Previous work looked only at the underwater pressure values or peak dB; this work calculated several descriptions of the underwater sonic boom noise, possibly giving a better idea of how marine mammals might hear the sound.

Chapter 5 explored the computational aspects of the simulations involved with the bulk of this research. After a brief outline of how to choose a computational method, the chosen finite difference method was formulated; two separate finite difference schemes were derived. The scheme applied to the air and water consists of a basic two-dimensional centered finite difference representation of the acoustic wave equation; the known horizontal derivative was easily derived, while deriving the vertical derivative, which accounts for a nonuniform grid, was slightly more difficult and is only sparsely shown in Ref. 25. The interface scheme derivation followed that in Ref. 56, an article from the geophysics community, and Ref. 59, although it was expanded here to account for a nonuniform grid in the vertical direction. Adding this special interface scheme allows a simulation to be performed with very large impedance changes in a computational domain, such as air to water. The combination of these air/water and interface finite difference schemes is unique to studying

sonic boom/ocean interaction and is essential for successful simulations. Known boundary conditions were also derived: a hard reflecting boundary applied to the left, top, and right sides of the computational domain was described along with an absorbing boundary applied to the bottom. Chapter 5 concludes with a stability analysis on the finite difference method.

The first results of the computational simulations resided in Chapter 6. As a check for the program's validity, the simulations in this chapter involved a *homogeneous* ocean with a *flat* surface. The initial waveform, a rounded sonic boom, was here formulated and illustrated. After listing the runs performed, the first full-field pressure results were graphically presented. The calculation of the intensity as a function of pressure was then introduced, equations derived mimicking an intensity probe analysis. Simulation results were then shown with the intensity vectors superimposed on the pressure field. Runs were performed which verified that the computer simulations correctly represented the physical problem: the sonic boom waveform penetrating the ocean surface was seen as an evanescent wave and a propagating wave, each case apparent when physics demanded it. The final section of Chapter 6 compared the simulated flat water results to previous theories. The theories of Sawyers[53] and Cook[12] were shown to apply to the simulations in that an evanescent wave was seen when the supersonic airplane was flying at a speed less than Mach 4.4. A quantitative comparison of the underwater sound level decay as a function of depth for the computational simulations to that for two analytical/numerical methods (one based on Sawyers' theory and one based on Cook's theory) showed that the decay rate of the evanescent sonic boom is approximately the same for each. Sparrow's conclusion[58] that larger Mach numbers are associated with greater penetration into a flat ocean was quantitatively verified in this chapter's simulations.

Chapter 7 went beyond a homogeneous ocean with a flat surface; it included computational simulations for a homogeneous ocean with a wavy surface. The concepts of wind-generated ocean waves and sonic booms interacting with these waves were discussed. Analytical formulations were given for ocean surface profiles with simple curvature and more complex curvature; the ocean wave parameters were chosen after investigating many sources. Several different wavy ocean surface profiles were successfully introduced into the code, some requiring special filters in order to maintain numerical stability. The results

qualitatively showed that as the sonic boom impinges upon a wavy ocean surface, its energy is focused in the ocean wave troughs and defocused over the ocean wave crests. The sonic boom was seen to penetrate the surface of the water as an evanescent wave, not a propagating wave. For a simple ocean surface, the results for a 50 Pa peak pressure, 300 ms duration sonic boom quantitatively showed that the peak pressure was changed at most 11.7% from the flat ocean surface results, the peak pressure sound level being augmented only 1 dB by the surface curvature. Two trends were discovered: 1.) increasing wind wave heights strengthens the focusing and defocusing of the evanescent pressure field, and 2.) increasing the Mach number strengthens the focusing and defocusing effects. It was also learned that the effects of curvature on the underwater pressure were stronger near the ocean surface than at greater depths.

Also in Chapter 7 an ocean surface with a complex profile revealed the largest effect due to curvature to be an 11.2% increase in peak pressure: this corresponds to less than 1 dB in sound pressure level. The first trend found with the simple surface was confirmed in the complex surface simulations, but the second was not. This inspired a study of wavelength comparisons, sonic boom effective wavelength to the wavy ocean surface wavelength, to find their relation to the effects due to curvature. This comparison provided a tool to help predict the magnitude of the effect due to simple curvature. The most significant effects were associated with the sonic boom effective wavelength being on the order of or less than the ocean surface wavelength. It was determined that this tool could not be applied to a complex surface, and that each complex case should be individually treated.

Chapter 8 also went beyond a flat homogeneous ocean. This time ocean surface waves remained suppressed and ocean bubbles were included. Concepts of ocean bubbles and their models were discussed. Two bubble models were successfully formulated, parameters chosen as a result of the information found in many sources, and implemented in the computational simulation. The first bubble model consisted of flat layers; each layer had a different sound speed, the lower speeds near the ocean surface and then increasing with depth. The second model consisted of one bubble plume also with layers. Results using both models indicate that the sonic boom penetrates the ocean surface only as an evanescent wave, and that bubbles only negligibly affect the pressure values in the evanescent pressure field, the magnitude of maximum change being less than 1 dB in sound pressure level.

Verification of the computational method and its results, beyond what was given in Chapter 6, was provided in Chapter 9. A 4th-order accurate finite difference scheme was derived here. Derivations for the known horizontal derivative (uniform grid) and for the vertical derivative (nonuniform grid) were presented: the 4th-order accurate formula for a derivative on a nonuniform grid involved a rather complicated derivation, the derivation and the solution not readily available in the literature. Results using a 4th-order accurate finite difference scheme indicated that the 2nd-order accurate code for a fixed spatial grid is sufficient and preferred to solve the sonic boom/ocean interaction problem. Less than a 1 dB difference in peak sound pressure level did not justify implementing the "more accurate" scheme which used up to 34% more CPU time than the 2nd-order accurate simulations. Runs made with a more N-shaped sonic boom waveform revealed that the rounded N wave simulation results are a good approximation to at least one of several alternate sonic boom shapes. Chapter 9, in addition to Chapter 6, confirmed that the methodology implemented in this research and the corresponding results provide a valid analysis of a sonic boom interacting with the ocean.

10.2. Revisiting the Basic Problem

Studying the effects that realistic ocean features have on an impinging sonic boom has revealed that the sonic boom itself is the noise source of concern: any underwater noise augmentation due to curvature on the ocean surface or bubbles beneath the surface is a minor effect. This returns us to the basic problem: a sonic boom impinging upon a homogeneous ocean with a flat surface. Having equations for calculating the acoustic pressure as well as having an understanding of the frequency content as a function of depth are important to analyzing the underwater sound field.

As was stated in Chapter 3, analytical theories involving a homogeneous ocean with a flat surface (also assuming the airplane is traveling at a speed less than Mach 4.4) were developed by Sawyers[53] and Cook.[12] Equations were formulated for pressure as a function of depth, the underwater acoustic disturbance being an evanescent wave. The underwater sonic boom noise was found to be a function of the sonic boom parameters (Mach number, peak pressure, duration) and the sound speeds in each medium.

In addition to the known equations for calculating the waveform, the frequency content can be analyzed. At the ocean surface, sonic booms vary in shape, some approximating an N wave, some having sharp high-pressure peaks, and some forming a rounded N wave, among other shapes. As these waveforms decay underwater, they quickly lose any sharpness in their shapes; their amplitudes decay with depth, and the duration lengthens. Examples of these decaying waveforms were shown in Chapter 6, Fig. 6.10. For a single frequency, the sound field decays as $e^{-|\omega|z/m}$, where ω is the angular frequency, z is the depth, and m is a function of the speed of the aircraft and the speed of sound in the ocean (please refer to Eq. (3.1), Sawyers' equation); this decay is also found in Eq. (2.6), the expression for the underwater evanescent pressure field for an incident plane wave of constant frequency. Regardless of the waveform shape, the frequency dependent exponential decay implies that higher frequencies decay more rapidly than the lower frequencies, as was discussed in Chapters 2, 3, 4, and 6.

The work described in Chapter 3 was heavily based on the simple two-fluid interface problem studied in introductory acoustics courses. It is somewhat remarkable after all the effects of bubbles and variations in ocean surface profiles have been accounted for that the theory based on the simple two-fluid interface does an excellent job of describing the sonic boom penetration into the ocean. The flat, homogeneous ocean model should be sufficient for most engineering predictions of the sonic boom underwater pressure field.

Using Eq. (3.1) developed by Sawyers or Eq. (3.2) developed by Cook, it is possible to calculate the pressure field as a function of depth: hypothetical HSCT parameters can be used as the input variables. A description of this underwater sonic boom noise, however, requires further analysis. If interested purely in the frequency content and not in the features of an incident waveform's shape, it is possible to follow the decay of each frequency component as a function of depth (please refer to Eq. (2.6)). Also, Chapter 4 explained metrics for describing the noise, some of the metrics revealing the changing frequency content and how the sonic boom might be perceived at various depths under the ocean. The methods in Chapter 4 can be applied to better evaluate the overall sonic boom noise for determining the impact on animals.

10.3. Limitations of the Computational Analysis and Suggestions for Future Work

Although this research has successfully investigated the effects of realistic ocean features on sonic boom noise penetration into the ocean using a two-dimensional simulation, there could be improvements to the code. One improvement would be an upgrade in the number of dimensions. Although this author believes that a three-dimensional simulation is not crucial in finding the strongest effects of wind-generated waves on an impinging sonic boom, at least one other author[8-10] has raised the issue that there may be some three-dimensional effects when solving the problem. The simulation code developed for this research can be upgraded to three dimensions; this could be a time-consuming task, especially since the code should then be written to run in parallel.

To also improve the simulation code, the boundary conditions could be upgraded. As it stands, the computational domain has three hard reflecting boundaries and one absorbing where it was absolutely crucial. All results had to be extracted from areas of the domain where reflections were not interfering with the solution. The part of the domain not contaminated by reflections could be expanded by applying absorbing boundary conditions all around. Another improvement involves the wavy ocean surface: as it stands in the simulation code, it is grid-block approximated. This limits the complexity of the surface profile, as was experienced when trying to implement a statistically generated ocean surface. To overcome the obstacles placed by the grid-block approximation, the grid could actually be warped in the air/water interface region such that the ocean surface is not ragged but rather a smooth curve. Another improvement to overcome instabilities due to the grid-block approximation might involve added dissipation. By adding dissipation to the finite difference schemes, it may be possible for the code to handle a ragged surface without applying any "smoothing" filters.

Other issues could be addressed with additions to or adaptations of the simulation code. One issue that might be addressed is finding effects on the underwater sonic boom noise due to structures. It is thought that when a sonic boom hits an object on the ocean surface, a ship, for example, this object will vibrate and radiate noise into the water. It may be possible to include such structures in the existing code and conduct simulations to determine

the underwater pressure field. Adapting the computer code, many other problems could be studied. Such problems might include the reciprocal of the one studied here: an underwater sound source and its reception in the air. The code could also accommodate different media, also with large impedance changes, or other interfaces that possess curvature: the ocean floor may be a good candidate if the proper equations are applied. Modifications could also account for nonlinear acoustic effects which would be present when a sonic boom is created *near* the ocean surface. The initial waveform could be one generated by something other than a supersonic aircraft, perhaps a missile or a helicopter, where underwater sound levels or waveform shapes may be of importance.

10.4. Concluding Remarks

A realistic evaluation of underwater noise due to a sonic boom generated by the proposed high speed civil transport in steady flight has been accomplished here through computational analysis. The computational method was constructed by gathering information from many different resources, crossing the boundaries of acoustics and extracting tools from ocean engineering and geophysics. The results found here indicate that the primary concern is the underwater noise from the actual sonic boom and not from any enhancement caused by a wavy ocean surface or bubbles near the surface. The task remains for marine biologists to determine if the predicted sound levels will affect marine mammals who happen to reside in or travel through an area insonified by these sonic booms.

References

1. Anderson. Dale A.. John C. Tannehill. and Richard H. Pletcher. *Computational Fluid Mechanics and Heat Transfer* (Hemisphere Publishing Corporation. New York. NY. 1984).

2. American National Standard Specification for Sound Level Meters, S1.4-1983. American Nat. Stand. Inst.. Inc.. February (1983).

3. Bascomb. Willard. *Waves and Beaches* (Anchor Books. Doubleday, New York, 1980). Chap. 3.

4. Bowles. Ann E.. "Responses of wildlife to noise." Chap. 8 of *Wildlife and Recreationists: Coexistence Through Management and Research* (Island Press. Washington, DC, 1995) edited by Richard L. Knight and Kevin J. Gutzwiller.

5. Brennen. Christopher E.. *Cavitation and Bubble Dynamics* (Oxford University Press. Inc.. New York, NY. 1995). Chap. 6.

6. Brown. D. and L.C. Sutherland. "Evaluation of outdoor-to-indoor response to minimized sonic booms." NASA CR-189643 (1992).

7. Brown, Joan. et al.. *Waves, Tides and Shallow-Water Processes*, edited by Gerry Bearman (The Open University. Paragom Press. Inc.. Tarrytown, NY, 1989). Chap. 1.

8. Cheng, H.K.. C.J. Lee. M.M. Hafez. and W.H. Guo. "Sonic boom propagation and its submarine impact: a study of theoretical and computational issues." AIAA Paper 96-0755. 34th Aerospace Sciences Meeting and Exhibit. Reno. NV (1996).

9. Cheng, H.K.. M.M. Hafez. C.J. Lee. and C.Y. Tang. "Assessment of an improved theory for further program development of sonic boom prediction and submarine impact study." Final report submitted to Armstrong Laboratories. OEBN. Noise Effects Branch. November 28 (1996).

10. Cheng, H.K. and C.J. Lee. "Submarine impact of sonic boom: a study comparing and reconciling results from two prediction approaches." Proceedings, Noise-Con 97, pp. 399–404 (1997).

11. Cleveland. Robin O. and David T. Blackstock. "Sonic boom rise time." *J. Acoust. Soc. Am.*, **96**(5), Pt. 2, p. 3275 (1994).

12. Cook. R.K.. "Penetration of a sonic boom into water." *J. Acoust. Soc. Am.*, **47**(5), Pt. 2, pp. 1430–1436 (1970).

13. Crighton. D.G.. A.P. Dowling, J.E. Ffowcs Williams. M. Heckl, and F.G. Leppington, *Modern Methods in Analytical Acoustics* (Springer-Verlag London Limited, Berlin. Germany, 1992), Chap. 21.

14. Cummings, William C.. "Sonic booms and marine mammals: informational status and recommendations." NASA CP-10132, pp. 219–231 (1993).

15. Davis, Sanford S., "Computational aeroacoustics using hyperbolic wave primitives," NASA CP-3300, pp. 27–45 (1995).

16. Dean, Robert G. and Robert A. Dalrymple, *Water Wave Mechanics for Engineers and Scientists* (Prentice Hall. Inc., Englewood Cliffs, NJ. 1984), Chap. 7.

17. Desharnais, Francine and David M.F. Chapman. "Underwater measurements of a sonic boom." Proceedings, Oceans 97 MTS/IEEE (1997).

18. Fahy, F.J., *Sound Intensity* (Elsevier Applied Science, London, 1989), pp. 91–93.

19. Farmer, David M. and Svein Vagle, "Waveguide propagation of ambient sound in the ocean-surface bubble layer." *J. Acoust. Soc. Am.*, **86**(5), pp. 1897–1908 (1989).

20. Frankel, Adam S. and Christopher W. Clark, "A review of the effects of sound on Cetaceans." *J. Acoust. Soc. Am.*, **96**(5), Pt. 2, p. 3250 (1994).

21. Fung, K.Y., Raymond S.O. Man, and Sanford Davis. "A compact solution to computational acoustics." NASA CP-3300, pp. 59-72 (1995).

22. Gaythwaite, John, *The Marine Environment and Structural Design* (Van Nostrand Reinhold, New York, 1981), Chap. 3.

23. Gionfriddo, Thomas A., "Quantification of sonic boom signature distortions from propagation through atmospheric turbulence." Masters Thesis, The Pennsylvania State University, University Park, PA (1992).

24. Green, G., "On the reflection and refraction of sound." *Trans. Camb. Phil. Soc.*, **6** (1838).

25. Hirsch, Charles, *Numerical Computation of Internal and External Flows: Volume 1: Fundamentals of Numerical Discretization* (John Wiley and Sons Ltd., New York, New York, 1988), Part II, also Chap. 1.

26. Hirsch, Charles, *Numerical Computation of Internal and External Flows: Volume 2: Computational Methods for Inviscid and Viscous Flows* (John Wiley and Sons Ltd., New York, New York, 1990), Part VI.

27. Hu, F.Q., M.Y. Hussaini, and J. Manthey, "Application of low dissipation and dispersion Runge-Kutta schemes to benchmark problems in computational aeroacoustics." NASA CP-3300, pp. 73-98 (1995).

28. ICASE/LaRC Workshop on Benchmark Problems in Computational Aeroacoustics (CAA), NASA Conference, October 1994, NASA CP-3300 (1995).

29. Intrieri, P. and G. Malcolm, "Ballistic range investigation of sonic-boom overpressures in water," *AIAA Journal*, **11**, pp. 510–516 (1973).

30. Jensen, Finn B., William A. Kuperman, Michael B. Porter, and Henrik Schmidt, *Computational Ocean Acoustics* (American Institute of Physics, Woodbury, NY, 1994), p. 38 and Chap. 7.

31. Khandekar, M.L., *Operational Analysis and Prediction of Ocean Wind Waves* (Springer Verlag, New York, 1989), Chap. 2.

32. Kryter, Karl D., *The Effects of Noise on Man*, 2nd ed. (Academic Press, Orlando, Florida, 1985).

33. Lamarre, Eric and W.K. Melville, "Void-fraction measurements and sound-speed fields in bubble plumes generated by breaking waves," *J. Acoust. Soc. Am.*, **95**(3), pp. 1317–1328 (1994).

34. Leatherwood, Jack D. and Brenda M. Sullivan, "Laboratory study of effects of sonic boom shaping on subjective loudness and acceptability." NASA TP-3269 (1992).

35. Lee, R.A. and J.M. Downing, Sonic booms produced by United States Air Force and United States Navy aircraft: Measured data, AL-TR-1991-0099, Air Force Systems Command, January (1991).

36. Leighton. T.G., *The Acoustic Bubble* (Academic Press Limited, San Diego, CA. 1994), Chap. 3.

37. Loewen. M.R. and W.K. Melville, "An experimental investigation of the collective oscillations of bubble plumes entrained by breaking waves," *J. Acoust. Soc. Am.*, **95**(3), pp. 1329-1343 (1994).

38. Lubard, Stephen C. and Patrick M. Hurdle, "Experimental investigation of acoustic transmission from air into a rough ocean," *J. Acoust. Soc. Am.*, **60**(5), pp. 1048-1052 (1976).

39. Macaluso. John A., "On transmission of sound from a monopole source through a finite, corrugated boundary between fluid media." Ph.D. Thesis, The Pennsylvania State University, University Park, PA (1970).

40. Maglieri. Domenic J. and Kenneth J. Plotkin. "Sonic boom," NASA RP-1258, 1, pp. 519-561 (1991) and appears as Chap. 10 of *Aeroacoustics of Flight Vehicles: Theory and Practice: Volume 1: Noise Sources* (Acoustical Society of America, Woodbury, NY. 1995) edited by Harvey H. Hubbard.

41. Malcolm, G. and P. Intrieri. "Ballistic range investigation of sonic-boom overpressures in water." AIAA Paper 72-657, presented at AIAA 5th Fluid and Plasma Dynamics Conference, Boston, MA. 26-28 June (1972).

42. Medwin, H., R.A. Helbig, and J.D. Hagy, Jr., "Spectral characteristics of sound transmission through the rough sea surface." *J. Acoust. Soc. Am.*, **54**(1), pp. 99-109 (1973).

43. Meecham. William C., "Point source transmission through a sinusoidal ocean surface," *J. Acoust. Soc. Am.*, **64**(5), pp. 1478-1481 (1978).

44. Pierce. Allan D., *Acoustics: An Introduction to Its Physical Principles and Applications* (Acoustical Society of America, Woodbury, NY. 1989), Chap. 11, also p. 67.

45. Pierce. Allan D., "Turbulence scales, rise times, caustics, and the simulation of sonic boom propagation." NASA CP-3335, pp. 1-19 (1996).

46. Plotkin, Kenneth J. and Louis C. Sutherland. "Sonic boom: prediction and effects," AIAA Professional Studies Series. Tallahassee. Florida. 25-26 October (1990).

47. Poisson-Quinton. Philippe. "Future SSTs, a European approach," *Aerospace America*. pp. 39-43. September (1994).

48. Rayleigh. J.W.S., "Hydrodynamical notes." *Philos. Mag.*, **21**, pp. 177-195 (1911).

49. Richardson. W. John, Charles R. Greene, Jr., Charles I. Malme, and Denis H. Thomson, *Marine Mammals and Noise* (Academic Press, Inc., San Diego, CA, 1995).

50. Roach, Patrick J., *Computational Fluid Dynamics* (Hermosa Publishers, Albuquerque, NM, 1972), Chaps. 3 and 5.

51. Rochat, Judith L. and Victor W. Sparrow, "Sound levels under the ocean due to sonic boom penetration," Proceedings. Inter-Noise 95, pp. 967-970 (1995).

52. Rochat, Judith L. and Victor W. Sparrow, "Two-Dimensional focusing of sonic boom noise penetrating an air-water interface," *AIAA Journal*, **35**(1), pp. 35-39 (1997) also AIAA Paper 96-1751. AIAA/CEAS 2nd Aeroacoustics Conference, State College, PA (1996).

53. Sawyers, K.N., "Underwater sound pressure from sonic booms," *J. Acoust. Soc. Am.*, **44**(2), pp. 523-524 (1968).

54. Schultz, Theodore J., *Community Noise Rating*, 2nd ed. (Applied Science Publishers, New York, New York, 1982), pp. 110–111.

55. Shepherd, Kevin P. and Brenda M. Sullivan, "A loudness calculation procedure applied to shaped sonic booms," NASA TP-3134 (1991).

56. Sochacki, J.S., J.H. George, R.E. Ewing, and S.B. Smithson, "Interface conditions for acoustic and elastic wave propagation," *Geophysics*, **56**(2), pp. 168-181 (1991).

57. Sorensen, Robert M., *Basic Wave Mechanics: For Coastal and Ocean Engineers* (John Wiley and Sons, Inc., New York, NY, 1993), Chap. 6.

58. Sparrow, Victor W., "The effect of supersonic aircraft on the penetration of sonic boom noise into the ocean," *J. Acoust. Soc. Am.*, **97**(1), pp. 159-162 (1995).

59. Sparrow, Victor W., "Evanescent wave penetration of sonic boom noise into the ocean," Second International Conference on Theoretical and Computational Acoustics, Honolulu, HI, August 22 (1995).

60. Sparrow, V., and T. Ferguson, "Penetration of shaped sonic boom noise into a flat ocean," AIAA Paper 97-0486, 35th Aerospace Sciences Meeting and Exhibit, Reno, NV (1997).

61. Sparrow, Victor W. and Richard Raspet, "Absorbing boundary conditions for a spherical monopole in a set of two-dimensional acoustics equations," *J. Acoust. Soc. Am.*, **87**(6), pp. 2422-2427 (1990).

62. Stevens, S.S., "Perceived level of noise by Mark VII and decibels(E)," *J. Acoust. Soc. Am.*, **51**(2), Pt. 2, pp. 575-601 (1972).

63. Stokes, G.G., "Supplement to a paper on the theory of oscillatory waves," Math. and Phys. Pap. (Cambridge University Press, 1880), **1**, pp. 314-326.

64. Thorpe, S.A., "On the clouds of bubbles formed by breaking wind-waves in deep water, and their role in air-sea gas transfer," *Phil. Trans. R. Soc. Lond. A*, **304**, pp. 155-210 (1982).

65. Thorpe, S.A., "The effect of Langmuir circulation on the distribution of submerged bubbles caused by breaking wind waves," *J. Fluid Mech.*, **142**, pp. 151-170 (1984).

66. Urick, R.J., "Noise signature of an aircraft in level flight over a hydrophone in the sea," *J. Acoust. Soc. Am.*, **52**(3), Pt. 2, pp. 993-999 (1972).

67. Vartabedian, Ralph, "Launching supersonic dreams," *Los Angeles Times*, June 1 (1994).

68. Waters, J., "Penetration of sonic boom energy into the ocean: an experimental simulation," in *Noise and Vibration Control Engineering*, edited by M. Crocker, *Proceedings of the Purdue Noise Control Conference*, pp. 554-557, 14-16 July (1971).

69. Waters, J., and R.E. Glass, "Penetration of sonic boom energy into the ocean: an experimental simulation," Hydrospace Research Corp. Final Report on Contract FA70-WAI-185, HRC TR 288, June (1970), available from NTIS/DTIC as AD 711 963.

70. Whitham, G.B., "The flow pattern of a supersonic projectile," *Communications on Pure and Applied Mathematics*, **5**, pp. 301-348 (1952).

71. Wilhite, Alan W. and Robert J. Shaw, "HSCT research picks up speed," *Aerospace America*, pp. 24-29, 41, August (1997).

72. Williams, Louis J., "HSCT research gathers speed," *Aerospace America*, pp. 32-37, April (1995).

73. Wolfram Research, Inc.. MATHEMATICA, Version 2.2 (Wolfram Research. Champaign, IL, 1993).

74. Young, Robert W., "Penetration of sonic booms into the ocean," *J. Acoust. Soc. Am..* **44**, p. 392 (1968).

75. Young, Robert W., "Day-Night average sound level (DNL) and sound exposure level (SEL) as efficient descriptors for noise compatibility planning," Proceedings, Inter-Noise 89. pp. 1289–1292 (1989).

Appendix A.
Mathematica Code for Weighted Sound Levels

Sound Level Functions:

Functions for N waves and Sound Exposure Levels

■ Construction of N wave

Function Nwavefn is produces a line between 2 points

```
Nwavefn[t_,time_,press_] :=
    ((t-(time/251))((-press)-press))/(time) + (press)
```

This next function Nwave2 allows the input of the duration
and peak pressure of an N-wave
The output is a list of data points, each pt being (time,pressure ampl)

```
Clear[Nwave2]

Nwave2[duratn_,press_] := Nwave2pt =
    Join[{{0.0,0.0},{(duratn/251),0.0}},
    Table[{t,Nwavefn[t,duratn,press]},
    {t,(duratn/251),duratn+(duratn/251),(duratn/251)}],
    Table[{t,0.0},{t,duratn+(duratn/251),
    duratn+(2 (duratn/251)),(duratn/251)}]];
    NoSamples = 256;
```

Function Nwave1 inputs duration and peak pressure of an N-wave
then outputs a list of just the pressure values over time
This function should be used if you are going to do a Fourier transform
- ref pressure is for air

```
Nwave1[duratn_,press_] := {
    duration = duratn;
    deltat = duratn/251;
    Nwave1pt =
    Join[{0,0},
    Table[Nwavefn[t,duratn,press],
    {t,deltat,duratn+deltat,deltat}],
    Table[0,{t,duratn+deltat,
    duratn+(2 deltat),deltat}]];
    pref = 20 10^(-6);
    NoSamples = 255;
    signaltime = duration + 2 deltat;
    T = signaltime/NoSamples;
    fs = 1/T;
    deltaf = fs/NoSamples;}
```

■ Fourier transform

Functions for changing pressure vs time to
unweighted pressure dB vs frequency (Hz)

Function fftlistPos produces a list including just positive frequencies

```
fftlistPos[list_] := {fftpressdB = 20 N[Log[10,
    ((Abs[N[Fourier[list]]])/
    (Sqrt[2] pref))]];
    fftpos =
    Table[fftpressdB[[i+1]],
    {i,1,(Quotient[NoSamples,2]-1)}]
    ;}
```

Function fftcomplex inputs a list of peak pressures and takes the
Fourier transform

```
fftcomplex[list_] := {fftpressz = 20 N[Log[10,
    ((Fourier[list])/
    (Sqrt[2] pref))]];
    fftpos =
    Table[{fftpressz[[i+1]]},
    {i,0,(Quotient[NoSamples,2]-1)}]
    ;}
```

Function invfft takes in a list of discete values, starting with zero
frequency value; it outputs a list of transformed values in time domain

```
invfft[list_] := {templist =
    N[Abs[InverseFourier[list]]];
    invfftlist = Flatten[templist];}
```

■ Weightings

C weighting, ANSI S1.4 (1983)

weighting constants

```
f1 = 20.598997;
f2 = 107.65265;
f3 = 737.86223;
f4 = 12194.22;
f5 = 158.48932;
K1 = 2.242881 10^16;
K2 = 1.025119;
K3 = 1.562339;
```

C weighting for frequency f (attenuation in dB)

```
WC[f_] := {
    10 Log[10,
    (K1 f^4) / ((f^2+f1^2)^2 (f^2+f4^2)^2)]}
```

C weighting unweighted pressure dB (complex value)

Function dBCpolar attenuates the magnitude of the complex number
and creates a list of polar comlex numbers C-weighted

```
dBCpolar[list_] := dBClist =
    Table[
    (Abs[list[[i]]] +
    WC[(i deltaf)]) Exp[I Arg[list[[i]]]],
    {i,1,(Length[list])}]
```

A weighting, ANSI S1.4 (1983)

A weighting for frequency f (attenuation in dB)

```
WA[f_] :=
    10 Log[10,
    ((K3 f^4) / ((f^2 + f2^2) (f^2 +
    f3^2)))] + WC[f]
```

A weighting unweighted pressure dB (complex value)

Function dBApolar attenuates the magnitude of the complex number
and creates a list of polar comlex numbers A-weighted

```
dBApolar[list_] := dBAlist =
    Table[
    (Abs[list[[i]]] +
    WA[(i deltaf)]) Exp[I Arg[list[[i]]]],
    {i,1,(Length[list])}]
```

■ Converting dB and pressure values

changing dB into pressure

```
pressfromdB[list_] := p =
    N[10^(list/20) Sqrt[2] pref]
```

changing peak pressure into dB

```
dBvalue[list_] := dB = 20 N[Log[10,
    (Abs[list]/(Sqrt[2] pref))]]
```

changing rms pressure into dB

```
dBvaluerms[list_] := dB = 20 N[Log[10,
    (Abs[list]/pref)]]
```

■ Manipulation function

Function Pos2Full:
changing positive freq valued list of dB vs freq back to pressure
vs pos and neg freqs; generates a list that can be transformed back
to the time domain

```
Pos2Full[list_] := {
    presslist = pressfromdB[list];
    revlist = Reverse[presslist];
    addlist = Conjugate[revlist];
    list2xform = Join[presslist,addlist];
    }
```

■ Signal Level

use preSEL and SEL together or use SELtheorhalf

Function preSEL takes a list of dB values (can be weighted), finds the
maximum dB value, then makes a list of dB values including the
maximum value and 10 dB down before and after the max value in time

```
preSEL[list_] := {
    maxdB = Max[list];
    Pos = Position[list, maxdB];
    Pos1 = Pos[[1]];
    maxPos = Pos1[[1]];
    LofT = {maxdB};
    j = 0;
    While[True,
        j = j + 1;
        If[(maxPos-j) > 0 &&
            (maxdB - list[[maxPos-j]]) <= 10,
            LofT = Prepend[LofT,list[[maxPos-j]]],
            Break[]];
    ];
    j = 0;
    While[True,
        j = j + 1;
        If[(maxPos+j) <= Length[list] &&
            (maxdB - list[[maxPos+j]]) <= 10,
            LofT = Append[LofT,list[[maxPos+j]]],
            Break[]];
    ];
}
```

SEL takes a list of dB values (can be weighted) and outputs an overall sound exposure level

```
SEL[list_] := {sel =
    10 Log[10, deltat Sum[10^(list[[j]]/10),
    {j,1,Length[list]}]];sel}
```

SELtheorhalf uses rms pressures (but reads in peak), eq 2.16 Schultz and also subtracts 3 dB since we should be using only half the energy since the duration of the N wave is long enough so that the shocks are distinguished as two separate bangs.

```
SELtheorhalf[list_] := N[10 Log[10,
    deltat Sum[(list[[j]] / Sqrt[2])^2 / pref^2,
    {j,1,Length[list]}]] - 3]
```

SELtheor uses rms pressures, eq 2.16 Schultz, reads in peak press

```
SELtheor[list_] := N[10 Log[10,
    deltat Sum[(list[[j]] / Sqrt[2])^2 / pref^2,
    {j,1,Length[list]}]]]
```

Runs for table information:

■ Pressure as a function of depth and variables used for all calculations

■ variables

```
cw = 1500. ;
cair = 343.;
pref = 20 10^(-6);
T = .300; (* duration *)
ppeak = 100; (* Pa pressure doubled at surface *)
deltatau = N[5/14783];
deltat = deltatau T;

NoSamples = 16384;
signaltime = 16383 deltat;
period = signaltime/NoSamples;
fs = 1/period;
deltaf = fs/NoSamples;

fs

9855.93
```

■ functions for pressure
(from Vic Sparrow; based on Sawyers' theory)

```
(* We know what m is from the plane speed and speed of
    sound in water: *)

 sawyerm := V / Sqrt[1-(V/cw)^2]

Clear[sawyerm2]

sawyerm2= ReleaseHold[sawyerm /. V-> Hold[Times[cair, M]]]
```

$$\frac{343. \, M}{Sqrt[1 - 0.0522884 \, M^2]}$$

```
Clear[press]

press[xi_,zeta_,tau_,p0_] :=  p0 (1/Pi) (1/Sqrt[2]) (
     (2 tau + 2 xi - 1) ArcTan[(tau + xi - 1)/zeta] +
     - (2 tau + 2 xi - 1) ArcTan[(tau + xi)/zeta] +
     zeta Log[(zeta^2 + (tau + xi)^2)/
                         (zeta^2 + (tau + xi - 1)^2) ]) ;

begline[t_,deltime_,press_] :=
     (t deltime)(press)/(799 deltime)


(* these are the functions that produce peak press *)

Clear[NwaveDepth1]

NwaveDepth1[depth_] := {
     pickzeta = (depth/( sawyerm2 T) /. M->pickmach);
     temp1 =
     Table[N[Sqrt[2] press[0,pickzeta,tau,ppeak]],
          {tau, -2, 3,deltatau}];
     temp2 =
     Table[begline[t,deltat,temp1[[1]]],
          {t,0,799}];
     endline =
          -Reverse[temp2];
     Ndepth1pt = Join[temp2,temp1,endline];}
```

■Mach 2.4, 16384 samples, N wave: peak pressure at surface 100 Pa,duration 300 ms

```
pickmach = 2.4;
```

output in the order of peak, SEL(U), SEL(C), SEL(A) (dB)

■ Examples:

□ depth of 0 meters

```
Nwave1[.300,100];
Max[Nwave1pt]
dBvalue[Max[Nwave1pt]]
```

```
100.
```

```
130.969
```

```
Nwave1[.300,100];
fftcomplex[Nwave1pt];
Pos2Full[fftpos];
invfft[list2xform];
SELtheorhalf[invfftlist]
```

117.971

```
Nwave1[.300,100];
fftcomplex[Nwave1pt];
dBCpolar[fftpos];
Pos2Full[dBClist];
invfft[list2xform];
SELtheorhalf[invfftlist]
```

107.966

```
Nwave1[.300,100];
fftcomplex[Nwave1pt];
dBApolar[fftpos];
Pos2Full[dBAlist];
invfft[list2xform];
SELtheorhalf[invfftlist]
```

94.4405

■ depth of 16 meters

```
NwaveDepth1[16.0];
Max[Ndepth1pt]
dBvalue[Max[Ndepth1pt]]
```

58.8602

126.366

```
fftcomplex[Ndepth1pt];
Pos2Full[fftpos];
invfft[list2xform];
SELtheorhalf[invfftlist]
```

115.258

```
dBCpolar[fftpos];
Pos2Full[dBClist];
invfft[list2xform];
SELtheorhalf[invfftlist]
```

87.8946

```
dBApolar[fftpos];
Pos2Full[dBAlist];
invfft[list2xform];
SELtheorhalf[invfftlist]
```

38.2229

Appendix B.
Two-Dimensional Second-Order Computer Code

This appendix contains the Fortran code for the two-dimensional simulation of a sonic boom interacting with the ocean. The finite difference schemes used for the wave propagation are 2nd-order accurate. Also contained in this appendix are the global variable file. the input files. and the output file: these files appear in the stated order following the wave propagation code.

Fortran code for wave propagation. prop2o.f:

```
c  ********************************************************************
c
c prop2o.f
c
c  ********************************************************************
c  ********************************************************************
c This program is J. Rochat and V. Sparrow's Fortran code for modeling
a
c  sonic boom propagating from air, reflecting from water, and creating
c  a penetrating evanescent wave.
c
c The Computational Domain:
c
c The top, right, and left boundaries of the rectangular domain
c  are rigid, and the bottom boundary is absorbing.
c
c Inhomogeneity is represented by a varying matrix c(i,j) to
c  represent the speed of sound, and the varying matrix rho(i,j) to
c  represent the density variations.
c
c This program was last modified on 3/31/97
c
c  ********************************************************************
c  ********************************************************************
c
c ** Declare global variables.
c Read in the common block.
```

```
          include 'metergrid.common'
c
c ** Declare local variables:
c  general variables
          integer i, j, k, itime, nosteps
          integer ocean
          real w
          real time, maxtime, mach, hfactor, vfactor, slope, thetai
          real const,halfdelz,diff,densetop,rangepts
          real layers,newrange,topdiff,botdiff
          real layer(10),diffb(10),bbot(10)
          real sumdelz,range
          real pmax,twotau,tau
          real avp1,avp2
c
c  variables for absorbing boundary condition
c cosbotangle - cosine of the bottom angle.
c sinbotangle - sine of the bottom angle.
c  The bottom angle is the angle away from the -z axis.
c  I.e., straight down is a bottom angle of zero.
          real cosbotangle, sinbotangle
c
c  variables for use in swell function
c eta - the height or depth of the ocean above or below ambient state.
c height - height of ocean swell.  This is the crest to trough measurement.
c distance - distance along the ocean at its ambient state.
c sampdist - the sampling distance.
c type - the type of wave we want.
c amplitude - the "amplitude" of the swell. For a sinusoidal swell,
c                the amplitude of the swell is one half of the height.
c lambda - the "wavelength" of the swell
          real swellfunc
          real eta, height, distance, sampdist, amplitude, lambda
          integer type
c
c  variables for input and output files
          character*9 inputfile
          character*10 outputfile
c
c opening files for input and output
          open(unit=1, file='inputfile', status='old')
          open(unit=2, file='outputfile', status='new',
     &        form="formatted")
          open(unit=3, file='bublayers', status='old')
c
c
c ** Outline of Program:
```

```
c  - read in bubble layer information
c  - define constants
c  - choose ocean surface type and wave height
c  - calculating the bubble layer profiles including 1 plume
c  - set grid values for entire domain: medium 1, interface, or medium
2
c  - choose Mach number and sonic boom parameters
c  - calculate initial pressure for the entire domain
c  - calculate pressure at later times
c
c
c *** Read in bubble layer information
c
c  indicate whether or not to include bubble layers in the program
        read(unit=1, fmt=4000)bubbles
        write(unit=2, fmt=3000)' '
        write(unit=2, fmt=3010)'**********************************'
        write(unit=2, fmt=3000)' '
        write(unit=2, fmt=4001)'1 for including bubbles
     &or 2 for not'
        write(unit=2, fmt=4002)'include bubble layers: ',bubbles
c
c  indicate whether or not to insert a bubble plume
        read(unit=1, fmt=5000)bubplume
        write(unit=2, fmt=3000)' '
        write(unit=2, fmt=5001)'1 for including a bubble plume
     &or 2 for not'
        write(unit=2, fmt=5002)'include a bubble plume: ',bubplume
c
c  read in bubble layer variables
          read(unit=3, fmt=4010)numlay
          write(unit=2, fmt=3000)' '
          write(unit=2, fmt=4011)'number of bubble layers: ',numlay
          layers = 0.
          do i=1,numlay
c  number of meters in depth of bubble layer
            read(unit=3, fmt=4008)layer(i)
c  speed of sound in bubble layer
            read(unit=3, fmt=4004)cb(i)
c  density of bubble layer
            read(unit=3, fmt=4004)rhob(i)
c
          write(unit=2, fmt=3000)' '
          write(unit=2, fmt=4005)'thickness of bubble layer #',i,':
',
     &                  layer(i),' m'
          write(unit=2, fmt=4006)'  speed of sound = ',cb(i),' m/s'
```

```
                     write(unit=2, fmt=4007)'  density = ',rhob(i),' kg/m^3'
                     layers = layers + layer(i)
                  end do
c
              write(unit=2, fmt=3000)' '
              write(unit=2, fmt=3010)'***********************************'
c
c
c *** Define constants
c
c Set the depth of the air-water interface.
              depthl = -340.
c Set the depth of the bottom of each bubble layer
              if (bubbles.eq.1) then
                 bbot(1) = depthl - layer(1)
                 do i=2,numlay
                    bbot(i) = bbot(i-1) - layer(i)
                 end do
              endif
c
c Set up initial constants.
c
c  maxtime - maximum time allow for pressure calculations
c  deltax  - grid block size in horizontal direction
c  deltaz  - grid block size in vertical direction, not near interface
c  delzinter - grid block size in vertical direction, near interface
c  range   - physical range of dense grid (around interface)
c  newrange - newrange is the physical range of the dense grid when
c             bubble layers are included
c  densetop  - physical location of top of dense grid range
c  ranget  - grid number representing the top of dense area
c  rangepts  - number of grid points in physical range
c  rangeb   - grid number representing the bottom of dense area
              maxtime= 0.794
              deltax = 1.0
              oodelxsq = 1/(deltax*deltax)
              deltaz = deltax
              delzinter = 0.25*deltaz
              halfdelz = .5*delzinter
              range = 6.
              if (bubbles.eq.1) then
                if (bubplume.eq.1) then
                   newrange = 6. + layers + numlay*delzinter
                elseif (bubplume.eq.2) then
                   newrange = 6. + layers
                end if
              else
```

```
              newrange = 6.
            endif
            densetop = depthl + range/2.
            ranget = zmax + int(densetop)
            rangepts = int(newrange/delzinter)
            rangeb = ranget - rangepts
c
c This loop assigns the vertical grid block size
c  - makes the grid dense around the interface region
            do j = zmax,1,-1
              if ((j.gt.ranget).or.(j.le.rangeb)) then
                delz(j) = deltaz
              else
                delz(j) = delzinter
              endif
            end do
c
c This loop assigns each grid point a physical location in the
c  vertical direction
c  Z coordinates:  as j goes from 1 to zmax, zpos goes
c  from -(zmax-1)*deltax to 0
            sumdelz = 0.
            do j=zmax,1,-1
              zpos(j) = -sumdelz
              sumdelz = sumdelz + delz(j)
            end do
c
c This loop assigns each grid point a physical location in the
c  horizontal direction
c  X coordinates:  as i goes from 1 to xmax, xpos goes
c  from 0 to (xmax-1)*deltax
            do i=1, xmax
                    xpos(i)= deltax*(i-1)
            end do
c
            write(unit=2, fmt=3000)' '
            write(unit=2, fmt=2050)'x total distance is approximately
   ,
     &deltax*xmax
            write(unit=2, fmt=2050)'z total distance is approximately
   ,
     &sumdelz
c
c The speed of sound will have a max of 1500, so must account for
this.
c  also, must use the smallest delta value on grid
            deltat = 0.7*delzinter/1500.
```

```
            deltsq = deltat*deltat
c
c Initialize c and rho
c   medium 1
            rho1 = 1.21
            c1 = 343.
c   medium 2 (ocean without bubbles)
            rho2 = 1000.
            c2 = 1500.
c
c Constants to be used in finite difference routine
            c1sqtsq = (c1*c1)*deltsq
            c2sqtsq = (c2*c2)*deltsq
            a1 = 1./(rho1*c1*c1)
            a2 = 1./(rho2*c2*c2)
            b1 = 1./rho1
            b2 = 1./rho2
            if (bubbles.eq.1) then
              do i=1,numlay
                csqtsq(i) = (cb(i)*cb(i))*deltsq
                abub(i) = 1./(rhob(i)*cb(i)*cb(i))
                bbub(i) = 1./rhob(i)
              end do
            endif
c
c
c *** Choose ocean surface type and wave height
c
c Read in from input file the choice of either a flat ocean
c   surface (1) or an ocean surface with waves (2)
            read(unit=1, fmt=1000)ocean
            write(unit=2, fmt=3000)' '
            write(unit=2, fmt=3010)'*********************************'
            write(unit=2, fmt=3000)' '
            write(unit=2, fmt=2000)'1 for a flat ocean surface
     &or 2 for wavy'
            write(unit=2, fmt=2010)'ocean type: ',ocean
c
c Set the amplitude and wavelength
c Read in from input file the wave height
c   (height is defined as the distance from crest
c   to trough)
            read(unit=1, fmt=1010)height
            write(unit=2, fmt=3000)' '
            write(unit=2, fmt=2001)'the height of the swell waves'
            write(unit=2, fmt=2020)'height = ',height,' meters'
            amplitude=0.5*height
```

```
c
c  lambda is twenty times the height of the swell.  This is the
c  common naval architects' definition.
c  Seven times the height is the minimum permissible wavelength
         lambda= 20.*height
c
c Read in from file the choice of ocean profile type:
c  sinusoidal (1) or trochoidal (2) waves
c  or combination #1, sine wave .5 wavelength of trochoidal (3)
c  or combination #2, sine wave .3 wavelength of trochoidal (4)
         read(unit=1, fmt=1020)type
         write(unit=2, fmt=3000)' '
         write(unit=2, fmt=2002)'type of ocean wave profile:'
         write(unit=2, fmt=2003)'(1) sinusoidal, (2) trochoidal,
      &(3) combination #1:
      &sine wave .5 wavelength '
         write(unit=2, fmt=2004)'of trochoidal, (4) combination #2:
      &sine wave .3 wavelength of trochoidal'
         write(unit=2, fmt=2030)'type: ',type
c
c This loop calculates the ocean surface shape according to the
c  wave type and height
         do i=1, xmax
           if (ocean.eq.2) then
             zinter(i)=swellfunc(xpos(i),amplitude,lambda,type)
      &                +depthl
           elseif (ocean.eq.1) then
             zinter(i)=depthl
           else
             print *,'invalid type of ocean'
           endif
         end do
c
c This loop smooths any ocean wave peak that has a single maximum
or
c  minimum grid point
         do i=2, xmax-1
           if ((zinter(i).gt.zinter(i-1)).and.
      &        (zinter(i).gt.zinter(i+1))) then
             zinter(i) = zinter(i-1)
           elseif ((zinter(i).lt.zinter(i-1)).and.
      &        (zinter(i).lt.zinter(i+1))) then
             zinter(i) = zinter(i-1)
           endif
         end do
c
c This loop adjusts the pixels near the peak of the ocean waveform;
```

```
c  the adjustment takes place only in the case of ocean profile type
3:
c  the trochoidal with 1/2 wavelength sinusoidal.
        if(type.eq.3) then
        do i=4, xmax-3
          if((nint(zinter(i)/delzinter).eq.
     &              nint(zinter(i+1)/delzinter)).and.
     &          (nint(zinter(i)/delzinter).eq.
     &          nint(zinter(i+2)/delzinter))) then
            if((nint(zinter(i-1)/delzinter).lt.
     &              nint(zinter(i)/delzinter)).and.
     &              (nint(zinter(i-1)/delzinter).gt.
     &          nint(zinter(i-2)/delzinter))) then
              if(nint(zinter(i-3)/delzinter).eq.
     &              nint(zinter(i-2)/delzinter)) then
                zinter(i-2) = zinter(i-1)
              endif
            endif
            if((nint(zinter(i+3)/delzinter).lt.
     &              nint(zinter(i+2)/delzinter)).and.
     &              (nint(zinter(i+3)/delzinter).gt.
     &          nint(zinter(i+4)/delzinter))) then
              if(nint(zinter(i+5)/delzinter).eq.
     &              nint(zinter(i+4)/delzinter)) then
                zinter(i+4) = zinter(i+3)
              endif
            endif
          endif
        end do
        endif
c
c *** Calculating the bubble layer profiles including 1 plume
c
        if (bubplume.eq.1) then
c the horizontal value where the middle of the plume resides
c        midplume = 400
        midplume = 270
        laydepth = 0
c calculating the profile for each layer
        do layno=1,numlay
          laydepth = laydepth + layer(layno)
          call plume
c
c This loop smooths any peak in the profile that has a single
c  maximum or minimum grid point
          do i=2, xmax-1
            if ((pprofile(i).gt.pprofile(i-1)).and.
```

```
     &                   (pprofile(i).gt.pprofile(i+1))) then
                         pprofile(i) = pprofile(i-1)
                    elseif ((pprofile(i).lt.pprofile(i-1)).and.
     &                   (pprofile(i).lt.pprofile(i+1))) then
                         pprofile(i) = pprofile(i-1)
                    endif
                end do
c

                do i=1,xmax
                    layprof(layno,i) = pprofile(i) + depthl
                end do
              end do
            endif
c
c
c *** Set grid values for entire domain
c This loop assigns grid values according to media and assigns
c  speed of sound values for the grid
c
c * Ocean without bubble layers
c         medium 1, interface, or medium 2
          if (bubbles.eq.2) then
c
          do i=1, xmax
            do j=1, zmax
            diff = zpos(j)-zinter(i)
c   medium 1
c     a grid value of 1 implies you are in medium 1
                if (diff.gt.halfdelz) then
                    grid(i,j) = 1
                    c(i,j) = c1
                    rho(i,j) = rho1
                    a(i,j) = a1
                    if (j.eq.ranget) then
                      grid(i,j) = 4
                    endif
c   medium 2
c     a grid value of 2 implies you are in medium 2
                elseif (diff.lt.-halfdelz) then
                    grid(i,j) = 2
                    c(i,j) = c2
                    rho(i,j) = rho2
                    a(i,j) = a2
                    if (j.eq.rangeb) then
                      grid(i,j) = 5
                    endif
c   at the interface
```

```
c     a grid value of 3 implies you are in the interface range
            else
                grid(i,j) = 3
                c(i,j) = c2
                rho(i,j) = rho2
c     percentage of the grid block surrounding the grid point which
c     is above the interface (this is an approximation)
                percntabv = (diff + halfdelz)/delzinter
c     this a function is used in the fd routine at the interface
                a(i,j) = percntabv*a1 + (1.-percntabv)*a2
            endif
          end do
        end do
c
c * Assigning grid values now with bubble layers included
c       air, ocean with bubble layers, ocean without bubble layers,
c       interfaces
        elseif (bubbles.eq.1) then
        do i=1, xmax
          do j=1, zmax
            diff = zpos(j)-zinter(i)
            do k=1, numlay
              if (bubplume.eq.2) then
                diffb(k) = zpos(j)-bbot(k)
              elseif (bubplume.eq.1) then
                diffb(k) = zpos(j)-layprof(k,i)
              endif
            end do
c air
c     a grid value of 1 implies you are in the air
            if (zpos(j).ge.(zinter(i)-halfdelz)) then
              if (diff.gt.halfdelz) then
                grid(i,j) = 1
                c(i,j) = c1
                rho(i,j) = rho1
                a(i,j) = a1
                if (j.eq.ranget) then
                  grid(i,j) = 4
                endif
              else
c     a grid value of 3 implies you at the air/bubble layer #1 interface
                grid(i,j) = 3
                c(i,j) = cb(1)
                rho(i,j) = rhob(1)
                percntabv = (diff + halfdelz)/delzinter
                a(i,j) = percntabv*a1 + (1.-percntabv)*abub(1)
              endif
```

```
                    endif
c   bubble layers
              do k=1,numlay
                if (k.eq.1) then
                   topdiff = zinter(i)-halfdelz
                else
                   if (bubplume.eq.2) then
                     topdiff = bbot(k-1)-halfdelz
                     botdiff = bbot(k)-halfdelz
                   elseif (bubplume.eq.1) then
                     topdiff = layprof(k-1,i)-halfdelz
                     botdiff = layprof(k,i)-halfdelz
                   endif
                endif
c     a grid value of k+9 implies you are in bubble layer #k
                if ((zpos(j).lt.topdiff).and.
     &                    (zpos(j).ge.botdiff)) then
                   if (diffb(k).gt.halfdelz) then
                     grid(i,j) = k+9
                     c(i,j) = cb(k)
                     rho(i,j) = rhob(k)
                     a(i,j) = abub(k)
                   else
c     a grid value of k+19 implies you at the bubble layer #k/bubble
c     layer #k+1 interface (ocean without bubbles for last layer)
                     grid(i,j) = k+19
                     percntabv = (diffb(k) + halfdelz)/delzinter
                     if (k.eq.numlay) then
                       c(i,j) = c2
                       rho(i,j) = rho2
                       a(i,j) = percntabv*abub(k) + (1.-percntabv)*a2
                     else
                       c(i,j) = cb(k+1)
                       rho(i,j) = rhob(k+1)
                       a(i,j) = percntabv*abub(k) + (1.-percntabv)*abub(k+1)
                     endif
                   endif
                endif
              end do
c   ocean without bubbles
c     a grid value of 2 implies you are in the ocean without bubbles
              if (bubplume.eq.2) then
                 botdiff = bbot(numlay)-halfdelz
              elseif (bubplume.eq.1) then
                 botdiff = layprof(numlay,i)-halfdelz
              endif
              if (zpos(j).lt.botdiff) then
```

```
                    grid(i,j) = 2
                    c(i,j) = c2
                    rho(i,j) = rho2
                    a(i,j) = a2
                    if (j.eq.rangeb) then
                       grid(i,j) = 5
                    endif
                 endif
c
              end do
           end do
c
           endif
c
c
c *** Choose Mach number and sonic boom parameters
c
c  Now the slope of the initial wave is defined according to the
c   speed of the aircraft, the Mach number.
c  This is where the Mach number is defined.
c Read in Mach number from input file
           read(unit=1, fmt=1030)mach
           write(unit=2, fmt=3000)' '
           write(unit=2, fmt=3010)'*********************************'
           write(unit=2, fmt=3000)' '
           write(unit=2, fmt=2040)'the aircraft is traveling at Mach
',mach
c
c Let the angle of incidence be defined as thetai
c
           thetai = asin(1./mach)
c
c
c The vertical and horizontal components of the initial wavefront
           vfactor = cos(thetai)
           hfactor = sin(thetai)
c
c Clearing main arrays
           do j=1,zmax
             do i=1,xmax
                p(i,j) = 0.
                ptmp1(i,j) = 0.
                ptmp2(i,j) = 0.
                intenx(i,j) = 0.
                intenz(i,j) = 0.
                psumx(i,j) = 0.
                psumz(i,j) = 0.
```

```
            end do
          end do
c
c Read in sonic boom parameters (peak pressure and duration)
          read(unit=1, fmt=1040)pmax
          write(unit=2, fmt=3000)' '
          write(unit=2, fmt=2060)'the peak pressure of the sonic
     &boom is ',pmax,' Pa'
          read(unit=1, fmt=1050)twotau
          write(unit=2, fmt=3000)' '
          write(unit=2, fmt=2070)'the duration of the sonic boom is
','
     &twotau,' seconds'
          tau = .5*twotau
c
c
c *** Calculate initial pressure for the entire domain
c
c Start the simulation:
c  The time is zero.  Set initial conditions.
          itime = 0
          time = deltat*itime
c
c Initialize p(i,j) to be initial data at time itime=0.
c  Here we define p to be the physical, i.e., real pressure.
c  Now the initial pressure is calculated
c
          do j=1,zmax
            do i=1,xmax
              const = hfactor*(xpos(i)) - vfactor*(zpos(j)-depth1-80.)
              p(i,j)= 1.5*pmax*(const/(c1*tau))*2.5*w(const,(c1*tau))
              const = hfactor*(xpos(i))
     x               - vfactor*(zpos(j)-depth1-80.
     x             -deltat*c(i,j)/vfactor)
              ptmp2(i,j)= 1.5*pmax*(const/(c1*tau))*2.5*w(const,(c1*tau))
            end do
          end do
c
c Initialize top boundary.
c  The bottom boundary will be an absorbing boundary condition.
          do i= 2,xmax-1
            p(i,zmax)= p(i,zmax-1)
          end do
c
c Initialize left and right boundaries.
          do j=1,zmax
            p(xmax,j)= p(xmax-1,j)
```

```
                         p(1,j)= p(2,j)
                       end do
c
c Initialize information for the bottom absorbing boundary condition.
                       do i= 2, xmax-1
                         botdist(i)=sqrt(xpos(i)*xpos(i)
     x                             +(zpos(1)-depthl)*(zpos(1)-depthl) )
                       end do
                       do i=2, xmax-1
                         cosbotangle= -(zpos(1)-depthl)/botdist(i)
                         sinbotangle= xpos(i)/botdist(i)
                         botangfac(i)=    - (cosbotangle
     x                             + sinbotangle*sinbotangle/cosbotangle )
                       end do
c
c Read in from input file:
c  whether or not to include a dashed outline of the refined
c  grid region (in the image); (1) yes, (2) no
                       read(unit=1, fmt=1060)outline
                       write(unit=2, fmt=3000)' '
                       write(unit=2, fmt=3010)'**********************************'
                       write(unit=2, fmt=3000)' '
                       write(unit=2, fmt=2006)'whether or not to include a
     &dashed outline of the refined grid region in the'
                       write(unit=2, fmt=2080)'output image, (1) yes, (2) no: ',outline
c
c Print out initial data.
                       call prraster(itime)
                       itime=itime+1
c
c Keeping initial pressure information for later use in intensity
c  calculations
                       do j = zmax,1,-1
                         do i = 1,xmax
                           if ((j.eq.1).or.(i.eq.1).or.(i.eq.xmax)) then
                             psumz(i,j) = 0.
                             psumx(i,j) = 0.
                           else
                             psumz(i,j) = psumz(i,j)+(p(i,j)-p(i,j-1))
                             avp1 = 0.5*(p(i-1,j) + p(i-1,j-1))
                             avp2 = 0.5*(p(i+1,j) + p(i+1,j-1))
                             psumx(i,j) = psumx(i,j)+(avp1-avp2)
                           endif
                         end do
                       end do
c
c
```

```
c *** Calculate pressure at later times
c
c Begin the loop and advance the solution a time step:
c
c  number of time steps
        nosteps = int(maxtime/deltat)
c
        do i=1,nosteps
c
c Do boundary precalculations.
        call prebound
c
c Do finite difference method.
        call sochdiff2d
c
c Do boundary conditions.
        call bounds
c
c Do intensity calculations.
        call intensity
c
c Print every 40 time steps:
        if(mod(itime,40).eq.0) then
c           print*, 'Printing at time ', time
            call prraster(itime/40)
        endif
c
c Do increment.
c For next time know:
        itime=itime+1
        time=itime*deltat
      end do
c
c We are done.
c
      write(unit=2, fmt=3000)' '
      write(unit=2, fmt=3010)'**********************************'
      write(unit=2, fmt=3000)' '
      write(unit=2, fmt=2005)'This program terminated normally.'
c
c formats for input and output files
c
1000        format(//T2,I1)
1010        format(T2,F4.2)
1020        format(/T2,I1)
1030        format(////T2,F3.1)
1040        format(T2,F4.1)
```

```
1050        format(T2,F5.3)
1060        format(T2,I1)
2000        format(A40)
2001        format(A29)
2002        format(A27)
2003        format(A76)
2004        format(A72)
2005        format(A33)
2006        format(A76)
2010        format(A12,I1)
2020        format(A9,F4.2,A7)
2030        format(A6,I1)
2040        format(A34,F3.1)
2050        format(A34,F5.1)
2060        format(A39,F4.1,A3)
2070        format(A34,F5.3,A8)
2080        format(A31,I1)
3000        format(A1)
3010        format(A35)
4000        format(////T2,I1)
4001        format(A36)
4002        format(A23,I1)
c4003        format(//T2,F4.1)
4004        format(T2,F6.1)
4005        format(A27,I2,A3,F4.1,A2)
4006        format(A19,F5.0,A4)
4007        format(A12,F5.0,A7)
4008        format(T2,F4.1)
4010        format(//T2,I2)
4011        format(A25,I2)
5000        format(//T2,I1)
5001        format(A43)
5002        format(A24,I1)
c
c closing input and output files
        close(unit=1)
        close(unit=2)
        close(unit=3)
        stop
        end
        subroutine prebound
c Subroutine to precalculate boundary conditions.
c  Written on 8/10/95 at 9:00 pm by V. Sparrow.
c  Right now we are only precalculating the bottom boundary condition.
        include 'metergrid.common'
c
c Local vars:
```

```
            integer i
c
c Precalculate the bottom absorbing boundary condition.
            do i=2, xmax-1
              bottomval(i) = ( (1. - c2*deltat/botdist(i))*p(i,1)
     x                 - (c2*deltat/deltaz)*botangfac(i)*(p(i,2)-p(i,1))
)
            end do
            return
            end
            subroutine bounds
c Subroutine to implement boundary conditions.
c  Modified on 8/10/95 at 9:00 pm by V. Sparrow to include the
c   absorbing boundary condition on the bottom.
            include 'metergrid.common'
c
c Local vars:
            integer i,j
c Top and Bottom:
            do i=2,xmax-1
              p(i,zmax)= p(i,zmax-1)
            end do
c
c Left and Right:
            do j=1,zmax
              p(xmax,j)= p(xmax-1,j)
              p(1,j)= p(2,j)
            end do
c
c Now recall our calculated bottom boundary condition values.
            do i=2, xmax-1
              p(i,1)= bottomval(i)
            end do
            return
            end
            subroutine prraster(imagenum)
c
c This subroutine takes our data and prints raster images useable
c  by the NCSA visualization routines.  It depends on the NCSA
c  Hiearchical Data Format (HDF) package.
c
            include 'metergrid.common'
c
c Declare subroutine argument.
c  imagenum - an integer, which image number we use.
            integer imagenum
c
```

```
c Declare external integer function.
        integer d8pimg
        integer dssdims,dspdata
c
c Local variables.
c imin, imax - the integer minimum and maximum values, respectively.
c  These numbers should be between 0 and 255, imin < imax.
c rmin, rmax - the real minimum and maximum values for the
c  acoustic pressure.
c charray - the character array used for our raster images.
c rarray - real temporary array
c rtmp - real vector in y2 direction
c chnum - a character string to represent the numbers 000 to 999.
c fname - the filename for our image
c ones, tens, huns - digits for ones, tens, and hundreds.
c istat - status of the HDF routines called, on return.
        real rmin, rmax, rarray(xmax, zmax), rtmp(zmax)
        real ixarray(xmax,zmax),izarray(xmax,zmax)
        real ixtmp(zmax),iztmp(zmax)
        integer imin, imax, ones, tens, huns, istat
        integer dims(2)
        character*1 charray(xmax,zmax)
        character*3 chnum
        character*8 fname,fnamereal,fnameintx,fnameintz
c
c        print*,'you have made it to the file saving loop'
c         print*,'at time step ',imagenum
c
c Set integer minimum and maximum.
        imin=2
        imax=254
c
c Set minimum and maximum for pressure.
        rmin = -120.0
        rmax = 120.0
c
c Find the filename for each image from imagenum.
c  This technique is taken from the PhD thesis of Dr. Mike White,
p. 94.
        ones = mod(imagenum,10)
        tens = ifix(mod(imagenum,100)/10.)
        huns = ifix(imagenum/100.)
c  Add 48 to make them printable, and squeeze together.
        chnum=char(huns+48)//char(tens+48)//char(ones+48)
c  Make filename.
        fname='pimno'//chnum
        fnamereal='rimno'//chnum
```

```
                fnameintx='intxx'//chnum
                fnameintz='intzz'//chnum
c
c Flip the array's top and bottom, so things are printed
c  in a Cartesian system, i.e. the top is up.
c (Fortran is strange, that you must do this!)
c
c Now do flipping.
            do i=1,xmax
               do j=1,zmax
c   change to this subroutine
                  rtmp(j)=p(i,j)
                  ixtmp(j)=intenx(i,j)
                  iztmp(j)=intenz(i,j)
               end do
               do j=1,zmax
                  rarray(i,j)=rtmp(zmax-j+1)
                  ixarray(i,j)=ixtmp(zmax-j+1)
                  izarray(i,j)=iztmp(zmax-j+1)
               end do
            end do
c
c Write image with realistic pressure values via a call
c  to an HDF routine
            if (((imagenum.ge.135).and.(imagenum.le.170).and.
     &                (mod(imagenum,5).eq.0)).or.
     &        ((imagenum.eq.100).or.(imagenum.eq.105).or.
     &        (imagenum.eq.110).or.(imagenum.eq.115).or.
     &        (imagenum.eq.120))) then
            dims(1) = xmax
            dims(2) = zmax
c
            istat = dssdims(2,dims)
            if (istat.ne.0) then
              print *,'Error writing HDF file ',fnamereal,'.'
              stop
            endif
c
            istat = dspdata(fnamereal,2,dims,rarray)
            if (istat.ne.0) then
              print *,'Error writing HDF file ',fnamereal,'.'
              stop
            endif
          endif
c
c Write image with instantaneous intensity values via a call
c  to an HDF routine
```

```fortran
      if ((imagenum.eq.160).or.(imagenum.eq.135).or.
     &     (imagenum.eq.100).or.(imagenum.eq.105).or.
     &     (imagenum.eq.110).or.(imagenum.eq.115).or.
     &     (imagenum.eq.120).or.
     &             (imagenum.eq.170)) then
        dims(1) = xmax
        dims(2) = zmax
c
        istat = dssdims(2,dims)
        if (istat.ne.0) then
          print *,'Error writing HDF file ',fnameintx,'.'
          print *,'Error writing HDF file ',fnameintz,'.'
          stop
        endif
c
        istat = dspdata(fnameintx,2,dims,ixarray)
        if (istat.ne.0) then
          print *,'Error writing HDF file ',fnameintx,'.'
          stop
        endif
        istat = dspdata(fnameintz,2,dims,izarray)
        if (istat.ne.0) then
          print *,'Error writing HDF file ',fnameintz,'.'
          stop
        endif
      endif
c
c Do scaling and printing for pressure.
c
c Now put a solid line along the two media interface.
      do i=1,xmax
        do j=1,zmax
          if ((grid(i,j).eq.3).or.(grid(i,j).ge.20)) then
            rarray(i,zmax-j+1)=-118.0
          endif
        end do
      end do
c
c Now put a dashed line around boundaries of refined grid region.
      if (outline.eq.1) then
      do i=1,xmax
        do j=1,zmax
          if ((grid(i,j).eq.4).or.(grid(i,j).eq.5)) then
            if ((mod(i,10).eq.1).or.(mod(i,10).eq.2).or.
     x          (mod(i,10).eq.3).or.(mod(i,10).eq.4).or.
     x          (mod(i,10).eq.5)) then
              rarray(i,zmax-j+1)=-118.0
```

```
                           endif
                         endif
                       end do
                     end do
                   endif
       c
       c Scale the real array rarray for putting in character form.
                   do i=1,xmax
                     do j=1,zmax
                       rarray(i,j)=(imax-imin)*( (rarray(i,j)-rmin)
            x                        /(rmax-rmin) ) + imin
                     end do
                   end do
       c
       c Make a character array based on the real array rarray.
                   do i=1,xmax
                     do j=1,zmax
                       charray(i,j) = char( ifix(rarray(i,j)) )
                     end do
                   end do
       c
       c Write image via a call to an HDF routine.
       c  0 = no compression.
                   if (((imagenum.ge.135).and.(imagenum.le.170).and.
            &                 (mod(imagenum,5).eq.0)).or.
            &            ((imagenum.eq.100).or.(imagenum.eq.105).or.
            &            (imagenum.eq.110).or.(imagenum.eq.115).or.
            &            (imagenum.eq.120).or.(imagenum.eq.000))) then
                     istat = d8pimg(fname,charray,xmax,zmax,0)
                     if (istat.ne.0) then
                       print*, 'Error writing HDF file ', fname, '.'
                       stop
                     endif
                   endif
       c Done.
                   return
                   end
                   subroutine sochdiff2d
                   include 'metergrid.common'
       c This routine implements two types of finite difference routines
       c  At the interface: a 2-D routine similar to Sochacki, et al.
       c  Rest of the domain: 2-D central difference routine
       c  - accounts for uniform grid in x direction and non-uniform grid
       in
       c    z direction
                   real b
                   real cnsqtsq,bb1,bb2
```

```
            integer j,k,i
c           print *,'you have entered the sochdiff2d subroutine'
c
            do k = zmax-1,2,-1
              do j = 2,xmax-1
c  if not at any interface
                if ((grid(j,k).eq.1).or.(grid(j,k).eq.4).or.
     &                (grid(j,k).eq.2).or.(grid(j,k).eq.5).or.
     &                ((grid(j,k).ge.10).and.(grid(j,k).lt.20))) then
                  if ((grid(j,k).eq.1).or.(grid(j,k).eq.4)) then
                    cnsqtsq = c1sqtsq
                  endif
                      if ((grid(j,k).eq.2).or.(grid(j,k).eq.5)) then
                    cnsqtsq = c2sqtsq
                  endif
                  do i=1,numlay
                        if (grid(j,k).eq.(i+9)) then
                      cnsqtsq = csqtsq(i)
                    endif
                  end do
                  ptmp1(j,k) = 2.*p(j,k)-ptmp2(j,k)
     &                    + cnsqtsq*((p(j+1,k)-2.*p(j,k)+p(j-1,k))*oodelxsq
     &              + ((p(j,k+1)-p(j,k))/delz(k+1)
     &              - (p(j,k)-p(j,k-1))/delz(k))*2./(delz(k+1)+delz(k)))
c                 if (ptmp1(j,k).gt.200.) then
c                   print*,'ptmp1(',j,',',k,')=',ptmp1(j,k)
c                 endif
c  the centered-difference approximation for a grid point
c  lying near the interface
                elseif ((grid(j,k).eq.3).or.(grid(j,k).ge.20)) then
                  if (grid(j,k).eq.3) then
                    bb1 = b1
                    if (numlay.eq.0) then
                      bb2 = b2
                    else
                      bb2 = bbub(1)
                    endif
                  endif
                  do i=1,numlay
                    if (grid(j,k).eq.(i+19)) then
                      bb1 = bbub(i)
                      if (i.eq.numlay) then
                        bb2 = b2
                      else
                        bb2 = bbub(i+1)
                      endif
                    endif
                  endif
```

```
               end do
c   (add more if statements here if adding bubble layers)
             ptmp1(j,k) = 2.*p(j,k)-ptmp2(j,k)
     &         +(deltsq/a(j,k))*(oodelxsq*(b(j,.5,k,0.,zpos(k),
     &    deltaz,delzinter,bb1,bb2,ranget,rangeb,zinter(j))*p(j+1,k)
     &    -(b(j,.5,k,0.,zpos(k),deltaz,delzinter,bb1,bb2,ranget,
     &    rangeb,zinter(j))+b(j,-.5,k,0.,zpos(k),deltaz,delzinter,
     &    bb1,bb2,ranget,rangeb,zinter(j)))*p(j,k)
     &    +b(j,-.5,k,0.,zpos(k),deltaz,delzinter,bb1,bb2,ranget,
     &    rangeb,zinter(j))*p(j-1,k))
     &    +(b(j,0.,k,.5,zpos(k),deltaz,delzinter,bb1,bb2,ranget,
     &    rangeb,zinter(j))*(p(j,k+1)-p(j,k))/delz(k+1)
     &    -b(j,0.,k,-.5,zpos(k),deltaz,delzinter,bb1,bb2,ranget,
     &    rangeb,zinter(j))*(p(j,k)-p(j,k-1))/delz(k))*2./(delz(k+1)
     &    +delz(k)))
c             if (ptmp1(j,k).gt.200.) then
c                print*,'ptmp1(',j,',',k,')=',ptmp1(j,k)
c             endif
           endif
         end do
       end do
c
c   advance the time
       do k = 1,zmax
         do j = 1,xmax
c          print *,'you are now advancing the time in sochdiff2d'
           ptmp2(j,k) = p(j,k)
           p(j,k) = ptmp1(j,k)
         end do
       end do
c
       return
       end
c
       subroutine intensity
       include 'metergrid.common'
c This subroutine calculates the instantaneous intensity over the
c  the computational domain using the pressure values
c
c  intenx: x component of intensity
c  intenz: z component of intensity
       integer i,j
       real zfactor,xfactor,pt1,pt2
       real pavt1,pavt2
c
       pt1 = 0.
       pt2 = 0.
```

```
                    pavt1 = 0.
                    pavt2 = 0.
                    do j = zmax,1,-1
                      do i = 1,xmax
                        if ((j.eq.1).or.(i.eq.1).or.(i.eq.xmax)) then
                          intenz(i,j) = 0.
                          intenx(i,j) = 0.
                        else
                          zfactor = deltat/(2.*rho(i,j)*delz(j))
                          xfactor = deltat/(2.*rho(i,j)*2.*deltax)
                          pt1 = p(i,j)
                          pt2 = p(i,j-1)
                          psumz(i,j) = psumz(i,j)+(pt1-pt2)
                          intenz(i,j) = zfactor*(pt1+pt2)*psumz(i,j)
                          pavt1 = 0.5*(p(i-1,j) + p(i-1,j-1))
                          pavt2 = 0.5*(p(i+1,j) + p(i+1,j-1))
                          psumx(i,j) = psumx(i,j)+(pavt1-pavt2)
                          intenx(i,j) = xfactor*(pavt1+pavt2)*psumx(i,j)
                        endif
                      end do
                    end do
c

                    return
                    end
c
c

                    subroutine plume
                    include 'metergrid.common'
c This function calculates the plume profile given some
c   laydepth          depth of the bottom of the bubble layer
c   layno         number of bubble layer (1 being top)
c   midplume          horizontal value for middle of bubble plume
c

                    real cosmult,adjprof,pi,xx
                    integer i
c   cosmult           multiplier for cosine function (adjusts the
c                     (wavelength appropriate to radius of plume)
c   adjprof           adjustment of the profile by a negative
c                     shift in the horizontal direction
c

                    pi = 4.0 * atan(1.0)
c
c Determine horizontal shift appropriate for the bubble layer
                    adjprof = layno*delzinter
                    cosmult = pi/(laydepth*2)
c
c Calculate the profile of the bubble plume across the horizontal
```

```
          xx = -laydepth
          do i=1,xmax
            if (i.le.(midplume-laydepth-1)) then
              pprofile(i) = 0.0 - adjprof
            elseif (i.ge.(midplume+laydepth+1)) then
              pprofile(i) = 0.0 - adjprof
            else
              pprofile(i) = -laydepth*(cos(cosmult*xx))**2
     &                      - adjprof
              xx = xx + 1
            end if
          end do
c
          return
          end
c
c
          real function b(m,mstep,n,nstep,zval,del,delinter,
     &                    bcon1,bcon2,rt,rb,zint)
c This function computes the b function which appears in the Sochacki
c  fd routine
          real mstep,nstep,zval,del,delinter,bcon1,bcon2,val,zint
          integer m,n,rt,rb
c  rt - top of dense range
c  rd - bottom of dense range
c
          if ((n.gt.rt).or.(n.lt.rb)) then
            val = zval + nstep*del
          elseif ((n.lt.rt).and.(n.gt.rb)) then
            val = zval + nstep*delinter
          elseif (n.eq.rt) then
            if (nstep.gt.0.) then
              val = zval + nstep*del
            else
              val = zval + nstep*delinter
            endif
          else
            if (nstep.gt.0.) then
              val = zval + nstep*delinter
            else
              val = zval + nstep*del
            endif
          endif
          if (val.gt.zint) then
            b = bcon1
          elseif (val.lt.zint) then
            b = bcon2
```

```
           else
             b = .5*(bcon1+bcon2)
           endif
           return
           end
c
           real function swellfunc(dist, amp, lambda, wtype)
c This function calculates the height of the ocean swell given some
c  dist - distance along the ocean
c  amp - amplitude of the ocean swell
c  lambda - approximate wavelength of the swell
c  wtype - sinusoidal (1) or trochoidal (2) waves or a combination
(3)
           real dist, amp, lambda
           integer wtype
c
c Wavenumber
           real kk,kk1,kk2
c
           real temp,amp1
c
c Calculate wavenumber.
           kk= 2.*3.141592654/lambda
           kk1= 2.*3.141592654/(0.5*lambda)
           kk2= 2.*3.141592654/(0.3*lambda)
c
c Which type?
           if(wtype.eq.1) then
             temp=amp*cos(kk*dist)
           elseif(wtype.eq.2) then
             temp= - amp*cos(kk*dist)
     x             + (1./2.)*kk*amp**2*cos(2.*kk*dist)
     x             - (3./8.)*kk**2*amp**3*cos(3.*kk*dist)
           elseif(wtype.eq.3) then
             temp= 0.6494*amp*cos(kk1*dist)
     x             - .5*amp*cos(kk*dist)
     x             + (1./2.)*kk*amp**2*cos(2.*kk*dist)
     x             - (3./8.)*kk**2*amp**3*cos(3.*kk*dist)
           elseif(wtype.eq.4) then
             amp1 = 1.2418*amp
             temp= 0.3*amp1*cos(kk2*dist)
     x             - .5*amp1*cos(kk*dist)
     x             + (1./2.)*kk*amp1**2*cos(2.*kk*dist)
     x             - (3./8.)*kk**2*amp1**3*cos(3.*kk*dist)
           else
             print*, 'swellfunc called with invalid argument'
           endif
```

```
            swellfunc=temp
            return
            end
c
c
            real function w(x,wl)
c This is the weighting function for the initial pressure
c  x - the position in the grid
c  wl - half wavelength of initial pressure
            real x,wl,pi
            pi = 4.0 * atan(1.0)
            if ((x.gt.wl).or.(x.lt.-wl)) then
              w = 0.0
            else
              w = .5 + .5*cos(pi*x/wl)
            endif
            return
            end
```

File for global variables (metergrid.common) - this file is read in by prop2o.f:

```
c  This is 'metergrid.common'.
c  Last update: J Rochat 3/12/97
c
c
c Declare global variables.
        integer xmax, zmax
        parameter(xmax=800, zmax=800)
        common /cons/ rho1,rho2,c1,c2,percntabv
        real rho1,rho2,c1,c2,percntabv
        common /cons2/ rhob(10),cb(10),csqtsq(10),abub(10),bbub(10)
        real rhob,cb,csqtsq,abub,bbub
        common /gridvars/ c(xmax,zmax),zinter(xmax),
     x    a(xmax,zmax),bottomval(xmax),rho(xmax,zmax)
        real c,zinter,a,bottomval,rho
        common /morecons/ deltat,deltax,deltaz,dtodx,oneortwo,
     x    newzmax
        real deltat,deltax,deltaz,dtodx
        integer oneortwo,newzmax
        common /evenmorecons/ a1,a2,b1,b2,gamma,deltsq,oodelxsq,
     x    oodelzsq,asumo2,bsumo2,c1sqtsq,c2sqtsq,delzinter
        real a1,a2,b1,b2,gamma,deltsq,oodelxsq,
     x    oodelzsq,asumo2,bsumo2,c1sqtsq,c2sqtsq,delzinter
        common /evenmoreplus/ grid(xmax,zmax),m2count,ranget,
     x    rangeb,outline,numlay,bubbles
        integer grid, m2count,ranget,rangeb,outline,numlay,bubbles
        common /geomvars/ depthl,botangfac,botdist
        real depthl,botangfac(xmax),botdist(xmax)
        common /moregridvars/ xpos(xmax),zpos(zmax),
     &     dtodxsq,p(xmax,zmax),ptmp1(xmax,zmax),ptmp2(xmax,zmax),
     &     delz(zmax),intenz(xmax,zmax),intenx(xmax,zmax),
     &     psumx(xmax,zmax),psumz(xmax,zmax)
        real xpos,zpos,dtodxsq,u,utmp1,utmp2,delz,intenz,intenx,
     &     psumx,psumz
        common /plumevars/ midplume,laydepth,layno,bubplume,
     &     pprofile(xmax),layprof(6,xmax)
        integer midplume,laydepth,layno,bubplume
        real pprofile,layprof
```

Example of input file (`inputfile`) - this file contains the values for the physical parameters in the simulation and is read in by `prop2o.f`:

```
This file contains all the information needed as input for
the sonic boom/ocean interaction simulation program; currently
this program is named prop2o.f.
2                          indication of whether or not to include
bubble
                           layers in the simulation, (1) yes, (2) no
                           (integer)
2                          indication of whether or not to include
a bubble
                           plume in the simulation, (1) yes, (2) no
                           (integer)
2                            ocean type, (1) flat, (2) wavy (integer)
3.75                           height of swell waves (height is defined
as the distance from crest to trough) (meters,real)
2                            type of wavy ocean profile (integer)
(1) sinusoidal
(2) trochoidal
(3) combination #1: trochoidal+1/2wavelength
sine
(4) combination #2: trochoidal+1/3wavelength
sine
2.4                            Mach number (real)
50.0                            peak pressure of sonic boom (Pa,real)
0.300                            duration of sonic boom (seconds,real)
1                          indication of whether or not to include
a dashed
                           outline of the refined grid region, (1) yes,
(2) no
                           (integer)
If a parameter does not apply to the particular problem, you must
type
0 for the value, careful to match number of digits.
```

Example of another input file (bublayers) - this file contains the physical parameters for the bubble layers and is read in by prop2o.f:

```
This file contains values for parameters in the bubble layers
06                    number of bubble layers
01.0                   (thickness, m) number of meters in depth for
bubble layer #1 (top) (real)
0841.0                  speed of sound (m/s) in bubble layer #1 (real)
1000.0                  density (kg/m^2) of bubble layer #1 (real)
01.0                   (thickness, m) number of meters in depth for
bubble layer #2 (top) (real)
1214.0                  speed of sound (m/s) in bubble layer #2 (real)
1000.0                  density (kg/m^2) of bubble layer #2 (real)
01.0                   (thickness, m) number of meters in depth for
bubble layer #3 (top) (real)
1375.0                  speed of sound (m/s) in bubble layer #3 (real)
1000.0                  density (kg/m^2) of bubble layer #3 (real)
01.0                   (thickness, m) number of meters in depth for
bubble layer #4 (top) (real)
1446.0                  speed of sound (m/s) in bubble layer #4 (real)
1000.0                  density (kg/m^2) of bubble layer #4 (real)
02.0                   (thickness, m) number of meters in depth for
bubble layer #5 (top) (real)
1483.0                  speed of sound (m/s) in bubble layer #5 (real)
1000.0                  density (kg/m^2) of bubble layer #5 (real)
03.0                   (thickness, m) number of meters in depth for
bubble layer #6 (top) (real)
1497.0                  speed of sound (m/s) in bubble layer #6 (real)
1000.0                  density (kg/m^2) of bubble layer #6 (real)
```

Example of output file (outputfile) - this file is created by prop2o.f along with the HDF-formatted pressure value and intensity value files:

```
***********************************
1 for including bubbles or 2 for not
include bubble layers: 1
1 for including a bubble plume or 2 for not
include a bubble plume: 1
number of bubble layers:  6
thickness of bubble layer # 1 :  1.0 m
  speed of sound =  841. m/s
  density = 1000. kg/m^3
thickness of bubble layer # 2 :  1.0 m
  speed of sound = 1214. m/s
  density = 1000. kg/m^3
thickness of bubble layer # 3 :  1.0 m
  speed of sound = 1375. m/s
  density = 1000. kg/m^3
thickness of bubble layer # 4 :  1.0 m
  speed of sound = 1446. m/s
  density = 1000. kg/m^3
thickness of bubble layer # 5 :  2.0 m
  speed of sound = 1483. m/s
  density = 1000. kg/m^3
thickness of bubble layer # 6 :  3.0 m
  speed of sound = 1497. m/s
  density = 1000. kg/m^3
***********************************
x total distance is approximately 800.0
z total distance is approximately 750.5
***********************************
1 for a flat ocean surface or 2 for wavy
ocean type: 1
the height of the swell waves
height = 0.00 meters
type of ocean wave profile:
(1) sinusoidal, (2) trochoidal, (3) combination #1: sine wave .5 wavelength
of trochoidal, (4) combination #2: sine wave .3 wavelength of trochoidal
type: 0
***********************************
the aircraft is traveling at Mach 2.4
the peak pressure of the sonic boom is 50.0 Pa
the duration of the sonic boom is 0.300 seconds
***********************************
```

whether or not to include a dashed outline of the refined grid region in the
output image, (1) yes, (2) no: 1
****
This program terminated normally.

Appendix C.
Statistically Generated Ocean Surface Profile

Statistical methods are often used to formulate the profile of waves in a fully developed sea.[3] Oceanographers attempt to study this type of sea surface by representing the irregular pattern with a series of small-amplitude sine waves of varying size and phase.[22] Information from several references are brought together here to construct a formulation for the surface profile as a function of the horizontal distance across the computational domain. Although this formulation is not used in any of the final calculations, it is shown here for its potential use as a tool for others whose computational constraints allow it.

Several one-dimensional spectrum models can be used to construct the ocean surface profile for a fully developed sea. The one chosen to use here is the Bretschneider Spectrum which is found in Ref. 57. In one direction on the ocean surface, the energy density S as a function of ocean wave period T is written

$$S(T) = \frac{3.44T^3 (\bar{H})^2}{(\bar{T})^4} e^{-0.675(T/\bar{T})^4} . \tag{C.1}$$

where $\bar{H}$ is the average wave height and $\bar{T}$ is the average period. This function follows a Gaussian distribution. In order to calculate the energy density as a function of frequency, $S(f)$, the relations $S(f) = S(T)T^2$, $f = 1/T$, and $\bar{f} = 1/\bar{T}$ are applied to get

$$S(f) = \frac{3.44f^{-5} (\bar{H})^2}{(\bar{f})^{-4}} e^{-0.675(\bar{f}/f)^4} . \tag{C.2}$$

where $\bar{f}$ is the average frequency. Now the energy density function follows a Rayleigh distribution. Eq. (C.2) permits the calculation of the energy density amplitude for a particular frequency, this amplitude is proportional to the wave amplitude squared,[16]

$$S(f) = \frac{A^2}{2\Delta f} . \tag{C.3}$$

where A is the ocean wave amplitude and Δf is the frequency spacing. Solving for A, the wave amplitude as a function of frequency can be written

$$A(f) = \left[2\Delta f \frac{3.44 (\bar{H})^2 (\bar{f})^4}{f^5} e^{-0.675(\bar{f}/f)^4} \right]^{\frac{1}{2}} . \tag{C.4}$$

The surface profile can now be calculated as the sum of sinusoids. The equation used to formulate the surface profile uses a finite number of frequencies $nofreqs$ for the sum of the sine waves with their statistically corresponding amplitude, Eq. (C.4), but no phase variation. This equation reads

$$\eta(x) = \sum_{n=1}^{nofreqs} A_n(f_n) \cos\left[\frac{2\pi f_n}{c_{ow}(f_n)} x \right] , \tag{C.5}$$

where η is the wave amplitude or distance from the mean water line. It should be noted that the speed of the ocean wave c_{ow} is frequency dependent. The deep water approximation for the phase speed of the ocean surface wave is[31.57]

$$c_{ow}\left(f_n\right) = \frac{g}{2\pi f_n} \ . \qquad (C.6)$$

where g is the gravitational constant. Eq. $(C.5)$ now becomes

$$\eta\left(x\right) = \sum_{n=1}^{nofreqs} A_n\left(f_n\right)\cos\left[\frac{\left(2\pi f_n\right)^2}{g}x\right] \ . \qquad (C.7)$$

Plugging the values $nofreqs = 10$, $\Delta f = 10$ Hz, $g = 9.8$ m/s^2, $\bar{H} = 1$, and $\bar{f} = 50$ Hz (picked to give maximum wave height of approximately 3.75 m) into Eq. $(C.7)$, the ocean surface profile is calculated to look like Fig. C.1. It is seen in the figure that the profile is not very smooth nor does it have a distinct periodicity, qualities expected from this statistically generated function, but qualities which are unmanageable in computational work with particular constraints.

Figure C.1: Complex statistically generated ocean surface profile using 10 frequencies.

Filters applied to the more simple ocean surface profiles to erase instabilities (explained in Chapter 7) do not work on this complex profile. The simple profile filters exploit the surface's periodicity by killing the periodic offenders, the pixel values without any side neighbors. Also, the grid surrounding the air-water interface is refined enough to accommodate the smoothness of the simple surface. On the other hand, the nonperiodicity of the statistically generated surface makes it difficult to formulate a filter which kills all the instabilities since the offending pixels are hard to recognize. Also the calculations would become very computationally intensive by trying to refine the grid enough to accommodate a relatively rough surface.

Again, it should be stated that the formulation given in this appendix is not actually used in any of the final complex ocean surface calculations. It is shown here as a possible useful tool and for the awareness of the complications of its use. In order to successfully implement a statistically generated ocean surface profile, it would be necessary to either formulate a general filter which would adjust parts of the surface to remove any instabilities, use a very refined grid around the air-water interface to smooth out the grid-block approximation of the surface, or possibly both.

Appendix D.
Two-Dimensional Fourth-Order Computer Code

This appendix contains a subroutine in the Fortran code for the two-dimensional simulation of a sonic boom interacting with the ocean. The main code used for the 2nd-order accurate simulation, prop2o.f. is used for the 4th-order code but now with the subroutine sochdiff2d replaced with a 4th-order accurate version.

Subroutine for two-dimensional wave equation. sochdiff2d:

```
      subroutine sochdiff2d
      include 'metergrid.common'
c This routine implements two types of finite difference routines
c  At the interface: a 2-D routine similar to Sochacki, et al.
c  Rest of the domain: 2-D central difference routine
c  - accounts for uniform grid in x direction and non-uniform grid
in
c    z direction
      real b
      real cnsqtsq,bb1,bb2
      real pxx,pzz,t1const,t2const
      integer j,k,i
c      print *,'you have entered the sochdiff2d subroutine'
c
      do k = zmax-1,2,-1
        do j = 2,xmax-1
c if not at any interface
          if ((grid(j,k).eq.1).or.(grid(j,k).eq.4).or.
     &              (grid(j,k).eq.2).or.(grid(j,k).eq.5).or.
     &              ((grid(j,k).ge.10).and.(grid(j,k).lt.20))) then
            if ((grid(j,k).eq.1).or.(grid(j,k).eq.4)) then
              cnsqtsq = c1sqtsq
            endif
              if ((grid(j,k).eq.2).or.(grid(j,k).eq.5)) then
              cnsqtsq = c2sqtsq
            endif
            do i=1,numlay
                if (grid(j,k).eq.(i+9)) then
```

```fortran
                          cnsqtsq = csqtsq(i)
                      endif
                   end do
c 2nd-order for borders of grid
                   if ((k.gt.(zmax-2)).or.(k.lt.3)
     &                   .or.(j.gt.(xmax-2)).or.(j.lt.3)) then
                          pxx = (p(j+1,k)-2.*p(j,k)+p(j-1,k))*oodelxsq
                      pzz = ((p(j,k+1)-p(j,k))/delz(k+1)
     &                 - (p(j,k)-p(j,k-1))/delz(k))*2./(delz(k+1)+delz(k))
c 4th-order for interior of grid (not including interface)
                   else
                      pxx = (oodelxsq/12.)*(-p(j+2,k)+16.*p(j+1,k)
     &                           -30.*p(j,k)+16.*p(j-1,k)-p(j-2,k))
                      t1const = 2./(3.*(delz(k+2)+delz(k+1)+delz(k)
     &                               +delz(k-1)))
                      t2const = 8./(3.*(delz(k+1)+delz(k)))
               pzz = t1const*((-p(j,k+2)+p(j,k))/(delz(k+2)+delz(k+1))
     &                           + (-p(j,k-2)+p(j,k))/(delz(k)+delz(k-1)))
     &                   + t2const*((p(j,k+1)-p(j,k))/delz(k+1)
     &                       + (p(j,k-1)-p(j,k))/delz(k))
                   endif
                   ptmp1(j,k) = 2.*p(j,k)-ptmp2(j,k)
     &                   + cnsqtsq*(pxx + pzz)
c                   if (ptmp1(j,k).gt.200.) then
c                      print*,'ptmp1(',j,',',k,')=',ptmp1(j,k)
c                   endif
c   the centered-difference approximation for a grid point
c   lying near the interface
                   elseif ((grid(j,k).eq.3).or.(grid(j,k).ge.20)) then
                      if (grid(j,k).eq.3) then
                        bb1 = b1
                        if (numlay.eq.0) then
                          bb2 = b2
                        else
                          bb2 = bbub(1)
                        endif
                      endif
                      do i=1,numlay
                        if (grid(j,k).eq.(i+19)) then
                          bb1 = bbub(i)
                          if (i.eq.numlay) then
                            bb2 = b2
                          else
                            bb2 = bbub(i+1)
                          endif
                        endif
                      end do
```

```fortran
c    (add more if statements here if adding bubble layers)
             ptmp1(j,k) = 2.*p(j,k)-ptmp2(j,k)
     &          +(deltsq/a(j,k))*(oodelxsq*(b(j,.5,k,0.,zpos(k),
     &       deltaz,delzinter,bb1,bb2,ranget,rangeb,zinter(j))*p(j+1,k)
     &       -(b(j,.5,k,0.,zpos(k),deltaz,delzinter,bb1,bb2,ranget,
     &       rangeb,zinter(j))+b(j,-.5,k,0.,zpos(k),deltaz,delzinter,
     &       bb1,bb2,ranget,rangeb,zinter(j)))*p(j,k)
     &       +b(j,-.5,k,0.,zpos(k),deltaz,delzinter,bb1,bb2,ranget,
     &       rangeb,zinter(j))*p(j-1,k))
     &       +(b(j,0.,k,.5,zpos(k),deltaz,delzinter,bb1,bb2,ranget,
     &       rangeb,zinter(j))*(p(j,k+1)-p(j,k))/delz(k+1)
     &       -b(j,0.,k,-.5,zpos(k),deltaz,delzinter,bb1,bb2,ranget,
     &       rangeb,zinter(j))*(p(j,k)-p(j,k-1))/delz(k))*2./(delz(k+1)
     &       +delz(k)))
c             if (ptmp1(j,k).gt.200.) then
c                print*,'ptmp1(',j,',',k,')=',ptmp1(j,k)
c             endif
           endif
         end do
       end do
c
c  advance the time
       do k = 1,zmax
         do j = 1,xmax
c          print *,'you are now advancing the time in sochdiff2d'
           ptmp2(j,k) = p(j,k)
           p(j,k) = ptmp1(j,k)
         end do
       end do
c
       return
       end
```

Appendix E.
Window Applied to Initial Waveform to Obtain More N-Shaped Sonic Boom

This appendix contains a function applied in the Fortran code (prop2o.f. found in Appendix B) for the two-dimensional simulation of a sonic boom interacting with the ocean. This function w2 is used instead of the w function found in prop2o.f.

Function for window to help generate more N-shaped initial waveform, w2:

```
      real function w2(x,wl)
c This is the weighting function for the initial pressure
c Homemade window
c  x - the position in the grid
c  wl - half wavelength of initial pressure
      real x,wl,pi
      pi = 4.0 * atan(1.0)
      if ((x.gt.wl).or.(x.lt.-wl)) then
        w2 = 0.0
      elseif (((x.le.wl).and.(x.gt.(.75*wl))).or.
     x        ((x.ge.-wl).and.(x.lt.(-.75*wl)))) then
        w2 = 3.9*cos(1.5*x/wl)*cos(1.5*x/wl)
      elseif (((x.le.(.75*wl)).and.(x.gt.(.6*wl))).or.
     x        ((x.ge.(-.75*wl)).and.(x.lt.(-.6*wl)))) then
        w2 = .152 + 1.37*cos(1.5*x/wl)
      else
        w2 = 1.0
      endif
      return
      end
```

VITA

Judith Lynn Rochat was born September 7. 1968 and raised in southern California. Throughout elementary and junior high school she was active in gymnastics then soccer and was steadily learning the art of violin playing. Unfortunately, high school restrictions demanded a choice: sports or music. Music persisted. She attended Rolling Hills High School where she enjoyed math and the sciences and heavily involved herself in the school's award-winning orchestra and marching band (playing percussion): she graduated in 1986 as a member of the National Honor Society.

Judith received her undergraduate education at the University of California, San Diego. After shuffling through some engineering/physics majors. she ultimately decided on mathematics. In addition to the many math courses. she was able to include music theory and acoustics in her coursework. She also played in the La Jolla University/Civic Symphony. In September 1990 she completed a Bachelor of Arts degree in Applied Mathematics and a minor in Music Science and Technology.

After graduating with a math degree she was able to make the following formulation: [poor economy] + [fresh graduate] = no job $\Rightarrow$ ski. That first winter was spent working and skiing in Vail. Colorado. Other random jobs followed. Finally she decided that graduate school was the only way to excel.

Judith started in the Graduate Program in Acoustics at The Pennsylvania State University in August 1992. In addition to her coursework and research. she played in the Nittany Valley Symphony. She made presentations at two Acoustical Society of America (ASA) conferences and wrote a paper on her masters work. In December, 1994 she earned a Masters degree in Acoustics (research area: structural acoustics).

Judith decided to further her education and stay in the acoustics program for a doctoral degree. Over the years she presented papers at the Inter-Noise conference. the American Institute of Aeronautics and Astronautics (AIAA) conference, and the ASA conference and published her research in the Inter-Noise proceedings and AIAA journal. In 1995. 1996. and 1997 she received departmental awards for outstanding effort in publications. The summer of 1996 was spent working part time at NASA Langley Research Center. Upon completion. Judith will earn her Ph.D. in Acoustics (research area: computational acoustics. specifically wave propagation).

IMAGE EVALUATION
TEST TARGET (QA-3)

150mm

6"
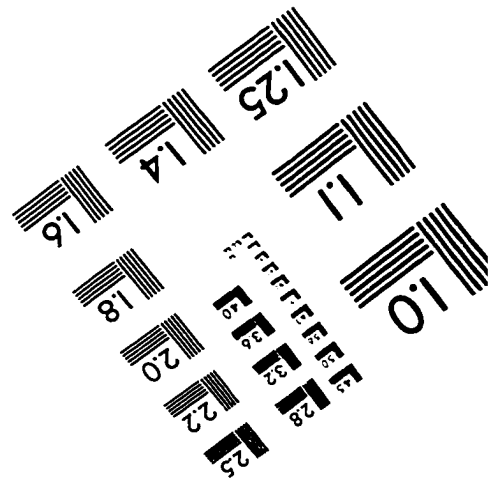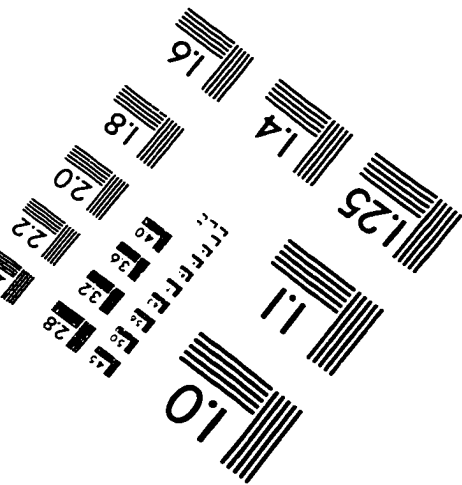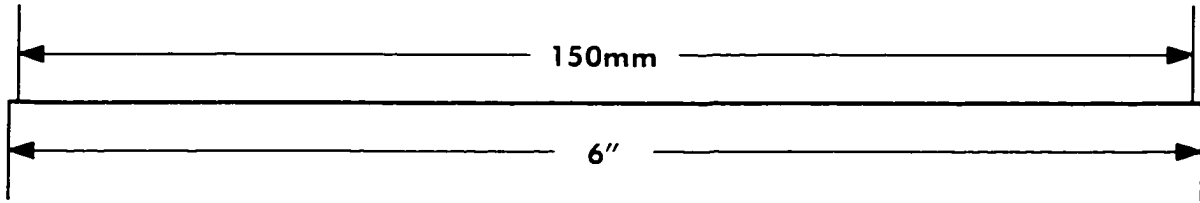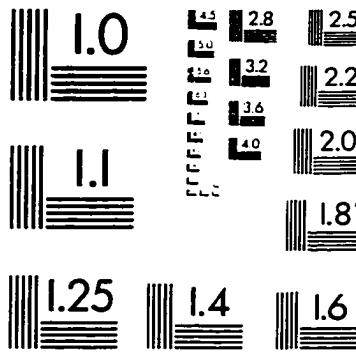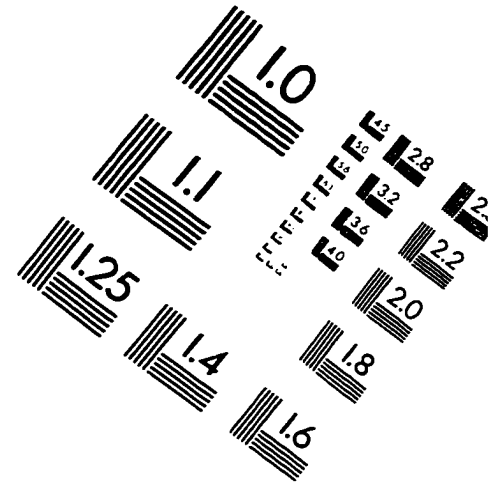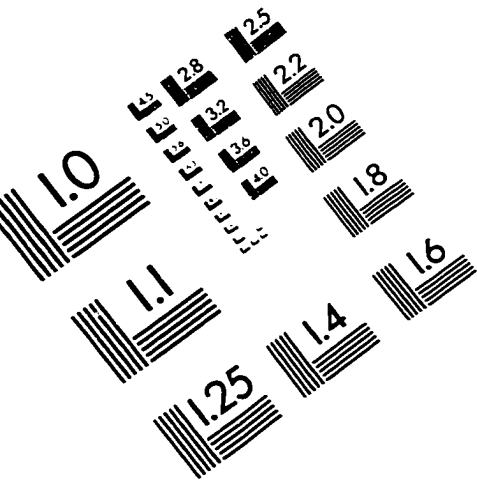
APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved