

AD-A221 510

The National Center for Physical Acoustics

DTIC
ELECTE
MAY 11 1990
S B D

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited



*The
University of Mississippi*

90 05 11 088

2

RADIAL INSTABILITIES OF A
PULSATING AIR BUBBLE IN WATER

By

STEVEN D. HORSBURGH

National Center for Physical Acoustics
University, MS 38677

January 1990

Technical Report
prepared for:

Office of Naval Research
Physics Division
Contract # N0014-87-K-0019

DTIC
ELECTE
MAY 11 1990
S B D

NCPA.LC.01.90

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution is unlimited

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NCPA.LC.01.90			7a. NAME OF MONITORING ORGANIZATION Office of Naval Research		
6a. NAME OF PERFORMING ORGANIZATION National Center for Physical Acoustics		6b. OFFICE SYMBOL (If applicable) LC	7b. ADDRESS (City, State, and ZIP Code) Department of the Navy Arlington, VA 22217		
6c. ADDRESS (City, State, and ZIP Code) NCPA Coliseum Drive University, MS 38677		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N 0014-87-K-0019			
8a. NAME OF FUNDING / SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)	10. SOURCE OF FUNDING NUMBERS		
8c. ADDRESS (City, State, and ZIP Code)			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
					WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) Radial Instabilities of a Pulsating Air Bubble in Water					
12. PERSONAL AUTHOR(S) Horsburgh, Steven D.; Crum, Lawrence A.					
13a. TYPE OF REPORT Technical		13b. TIME COVERED FROM 10-1-86 TO 9-30-89		14. DATE OF REPORT (Year, Month, Day) 90 January 30	
				15. PAGE COUNT 276	
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Physical Acoustics Nonlinear Dynamics		
			Mie-Scattering Shape Oscillations		
			Radial Instability Lasers		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Recently, several theoretical studies have been originated concerning the spherical stability of a pulsating air bubble in water. In this study, a light scattering technique was utilized to "photograph" in real-time the motion of the bubble surface. The main objective was to experimentally determine the shape oscillation threshold of an acoustically driven air bubble in water. These thresholds were defined as the minimum pressure necessary to drive a bubble, at a particular radius, into non-radial motion. The results of this study show a marked improvement over previous experiments in detecting the radial instabilities resulting in shape oscillation. Further, by using a numerical integration technique applied to a simplified theory, the experimental shape oscillation thresholds were predicted quite well.					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL Lawrence A. Crum			22b. TELEPHONE (Include Area Code) (601) 232-5815		22c. OFFICE SYMBOL LC

ACKNOWLEDGEMENTS

I would like to acknowledge the following individuals for their contributions to myself and to this dissertation.

Dr. Lawrence A. Crum, for serving as my research advisor and for giving me a chance to pursue this dissertation.

Dr. R. Glynn Holt, for many helpful discussions and for paving the way for this study.

Kimberly Frame, for laboratory assistance and personal support.

D. Felipe Gaitan, for technical assistance and for enlightening discussions.

Milena Gaitan, Nancy Roy, and Sandra Smith, for help in preparing this document.

Dr. Ron Roy, for constantly challenging me to re-evaluate assumptions.

Dr. Charles C. Church, for theoretical help in the area of rectified diffusion.

I would also like to acknowledge all the other individuals that contributed to this study and make it a success.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

ABSTRACT

RADIAL INSTABILITIES OF A PULSATING AIR BUBBLE IN WATER

HORSBURGH, STEVEN DEWITT. B. S. , University of Mississippi, 1983.
M.S. , University of Mississippi, 1984. Ph. D. , University of
Mississippi 1990. Dissertation directed by Dr. Lawrence C. Crum.

Recently, several theoretical studies have been originated concerning the spherical stability of a pulsating air bubble in water. In this study, a light scattering technique was utilized to "photograph" in real-time the motion of the bubble surface. The main objective was to experimentally determine the shape oscillation threshold of an acoustically driven air bubble in water. These thresholds were defined as the minimum pressure necessary to drive a bubble, at a particular radius, into non-radial motion. The results of this study show a marked improvement over previous experiments in detecting the radial instabilities resulting in shape oscillation. Further, by using a numerical integration technique applied to a simplified theory, the experimental shape oscillation thresholds were predicted quite well.

Table of Contents

LIST OF FIGURES	viii
-----------------------	------

Chapter

I. INTRODUCTION	1
II. FORMULATION OF THE PROBLEM	6
III. EQUATIONS OF MOTION	12
A. Introduction	12
B. Radial terms	15
C. Potential correction terms - deviation from spherical symmetry	19
D. Potential correction terms due to viscosity	22
E. Equations of motion	24
F. Recapitulation	25
IV. ANALYTICAL SOLUTION	27
A. Introduction	27

B. Internal bubble pressure	28
C. Third order approximate solution	29
1. Radial equation	29
2. Perturbation equation	30
3. Solutions to Hill's equation	33
D. Threshold criteria	37
E. Solutions to the analytic threshold equations.....	39
 V. NUMERICAL SOLUTION	 49
A. Introduction	49
B. Numerical integration	50
C. Solutions	52
D. Shape Oscillation Equation solver	55
E. Thresholds	59
F. Comparison with analytical solutions	62
 VI. MIE SCATTERING	 65
A. Theory	65
B. Calculation	67
1. Scattering intensity as a function of angle	68
2. Scattering at a specific angle for various bubble radii	 75

VII.	EXPERIMENT	81
A.	Method	81
B.	Apparatus	85
1.	Levitation cell	85
2.	Optical scattering system	88
3.	Support equipment	92
C.	Tuning the levitation cell	93
D.	Bubble radius calibration	96
E.	Pressure calibration	105
F.	Laser radiation pressure effects	121
VIII.	SHAPE THRESHOLDS	126
A.	Introduction	126
B.	Apparatus	127
C.	First experimental method	129
D.	Second experimental method	141
E.	Experimental observations	143
1.	Type 1-5 behaviors	143
2.	Pressure hysteresis	149
3.	Super-stability	150
4.	Instability width	150
5.	Beaconing effect	151
F.	Results	152

IX. SUMMARY	155
-------------------	-----

Appendices

A. Rectified diffusion rates	157
B. Mie scattering theory	162
C. Additional figures	172
D. Computer programs	205

LIST OF REFERENCES	272
--------------------------	-----

BIOGRAPHICAL SKETCH OF THE AUTHOR	276
---	-----

Figures

Figure no.	Description
1	Roots of the analytical solution, exhibiting a single threshold point.
2	Roots of the analytical solution, exhibiting multiple threshold points.
3	Analytical thresholds ignoring "bubble" feature for $n=2$ and $j=1$.
4	Analytical thresholds for the lowest pressure threshold for $n=2$ and $j=1$.
5	Combination of figures 3 and 4.
6	Analytical thresholds for $n=2$; $j=1,2,3$.
7	Analytical thresholds for $n=3$; $j=1,2,3$.
8	Analytical thresholds for $n=4$; $j=1,2,3$.

- 9 Combination of figures 6, 7, and 8 keeping only the lowest thresholds.
- 10 Comparison of analytical thresholds between polytropic exponents of $\kappa = 1.0$ (isothermal), 1.4 (adiabatic), and calculated from equation (3.3).
- 11 SOE numerical thresholds for $n=2$.
- 12 SOE numerical thresholds for $n=3$.
- 13 SOE numerical thresholds for $n=4$.
- 14 Minimum thresholds of figures 11, 12, and 13.
- 15 Overlay of figures 9 and 14. Comparison between the analytic and numerical solutions to the equations of motion.
- 16 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of scattering angle for a 50μ bubble.
- 17 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a solid angle of 4 degrees as a function of scattering angle for a 50μ bubble.

- 18 Composite I_1 angular Mie scattering result (over a 4 degree solid angle) for bubble radii: 10μ , 30μ , 50μ , 70μ , and 90μ .
- 19 Composite I_2 angular Mie scattering result (over a 4 degree solid angle) for bubble radii: 10μ , 30μ , 50μ , 70μ , and 90μ .
- 20 Experimental angular Mie scattering results (asterisks) for a 50μ bubble, overlayed by I_1 (solid line) and I_2 (dotted line).
- 21 Experimental angular Mie scattering results (asterisks) for a 60μ bubble, overlayed by I_1 (solid line) and I_2 (dotted line).
- 22 Experimental angular Mie scattering results (asterisks) for a 80μ bubble, overlayed by I_1 (solid line) and I_2 (dotted line).
- 23 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of bubble radius for an 80 degree scattering angle.
- 24 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a 4 degree solid angle, as a

- function of bubble radius for an 80 degree scattering angle.
- 25 Experimental data from DC detector at 80 degrees as a function of bubble radius.
- 26 Experimental data from DC detector at 80 degrees as a function of bubble radius. Overlaid by the Mie scattering prediction, I_1 (solid line) and I_2 (dotted line).
- 27 Figurative acoustic levitation technique.
- 28 Functional diagram of complete experimental system.
- 29 Detail of acoustic levitation cell.
- 30 Detail of optical scattering system.
- 31 Photodiode detector circuit.
- 32 Detail of sound cutoff apparatus.
- 33 Functional diagram of rise-time measurements experimental apparatus.
- 34 Detail of graduated, viewing microscope.
- 35 Bubble rise-times vs. event with error bars.
- 36 Rise-time vs. DC scattered voltage.

- 37 Bubble radius vs. DC scattered voltage.
- 38 Overlay of Fig. 37 and normalized Mie scattering prediction.
- 39 Schematic of pressure calibration technique using bubble position in the levitation cell.
- 40 Results of bubble size vs. Z position.
- 41 Pressure amplitude (given in hydrophone voltage units) in the cell as a function of height at 0.1 bar.
- 42 Slice of the pressure in the x direction along the $y=z$ =center axis.
- 43 Slice of the pressure in the y direction along the $x=z$ =center axis.
- 44 Slice of the pressure in the z direction along the $x=y$ =center axis.
- 45 Results from the SLICE program.
- 46 Results from the SLICE program.
- 47 Figure of large tank calibration apparatus.
- 48 Needle and B&K probe voltage vs. generator voltage.

49	Pressure in the cell vs. applied generator voltage.
50	Experimental data set #1 (background)
51	Experimental data set #2
52	Experimental data set #3
53	Experimental data set #4
54	Experimental data set #5
55	Experimental data set #6
56	Type 1 trace.
57	Type 2 trace.
58	Type 3 trace.
59	Type 4 trace.
60	Type 5 trace.
61	Experimental thresholds for radial instabilities.
62	Experimental thresholds vs. analytic prediction.
63	Experimental thresholds vs. numerical prediction.

A1	Rectified diffusion case #1
A2	Rectified diffusion case #2
A3	Rectified diffusion case #3
A4	Rectified diffusion case #4
C1-18	SOE example figures.
C19	Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of scattering angle for 10μ , 20μ , 30μ , and 40μ bubble radii.
C20	Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of scattering angle for 50μ , 60μ , 70μ , and 80μ bubble radii.
C21	Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a solid angle of 4 degrees as a function of scattering angle for 10μ , 20μ , 30μ , and 40μ bubble radii.
C22	Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a solid angle of 4 degrees as a

function of scattering angle for 50μ , 60μ , 70μ , and 80μ bubble radii.

- C23 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of bubble radius for 10, 20, 30, and 40 degree scattering angles.
- C24 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of bubble radius for 50, 60, 70, and 80 degree scattering angles.
- C25 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of bubble radius for 90, 100, 110, and 120 degree scattering angles.
- C26 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of bubble radius for 130, 140, 150, and 160 degree scattering angles.
- C27 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of bubble radius for 170 and 180 degree scattering angles.
- C28 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a 4 degree solid angle, as a function of bubble radius for 10, 20, 30, and 40 degree scattering angles.

- C29 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a 4 degree solid angle, as a function of bubble radius for 50, 60, 70, and 80 degree scattering angles.
- C30 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a 4 degree solid angle, as a function of bubble radius for 90, 100, 110, and 120 degree scattering angles.
- C31 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a 4 degree solid angle, as a function of bubble radius for 130, 140, 150, and 160 degree scattering angles.
- C32 Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a 4 degree solid angle, as a function of bubble radius for 170 and 180 degree scattering angles.

Introduction

Recently, several theoretical studies have been originated concerning the spherical stability of a pulsating air bubble in water. One of the earliest studies was done by Taylor (1950) who discussed the stability of a plane interface between two fluids of different densities undergoing an accelerated motion. Plesset (1954) extended Taylor's argument to describe the spherical stability of the motion of two immiscible, incompressible, inviscid fluids. A first order perturbation technique was used to describe the deviation from spherical symmetry based on a linearization of the radial motion of the bubble. Eller and Crum (1970) went further in describing the radial motion of the bubble by using a polytropic formulation of the Rayleigh-Plesset equation for inviscid liquids. They derived an analytical solution to the instability problem by using a third order expansion solution to the radial equation and by expressing the motion of the bubble surface as a perturbation to the spherical shape (i.e., $r = r_s(\theta, \phi, t) = R(t) + \sum_{n=2}^{\infty} a_n(t) Y_n(\theta, \phi)$). A description of their analytical solution and results will be given later. Prosperetti (1977) extended this formulation to include viscous effects. He assumed that the radial motion could be expressed as a superposition of spherical harmonics. Thus the motion of

the bubble surface could be expressed as: $r - R(t) - \epsilon a(t)Y_n^m(\theta, \phi) = 0$, with $0 < \epsilon \ll 1$. By expressing the velocity and pressure fields as sums of radial, potential, and viscous terms, Prosperetti was able to derive the equations of motion which included a radial equation and a shape distortion equation which represented the deviation from spherical symmetry. In this work, a procedure similar to Prosperetti's will be used with an additional irrotational flow restriction on the motion of a pulsating air bubble in water.

Experimental observations have shown that a pulsating air bubble in water will exhibit a radial instability (shape oscillation) threshold (Eller and Crum, 1970). The threshold is such that if the bubble's equilibrium radius remains constant and the driving pressure is increased, or if the driving pressure remains constant and the bubble's equilibrium radius changes, then there exists a sharp demarcation between regions where any perturbation to the bubble's spherical shape will grow and where the perturbation will quickly damp out.

Previous experimental observations of radial instability thresholds of a driven air bubble in water have been made both by Gould (1974) and by Eller and Crum (1970). In these observations, the thresholds were defined by the appearance of a "dancing" motion of the bubble. While the bubble was clearly exhibiting non-spherical motion at this "dancing" threshold, the radial instability threshold had occurred at a different point. The

"dancing" threshold criterion only allowed for nonstable shape oscillations in which there was significant motion of the center of mass of the bubble.

In this study, a light scattering technique was utilized to "photograph" in real-time the motion of the bubble surface. The main objective was to experimentally determine the shape oscillation threshold of an acoustically driven air bubble in water. These thresholds were defined as the minimum pressure necessary to drive a bubble, at a particular radius, into non-radial motion. To accomplish this, a single air bubble was acoustically levitated in a cell containing filtered water, a laser beam illuminated the bubble and the scattered light was detected by fast silicon photodiode detectors. Ordinary photography using a microscope and stroboscopic techniques was deemed insufficient to obtain radius vs. time curves to describe the motion of the bubble.

The results of this study show a marked improvement over previous experiments in detecting the radial instabilities resulting in shape oscillation. Further, by using a numerical integration technique applied to a simplified theory, the experimental shape oscillation thresholds were predicted quite well. In addition, the frequencies of the shape oscillations agree quantitatively over the range of bubble radii used (~10-100 microns, driven below resonance radius at 22.22 kHz).

This dissertation essentially contains three main sections: Theoretical Development and Discussion, Analytical and Numerical Solutions to the Theoretical Problem, and Experimental Results and Discussion. Each section is basically self contained, in that they need not be read in the order presented, although it is assumed that an orderly progression through the entire dissertation would be the common approach. There are, however, comparisons between sections, especially between the theoretical and experimental results.

The material is presented in the following sequence. First, a simple development of the Navier-Stokes equation is given in Chapter II. Next, in Chapter III, the equations of motion for a driven, pulsating air bubble in water are derived and simplifying assumptions are given to arrive at a set of two coupled, non-linear, second-order differential equations that describe the volume pulsations and shape distortions exhibited by a bubble. Chapter IV presents an analytical solution to the equations derived in Chapter III. The analytical solution decouples the equations by assuming a functional form for the bubble's radius as a function of time. This functional form is then inserted into the shape distortion equation which is solved in terms of spherical harmonics. Chapter V directly solves the equations in Chapter III by numerical integration. Chapters IV and V introduce the idea of a shape oscillation threshold (i.e., a radial instability threshold), and graphically present the shape oscillation thresholds as driving pressure vs. the equilibrium bubble radius. Chapter VI is a theoretical and

experimental description of the phenomenon of Mie scattering. The theoretical derivation of the scattering amplitudes associated with Mie scattering is the next logical step, but because of its complicated nature and relative importance, this derivation is regulated to Appendix B. Chapter VII details the experimental apparatus. Chapter VIII is the focal point of this dissertation. The other chapters lay groundwork and definitions for this experimental discussion chapter. First, the experimental methods are outlined, then a discussion of related experimental observations are given, and finally the experimental results are presented. Lastly, in Chapter IX, a summary of important points, a list of the major contributions, and a conclusion are given.

Formulation of the Problem

We will consider the problem of an interface separating two immiscible, viscous liquids of differing densities. Following a procedure similar to Morse and Ingard's (1986) approach we begin as follows. Assume that the velocity of the interface is a function of position and time:

$$\mathbf{u} = \mathbf{u}(\mathbf{r}, t). \quad (1.1)$$

In general a continuity equation can be expressed in terms of a tensor flux as

$$\frac{d\mathbf{f}}{dt} + \nabla \cdot \mathbf{I} = \mathbf{F} , \quad (1.2)$$

where $\mathbf{f}$ is some physical property, $\mathbf{F}$ is a source of $\mathbf{f}$, and $\mathbf{I} = \mathbf{B} + \mathbf{T}$ is a tensor which is the sum of the stress tensor ($\mathbf{B}$) and the momentum flux tensor ($\mathbf{T}$). If we let $\mathbf{f} = \rho \mathbf{u}$, the momentum per unit volume of the fluid, where ρ is the liquid density, and $\mathbf{F} = -\nabla(\rho\phi)$ be some external force such as gravity, where ϕ is some potential function, then Eq. (1.2) can be written as

$$\frac{d(\rho \mathbf{u})}{dt} + \nabla \cdot (\mathbf{B} + \mathbf{T}) = -\nabla(\rho \phi) , \quad (1.3)$$

where

$$\mathbf{B} \equiv \begin{bmatrix} P_{xx} & P_{xy} & P_{xz} \\ P_{yx} & P_{yy} & P_{yz} \\ P_{zx} & P_{zy} & P_{zz} \end{bmatrix} \text{ and } \mathbf{T} \equiv \begin{bmatrix} \rho u_x u_x & \rho u_x u_y & \rho u_x u_z \\ \rho u_y u_x & \rho u_y u_y & \rho u_y u_z \\ \rho u_z u_x & \rho u_z u_y & \rho u_z u_z \end{bmatrix}. \quad (1.4)$$

From Eq. (1.4), we see that $\mathbf{T}$ may be expressed as

$$\nabla \cdot \mathbf{T} = \mathbf{u} \nabla \cdot (\rho \mathbf{u}) + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} . \quad (1.5a)$$

Noting that by definition:

$$\frac{d(\rho \mathbf{u})}{dt} \equiv \rho \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \nabla \cdot (\rho \mathbf{u}) \quad (1.5b)$$

we see that Eq. (1.3) becomes

$$\rho \frac{d\mathbf{u}}{dt} = -\nabla(\rho \phi) - \nabla \cdot \mathbf{B} . \quad (1.6)$$

For inviscid liquids the stress tensor is simply

$$\mathbf{B} = \begin{bmatrix} P & 0 & 0 \\ 0 & P & 0 \\ 0 & 0 & P \end{bmatrix} , \quad (1.7)$$

where P is the pressure in the liquid. In general, the stress tensor, $\mathbf{B}$, is written as

$$\mathbf{B} = \left[P - \left(\eta - \frac{2}{3}\mu \right) \nabla \cdot \mathbf{u} \right] \Delta - \mu (\nabla \mathbf{u} + \mathbf{u} \nabla) , \quad (1.8)$$

where

$$\Delta_{ij} = \delta_{ij} \quad (\nabla \mathbf{u})_{ij} = \frac{\partial u_j}{\partial x_i} \quad (\mathbf{u} \nabla)_{ij} = \frac{\partial u_i}{\partial x_j} . \quad (1.9)$$

Here $\mu \equiv$ coefficient of shear viscosity and $\eta \equiv$ coefficient of bulk viscosity.

Using the relations

$$[\nabla \cdot (\mathbf{u} \nabla)]_j = [\nabla (\nabla \cdot \mathbf{u})]_j , \quad (1.10)$$

and

$$[\nabla \cdot (\nabla \mathbf{u})]_j = \nabla^2 u_j = [\nabla (\nabla \cdot \mathbf{u})]_j - [\nabla \times (\nabla \times \mathbf{u})]_j , \quad (1.11)$$

we find that

$$\nabla \cdot \mathbf{B} = \nabla \cdot \left\{ \left[P - \left(\eta - \frac{2}{3}\mu \right) \nabla \cdot \mathbf{u} \right] \Delta \right\} - \nabla \cdot [\mu (\nabla \mathbf{u} + \mathbf{u} \nabla)] \quad (1.12)$$

$$\begin{aligned}
&= \nabla P - \nabla \left(\eta - \frac{2}{3} \mu \right) \nabla \cdot \mathbf{u} - (\mu \nabla^2 \mathbf{u}) - (\mu \nabla \cdot \mathbf{u} \nabla) \\
&= \nabla P - \nabla \left(\eta + \frac{4}{3} \mu \right) \nabla \cdot \mathbf{u} + \mu \nabla \times \nabla \times \mathbf{u} ,
\end{aligned}$$

so that the continuity equation Eq. (1.6) becomes

$$\rho \frac{d\mathbf{u}}{dt} = -\nabla(\rho\phi) - [\nabla P - \nabla \left(\eta + \frac{4}{3} \mu \right) \nabla \cdot \mathbf{u} + \mu \nabla \times \nabla \times \mathbf{u}] . \quad (1.13)$$

Rearranging the terms in Eq. (1.13) we obtain the Navier-Stokes equation of the motion of a viscous, compressible fluid.

$$\rho \frac{d\mathbf{u}}{dt} = -\nabla \left[P + \rho\phi - \left(\eta + \frac{4}{3} \mu \right) \nabla \cdot \mathbf{u} \right] - \mu \nabla \times \nabla \times \mathbf{u} . \quad (1.14)$$

Next, using the relations:

$$\nabla \times \nabla \times \mathbf{u} \equiv \nabla(\nabla \cdot \mathbf{u}) - \nabla^2 \mathbf{u} , \quad (1.15)$$

and

$$\frac{d\mathbf{u}}{dt} \equiv \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} , \quad (1.16)$$

Eq. (1.14) becomes

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] = -\nabla \left[P + \rho \phi - \left(\eta + \frac{4}{3} \mu \right) \nabla \cdot \mathbf{u} \right] - \mu \nabla (\nabla \cdot \mathbf{u}) + \mu \nabla^2 \mathbf{u} , \quad (1.17)$$

or

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] = -\nabla (P + \rho \phi) - \nabla \left(\eta + \frac{1}{3} \mu \right) \nabla \cdot \mathbf{u} + \mu \nabla^2 \mathbf{u} . \quad (1.18)$$

The mass continuity equation may be written as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 . \quad (1.19)$$

If we assume that the density is essentially constant in space and time, then Eqs. (1.18) and (1.19) become

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] = -\nabla P - \rho \nabla \phi + \nabla \left(\eta + \frac{1}{3} \mu \right) \nabla \cdot \mathbf{u} + \mu \nabla^2 \mathbf{u} , \quad (1.20)$$

and

$$\nabla \cdot \mathbf{u} = 0 . \quad (1.21)$$

Combining Eqs. (1.20) and (1.21) we have

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] = -\nabla P - \rho \nabla \phi + \mu \nabla^2 \mathbf{u} . \quad (1.22)$$

If we neglect body forces such as gravity, Eq. (1.22) becomes

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] = -\nabla P + \mu \nabla^2 \mathbf{u} . \quad (1.23)$$

The problem is now defined; all that remains is to find an expression for the velocity field $\mathbf{u}$, and to apply the appropriate boundary conditions imposed by the problem of interest.

Equations of Motion

Introduction

The Navier-Stokes equation (1.14) describes the general motion of a viscous, compressible fluid element. For this particular study, we will describe the surface motion of a driven, pulsating air bubble in water. We will further assume that the densities of the water and air are essentially constant (i.e., the fluids are incompressible) so that Eq. (1.23) holds. While this incompressibility condition seems artificial at first, we note that the density of water is very much greater than the air inside the bubble and that the pressures used in this experiment do not change the density of the water appreciably. In addition, we will assume small amplitude volume pulsations for the bubble so that the density of the gas will not change appreciably. If the bubble is driven at sufficiently low pressure amplitudes then the steady-state motion of the bubble is spherically symmetric. The surface tension of the water/air interface and the viscosity of the water tend to keep the bubble spherical. Any small perturbation introduced to the surface will quickly damp out and the bubble will resume a spherical shape. In this study we are interested in determining the maximum driving pressure applied to the bubble for which the bubble will retain its

spherical shape. We will show that a pressure threshold exists so that the bubble does not gradually exhibit non-radial motion as a function of pressure; rather, a sharp demarcation exists between the point when any small perturbation will damp out and when that perturbation will either grow exponentially or become a stable shape oscillation.

For our theoretical model, we will assume that the bubble is slightly distorted from a spherical surface of radius $R(t)$ to a surface described by a superposition of spherical harmonics:

$$r_s \equiv R(t) + \epsilon a(t) Y_n^m(\theta, \phi) , \quad (2.1)$$

where $0 < \epsilon \ll 1$ and Y_n^m is a spherical harmonic of degree $n \geq 2$ ($n=1$ corresponds to a translational motion which is not of interest in this experiment). We will only concern ourselves with terms in the first order of ϵ in the following treatment. We will also make the assumption that no mass transfer takes place across the bubble surface. The predominant mass transport mechanism is known by the term "rectified diffusion". In this experiment it was shown that while this process changed the amount of air in the bubble (i.e., a volume change), the time scale ($\sim 10^2$ sec) for such processes is many orders of magnitude longer than the time scales used in this experiment ($\sim 10^{-3}$ sec). Appendix A describes the rectified diffusion measurements in more detail.

To solve for the equations of motion we begin by representing the velocity and pressure by

$$\mathbf{u} = \mathbf{u}_r + \varepsilon \mathbf{u}_p + \varepsilon \mathbf{u}_v , \quad (2.2)$$

$$P = P_r + \varepsilon P_p + \varepsilon P_v , \quad (2.3)$$

where the subscripts refer to radial terms, r, potential correction terms, p, and correction terms due to viscosity, v. From Eqs. (1.21) and (1.22) we have

$$\nabla \cdot \mathbf{u} = \nabla \cdot (\mathbf{u}_r + \varepsilon \mathbf{u}_p + \varepsilon \mathbf{u}_v) = 0 , \quad (2.4)$$

$$\rho \left[\frac{\partial (\mathbf{u}_r + \varepsilon \mathbf{u}_p + \varepsilon \mathbf{u}_v)}{\partial t} + [(\mathbf{u}_r + \varepsilon \mathbf{u}_p + \varepsilon \mathbf{u}_v) \cdot \nabla] (\mathbf{u}_r + \varepsilon \mathbf{u}_p + \varepsilon \mathbf{u}_v) \right] = \quad (2.5)$$

$$-\nabla (P_r + \varepsilon P_p + \varepsilon P_v) + \mu \nabla^2 (\mathbf{u}_r + \varepsilon \mathbf{u}_p + \varepsilon \mathbf{u}_v) ,$$

which become the following six equations (to first order in ε):

$$\nabla \cdot \mathbf{u}_r = 0 , \quad (2.6)$$

$$\nabla \cdot \mathbf{u}_p = 0 , \quad (2.7)$$

$$\nabla \cdot \mathbf{u}_v = 0 , \quad (2.8)$$

$$\rho \left[\frac{\partial \mathbf{u}_r}{\partial t} + (\mathbf{u}_r \cdot \nabla) \mathbf{u}_r \right] = -\nabla P_r + \mu \nabla^2 \mathbf{u}_r , \quad (2.9)$$

$$\rho \left[\frac{\partial \mathbf{u}_p}{\partial t} + (\mathbf{u}_r \cdot \nabla) \mathbf{u}_p + (\mathbf{u}_p \cdot \nabla) \mathbf{u}_r \right] = -\nabla P_p + \mu \nabla^2 \mathbf{u}_p , \quad (2.10)$$

$$\rho \left[\frac{\partial \mathbf{u}_v}{\partial t} + (\mathbf{u}_r \cdot \nabla) \mathbf{u}_v + (\mathbf{u}_v \cdot \nabla) \mathbf{u}_r \right] = -\nabla P_v + \mu \nabla^2 \mathbf{u}_v . \quad (2.11)$$

Radial terms

Equation (2.9) is simply the radial equation. To find u_r , using Eq. (2.6) in spherical coordinates with the origin located at the center of the bubble and assuming $\mathbf{u} = u(\mathbf{r}, t)$, then:

$$\nabla \cdot \mathbf{u} = \frac{\partial(r^2 u)}{r^2 \partial r} = \frac{\partial u}{\partial r} + \frac{2u}{r} = 0. \quad (2.12)$$

For the rest of this section the subscript on u_r will be dropped. Integrating Eq. (2.12) over velocity and radial limits we obtain

$$\int_{u_0}^u \frac{du}{u} = -2 \int_{r_0}^r \frac{dr}{r}, \quad (2.13)$$

which gives

$$u = u_0 \left(\frac{r_0}{r} \right)^2. \quad (2.14)$$

Noting that $r_0 = R$ and $u_0 = dR/dt = R'^1$ then Eq. (2.14) becomes

¹Note: in general, time derivatives will be denoted by primes.

$$u = \frac{R^2 R'}{r^2}, \quad (2.15)$$

where $R=R(t)$ is the position of the bubble wall as a function of time. Now that we have an expression for the fluid velocity u_r , we can obtain the following relations:

$$(u \cdot \nabla)u = u \frac{\partial u}{\partial r} = -2 \frac{R^4 R'^2}{r^5}, \quad (2.16)$$

$$\nabla^2 u = \frac{\partial(r^2 u)}{r^2 \partial r} = 0, \quad (2.17)$$

and

$$\frac{\partial u}{\partial t} = \frac{R^2 R''}{r^2} + \frac{2RR'^2}{r^2}. \quad (2.18)$$

Inserting Eqs. (2.16-2.18) into Eq. (2.9) we have:

$$\rho \left[\frac{R^2 R''}{r^2} + 2 \frac{RR'^2}{r^2} - \frac{2R^4 R'^2}{r^5} \right] = - \frac{\partial P}{\partial r}. \quad (2.19)$$

Integrating r from ∞ to R and P from P_∞ to P gives

$$P = P_\infty + \rho \left(R''R + 2R'^2 - \frac{1}{2}R'^2 \right). \quad (2.20)$$

To determine the boundary condition at the bubble surface we use the requirement of continuity of stress across the interface. From Eq. (1.8) the stress tensor is given by

$$\mathbf{B} = \left[P - \left(\eta - \frac{2}{3}\mu \right) \nabla \cdot \mathbf{u} \right] \Delta - \mu (\nabla \mathbf{u} + \mathbf{u} \nabla) . \quad (2.21)$$

The radial component of $\mathbf{B}$ is

$$B_{rr} = \left[P - \left(\eta - \frac{2}{3}\mu \right) \nabla \cdot \mathbf{u}_r \right] - \mu (\nabla u_r + u_r \nabla) . \quad (2.22)$$

From the relations in Eq. (1.9) we find

$$(\nabla \mathbf{u})_{r,r} = \frac{\partial u_r}{\partial r} , \quad (\mathbf{u} \nabla)_{r,r} = \frac{\partial u_r}{\partial r} , \quad (2.23)$$

and from Eq. (2.12) we see that

$$\nabla \cdot \mathbf{u}_r = \frac{\partial u_r}{\partial r} + \frac{2u_r}{r} = 0 . \quad (2.24)$$

Thus, B_{rr} becomes

$$B_{rr} = P - 4\mu \frac{R^2 R'}{r^3} . \quad (2.25)$$

At the bubble surface, $r=R$ we have

$$B_{RR} = P - 4\mu \frac{R'}{R} , \quad (2.26)$$

where $P=P_i$ is the pressure inside the bubble. Applying the dynamical boundary condition that the discontinuity in the normal stress (B_{rr}) will equal the surface tension σ times the total curvature of the surface we have

$$B_{rr}^{(1)} - B_{rr}^{(2)} = \frac{2\sigma}{R} , \quad (2.27)$$

(for $r=R$, (1) = inside, (2) = outside)

or

$$\left(P_i - 4\mu \frac{R'}{R} \right) - P = \frac{2\sigma}{R} , \quad (2.28)$$

where P is the pressure in the liquid. This equation may be used to describe the pressure at the bubble's surface:

$$P = P_i - \frac{2\sigma}{R} - 4\mu \frac{R'}{R} . \quad (2.29)$$

Equating Eqs. (2.20) and (2.29) and rearranging terms we arrive at a form of the Rayleigh-Plesset equation

$$R''R + \frac{3}{2}R'^2 = \frac{1}{\rho} \left[P_i - P_\infty - \frac{2\sigma}{R} - 4\mu \frac{R'}{R} \right] . \quad (2.30)$$

Potential correction terms – deviation from spherical symmetry

Next, we turn our attention to Eq. (2.10) which represents the deviation from sphericity. As before, the problem is to determine u_p . To begin, we will assume that the position of the surface can be represented by

$$r_s = R(t) + \epsilon a(t) Y_n^m(\theta, \phi) . \quad (2.31)$$

We will assume that $\mathbf{u}$ may be described by the gradient of a potential $\mathbf{u} = \nabla \phi$ and that Laplace's equation $\nabla^2 \phi = 0$ holds. Thus we will have solutions of Laplace's equation on both sides of the bubble wall interface so that the potential terms ϕ_1 and ϕ_2 may be written as (Plesset, 1954)

$$\phi_1 = \frac{R^2 R'}{r} + b_1 r^n Y_n \quad r < R , \quad (2.32)$$

$$\phi_2 = \frac{R^2 R'}{r} + \frac{b_2 Y_n}{r^{n+1}} \quad r > R . \quad (2.33)$$

This form of the velocity potentials assumes axisymmetry (i.e., the spherical harmonic $Y_n^m(\theta, \phi) = Y_n(\theta)$). While this assumption may not be strictly true, as shown in experimental studies by Holt (1989), this

approximation may prove to be quite good for small perturbations to the spherical shape. Using the boundary condition

$$\left. \frac{\partial \phi_1}{\partial r} \right)_{r_s} = \left. \frac{\partial \phi_2}{\partial r} \right)_{r_s} = u = R' + \epsilon a' Y_n , \quad (2.34)$$

and using Eqs. (2.32) and (2.33), we can solve for b_1 and b_2 as follows:

$$\frac{R^2 R'}{r^2} - n b_1 r^{n-1} Y_n = \frac{R^2 R'}{r^2} + \frac{b_2 Y_n}{r^{n+2}} (n+1) = R' + \epsilon a' Y_n , \quad (2.35)$$

$$b_1 = - \frac{b_2 (n+1)}{n r^{2n+1}} , \quad (2.36)$$

and

$$b_2 = \frac{\left(R' + \epsilon a' Y_n - \frac{R^2 R'}{r^2} \right)}{\frac{(n+1) Y_n}{r^{n+2}}} . \quad (2.37)$$

Inserting Eq. (2.31) into Eqs. (2.36) and (2.37) and simplifying we obtain:

$$b_2 = \frac{R^{n+2} \epsilon (2aR'/R + a')}{(n+1)} . \quad (2.38)$$

Next, to obtain an expression for b_1 in Eq. (2.36), using Eqs. (2.38) and (2.31), we find

$$b_1 = -\epsilon \frac{2aR'/R + a'}{nR^{n-1}}. \quad (2.39)$$

Therefore inserting Eqs. (2.38) and (2.39) into Eqs. (2.32) and (2.33) we arrive at the following:

$$\phi_1 = \frac{R^2 R'}{r} - \epsilon \frac{r^n Y_n}{n R^{n-1}} [2aR'/R + a'], \quad (2.40)$$

$$\phi_2 = \frac{R^2 R'}{r} + \epsilon \frac{R^{n+2} Y_n}{(n+1)r^{n+1}} [2aR'/R + a']. \quad (2.41)$$

The deviation from sphericity is the second term in Eqs. (2.40) and (2.41) so that ϕ_1 and ϕ_2 may be rewritten as

$$\phi_1 = -\epsilon \frac{r^n Y_n}{n R^{n-1}} [2aR'/R + a'], \quad (2.42)$$

$$\phi_2 = \epsilon \frac{R^{n+2} Y_n}{(n+1)r^{n+1}} [2aR'/R + a']. \quad (2.43)$$

Inserting these into Eq. (2.10) we find

$$\rho \left[\frac{\partial(\nabla\phi)}{\partial t} + \frac{R^2 R'}{r^2} \nabla(\nabla\phi) + \nabla\phi \left(-2 \frac{R^2 R'}{r^3} \right) \right] = -\nabla P + \mu \nabla^2 \nabla(\phi). \quad (2.44)$$

Equation (2.44) may be integrated to give (using only the spherical deviation parts)

$$P_p = -\rho \left[\frac{\partial \phi}{\partial t} + \frac{R^2 R'}{r^2} \frac{\partial \phi}{\partial r} \right] . \quad (2.45)$$

By inserting Eqs. (2.42) and (2.43) into Eq. (2.45) one can arrive at the following equations:

$$P_{p1} = -\frac{\rho_1}{n} (Ra'' + 3R'a' + 2R''a) Y_n , \quad (2.46)$$

$$P_{p2} = \frac{\rho_2}{(n+1)} (Ra'' + 3R'a' + 2R''a) Y_n . \quad (2.47)$$

Combining Eqs. (2.46) and (2.47), and using $P_p = P_{p1} - P_{p2}$, we have

$$P_p = \left(\frac{\rho_2}{(n+1)} + \frac{\rho_1}{n} \right) (Ra'' + 3R'a' + 2R''a) Y_n . \quad (2.48)$$

Potential correction terms due to viscosity

The viscous correction terms, u_v and p_v , are difficult to derive. Prosperetti (1977) gives an excellent treatment of the problem. Using the results from Prosperetti one can arrive at

$$P_{v1} = (n+1)\rho_1 Y_n \left\{ -v_1 \frac{T_1(R,t)}{R} + \frac{R'}{R} \int_0^R \left(\frac{s}{R} \right)^{n-2} \left[1 - \left(\frac{s}{R} \right)^3 \right] T_1(s,t) ds \right\} , \quad (2.49)$$

$$P_{v2} = n\rho_2 Y_n \left\{ v_2 \frac{T_2(R,t)}{R} + \frac{R'}{R} \int_R^\infty \left(\frac{R}{s} \right)^n \left[\left(\frac{R}{s} \right)^3 - 1 \right] T_2(s,t) ds \right\} , \quad (2.50)$$

where $v = \mu/\rho$ is the kinematic viscosity and T is the toroidal field on the inner (1) and outer (2) side of the bubble wall. The boundary condition that requires the tangential stress vanish at the bubble surface requires that $T(r,t)$ satisfy

$$2vR^{n-1} \int_R^\infty s^{-n} T(s,t) ds + vT(R,t) = \frac{2}{n+1} v \left[(n+2)a'_n - (n-1) \frac{R'a_n}{R} \right] . \quad (2.51)$$

If we ignore the integral contributions in Eqs. (2.49 - 2.51) and eliminate $T(R,t)$ in Eqs. (2.49) and (2.50) by using Eq. (2.51) we have

$$P_{v1} = (n+1)\rho_1 Y_n \left\{ \frac{2v_1}{(n+1)R} \left[(n+1)a'_n - (n-1) \frac{R'a_n}{R} \right] \right\} , \quad (2.52)$$

$$P_{v2} = n\rho_2 Y_n \left\{ \frac{2v_2}{(n+1)R} \left[(n+1)a'_n - (n-1) \frac{R'a_n}{R} \right] \right\} . \quad (2.53)$$

Thus $P_v = P_{v1} - P_{v2}$ is

$$P_v = \left[\frac{2\mu_1}{R} - \frac{2n\mu_2}{(n+1)R} \right] \left[(n+1)a'_n - (n-1) \frac{R'a_n}{R} \right] Y_n . \quad (2.54)$$

Equations of motion

Combining Eqs. (2.11), (2.48), and (2.54) and applying the boundary conditions at the bubble interface described above we may write the spherical deviation terms (those multiplied by Y_n) as²

$$\left[\frac{\rho_2}{(n+1)} + \frac{\rho_1}{n} \right] R a'' + \left[3 \left(\frac{\rho_2}{(n+1)} + \frac{\rho_1}{n} \right) R' - \frac{2}{R} (n-1)(n+2)(\mu_2 - \mu_1) \right] a' + \left[\left(\frac{n+2}{n} \rho_1 - \frac{n-1}{n+1} \rho_2 \right) R'' + 2(n-1)(n+2)(\mu_2 - \mu_1) \frac{R'}{R^2} + (n-1)(n+2) \frac{\sigma}{R^2} \right] a = 0. \quad (2.55)$$

For an air bubble in water $\rho_1 \ll \rho_2$ and $\mu_1 \ll \mu_2$. So, setting $\rho_1 = 0$ and $\mu_1 = 0$ in Eq. (2.55) we have

$$\left[\frac{\rho}{(n+1)} \right] R a'' + \left[3 \left(\frac{\rho}{(n+1)} \right) R' - \frac{2}{R} (n-1)(n+2)\mu \right] a' + \left[- \left(\frac{n-1}{n+1} \right) \rho R'' + 2(n-1)(n+2)\mu \frac{R'}{R^2} + (n-1)(n+2) \frac{\sigma}{R^2} \right] a = 0, \quad (2.58)$$

which can be rearranged to obtain

²Note that if we ignore viscosity (i.e., $\mu_1 = \mu_2 = 0$) then the above equation can be written as

$$a'' + \frac{3R'}{R} a' + A a = 0, \quad (2.56)$$

where

$$A = \frac{[n(n-1)\rho_2 - (n+1)(n+2)\rho_1]R'' - (n-1)n(n+1)(n+2)\frac{\sigma}{R^2}}{R[n\rho_2 + (n+1)\rho_1]}, \quad (2.57)$$

which is the form found by Plesset (1954).

$$a'' + \left[\frac{3R'}{R} - 2(n-1)(n+1)(n+2)\frac{v}{R^2} \right] a' + (n-1) \left[\frac{-R''}{R} + 2(n+1)(n+2)\frac{R'}{R^3} + (n+1)(n+2)\frac{\sigma}{\rho R^3} \right] a = 0. \quad (2.59)$$

Thus, Eqs. (2.30) and (2.59) will be used to describe the motion of a pulsating air bubble in water.

Recapitulation

To re-iterate the basic assumptions, we have assumed that

- the bubble is basically spherical with small amplitude oscillations
- the deviation from sphericity can be expressed in terms of a superposition of spherical harmonics
- the density of the water is much greater than that of air
- the viscosity of water is much greater than that of air
- rotational flow and toroidal fields can be ignored
- there are no body forces acting on the bubble
- initial spherical deviations are very small compared to the equilibrium radius of the bubble
- the bubble behaves axisymmetrically (i.e., $Y_n^m = Y_n$)
- there is no mass transfer across the interface

- the equations of motion describing the motion of the bubble surface are given by:

$$R''R + \frac{3}{2}R'^2 = \frac{1}{\rho} \left[P_i - P_\infty - \frac{2\sigma}{R} - 4\mu \frac{R'}{R} \right], \quad (2.60)$$

$$a'' + \left[\frac{3R'}{R} - 2(n-1)(n+1)(n+2) \frac{v}{R^2} \right] a' + \quad (2.61)$$

$$(n-1) \left[\frac{-R''}{R} + 2(n+1)(n+2) \frac{R'}{R^3} + (n+1)(n+2) \frac{\sigma}{\rho R^3} \right] a = 0 \quad (n \leq 2).$$

Analytical and numerical solutions will be presented later and compared with experimental data.

Analytical Solution

Introduction

The instability of the radial motion of a pulsating bubble in a sound field was studied by Eller and Crum in 1970. Following their theoretical outline an analytical solution to the equations of motion, Eqs. (2.60) and (2.61), can be found. This approach can be summarized as follows: We begin with the form of the Rayleigh-Plesset equation given as Eq. (2.60):

$$R''R + \frac{3}{2}R'^2 = \frac{1}{\rho} \left[P_i - P_\infty - \frac{2\sigma}{R} - 4\mu \frac{R'}{R} \right] . \quad (3.1)$$

Here $R=R(t)$ is the bubble radius, ρ is the density of the water, P_i is the pressure inside the bubble, P_∞ is the pressure in the water far from the bubble (i.e., the driving pressure), σ is the surface tension, μ is the viscosity of the water, and the primes indicate differentiation with respect to time. The quantities ρ , σ and μ are considered to be constants and P_∞ is given. The internal bubble pressure and a functional form of the bubble surface are needed to solve the equation.

Internal bubble pressure

Characterizing the pressure inside the bubble, P_i , has received considerable attention (Prosperetti, et al., 1988). If one assumes that the bubble's radial oscillation is small compared with its equilibrium radius then one may assume that the pressure inside the bubble obeys a polytropic law. This simply means that the pressure inside the bubble can be related to its volume by $PV^\kappa = \text{constant}$. When an isothermal approximation is made, the polytropic exponent is $\kappa=1$. For an adiabatic approximation the polytropic exponent is simply the ratio of specific heats, $\kappa=\gamma=c_p/c_v$. Thus, the pressure inside the bubble can be expressed as

$$P_i = P_0 \left[\frac{R_0}{R} \right]^{3\kappa}, \quad (3.2)$$

where $1 \leq \kappa \leq \gamma$, $P_0 = P_{\text{amb}} + 2\sigma/R_0$ is the initial equilibrium pressure inside the bubble, P_{amb} is the ambient pressure in the liquid, and R_0 is the equilibrium bubble radius. The actual value of κ has been studied by various researchers (Devin 1959, Eller 1970, Crum 1983, and Prosperetti 1986). In this study the form derived by Eller (1970) was used:

$$\kappa = \gamma(1+d_{th}^2)^{-1} \left[1 + 3 \frac{(\gamma-1)}{X} \left(\frac{\sinh X - \sin X}{\cosh X - \cos X} \right) \right]^{-1}, \quad (3.3)$$

with the following definitions:

$$X = R_0(2\omega/D_1)^{1/2} , \quad (3.4)$$

$$D_1 = \frac{K_1}{\rho_1 c_{p1}} , \quad (3.5)$$

$$d_{th} = 3(\gamma-1) \left[\frac{X(\sinh X + \sin X) - 2(\cosh X - \cos X)}{X^2(\cosh X - \cos X) + 3(\gamma-1)X(\sinh X - \sin X)} \right] , \quad (3.6)$$

where R_0 is the equilibrium bubble radius, $\omega=2\pi f$ is the angular driving frequency, K_1 is the thermal conductivity of the gas in the bubble, ρ_1 is the gas density, and c_{p1} is the specific heat at constant pressure of the gas. The various formulations give essentially equivalent values of the polytropic exponent for the conditions of interest in this study.

Third order approximate solution

Radial equation

If we assume that the bubble is driven by a sinusoidally varying pressure field given by:

$$P_\infty = P_{amb}[1 + P_a \cos(\omega t)] , \quad (3.7)$$

where P_a is the amplitude of the driving pressure, then Eq. (3.1) may be written as

$$R''R + \frac{3}{2}R'^2 = \frac{1}{\rho} \left\{ P_0 \left[\frac{R_0}{R} \right]^{3\kappa} - P_{amb}[1 + P_a \cos(\omega t)] - \frac{2\sigma}{R} - 4\mu \frac{R'}{R} \right\} . \quad (3.8)$$

If we ignore surface tension and viscosity (following Eller and Crum), then Eq. (3.8) becomes

$$R''R + \frac{3}{2}R'^2 + \frac{P_{amb}}{\rho} \left\{ 1 - \left[\frac{R_0}{R} \right]^{3\kappa} + P_a \cos(\omega t) \right\} = 0 , \quad (3.9)$$

which represents the radial equation of motion. This equation can be numerically integrated to obtain $R=R(t)$.

Perturbation equation

To describe the bubble's deviation from sphericity, we will first assume that the liquid is inviscid so that Eq. (2.61) becomes

$$a'' + \frac{3R'}{R} a' + (n-1) \left[\frac{-R''}{R} + (n+1)(n+2) \frac{\sigma}{\rho R^3} \right] a = 0 . \quad (3.10)$$

If we define a new variable $y_n(t)$ as

$$y_n(t) = R^{\frac{3}{2}}(t)a_n(t) , \quad (3.11)$$

then Eq. (3.10) becomes

$$y''_n + \left[\frac{(n+2)(n^2-1)\sigma}{\rho R^3} - \frac{3}{4} \left(\frac{R'}{R} \right)^2 - \left(n + \frac{1}{2} \right) \frac{R''}{R} \right] y_n = 0 . \quad (3.12)$$

This equation represents the spherical deviation equation of motion. Thus Eqs. (3.9) and (3.12) are the coupled equations of motion describing a pulsating air bubble in water. To derive an analytical solution one may assume that steady state radial oscillations can be approximated by the following truncated expression (Hall and Seminara, 1980):

$$R/R_0 = (1 + A^2b) + A \cos(\omega t) + A^2c \cos(2\omega t) + A^2d \cos(3\omega t) , \quad (3.13)$$

where

$$A = \frac{P_a}{3\gamma(\beta^2 - 1)} , \quad (3.14)$$

$$b = \frac{(3\gamma + 1 - \beta^2)}{4} , \quad (3.15)$$

$$c = \frac{(3\gamma + 1 + 5\beta^2)}{4(1 - \beta^2)} , \quad (3.16)$$

$$d = \frac{12(3\gamma + 1 + 11\beta^2)c - (3\gamma + 1)(3\gamma + 2)}{24(1 - 9\beta^2)} , \quad (3.17)$$

and

$$\beta = \omega/\omega_0 . \quad (3.18)$$

The angular resonance frequency, ω_0 , of the oscillating bubble is given by

$$\omega_0 = \sqrt{\frac{3\gamma P_0}{\rho R_0^2}} . \quad (3.19)$$

Inserting Eq. (3.13) into Eq. (3.12) and making the following substitution:

$$z = \frac{1}{2}\omega t , \quad (3.20)$$

one will arrive at the following (Hayashi, 1964):

$$y''_n + (\theta_0 + 2\theta_2\cos(2z) + 2\theta_4\cos(4z) + \dots)y_n = 0 . \quad (3.21)$$

This is simply a linear, second-order differential equation with periodic coefficients and can be written as a form of Hill's equation:

$$\frac{d^2y}{dz^2} + \left(\theta_0 + 2 \sum_{j=1}^{\infty} \theta_{2j}\cos(2jz) \right) y = 0 . \quad (3.22)$$

The θ_{2j} 's are given to third order in A (Eller and Crum, 1969) by

$$\theta_0 = a + \left[3a(1-b) - (2n + \frac{5}{2}) \right] A^2 , \quad (3.23)$$

$$\theta_2 = (2n + 1 - \frac{3}{2}a)A + \left[3a(2b + c - \frac{5}{4}) - 3c + \frac{3}{4} - (n + \frac{1}{2})(5c + 2b - \frac{3}{2}) \right] A^3 , \quad (3.24)$$

$$\theta_4 = \left[\frac{3}{2}a(1-c) + \frac{3}{4} + (n + \frac{1}{2})(8c - 1) \right] A^2 , \quad (3.25)$$

and

$$\theta_6 = \left[a(3c - \frac{3}{2}d - \frac{5}{4}) + 3c - \frac{3}{4} + (n + \frac{1}{2})(18d + \frac{1}{2} - 5c) \right] A^3 , \quad (3.26)$$

where

$$a = \frac{4\sigma(n+1)(n^2-1)}{\rho\omega^2 R_0^3} . \quad (3.27)$$

Solutions to Hill's equation

The solutions to Hill's equation are examined in detail by McLachlan, (1947). A solution to Hill's equation is:

$$y_n(z) = e^{vz} \sum_{j=-\infty}^{\infty} c_{2j} e^{2jzi} , \quad (3.28)$$

where the c_{2j} 's are expansion coefficients and v may be real, imaginary, or complex. For our purposes, Eq. (3.28) may be rewritten as

$$y_n(z) = e^{vz} \left[c_0 + 2 \sum_{j=1}^{\infty} c_{2j} e^{2jzi} \right], \quad (3.29)$$

using the fact that $c_{2j} = c_{-2j}$. This solution will be defined as unstable if it tends to ∞ as $z \rightarrow \infty$. The solution will be defined to be stable if it tends to zero or remains bounded as $z \rightarrow \infty$. From McLachlan, the solution is unstable if v is any real number not equal to 0, or if v is complex. The solution is stable and periodic if $v = i\beta$ and if β is a rational fraction. The solution is also stable and non-periodic if $v = i\beta$ with β irrational. Since we are concerned with shape instabilities we will examine the unstable solutions. Since later we will be comparing v with a physical quantity, we will limit ourselves to the solutions for which v is real. Equation (3.29) may be written as:

$$y_n = e^{vz} \Phi(z, \sigma), \quad (3.30)$$

where

$$\Phi(z, \sigma) = \sin(jz - \sigma) + \theta_2 f_2(z, \sigma) + \theta_4 f_4(z, \sigma) + \dots + O(\theta^2) + \dots \quad (3.31)$$

Equation (3.30) may be approximated by using Eq. (3.31) as

$$y_n = e^{\nu z} \sin(jz - \sigma) = e^{\nu \omega t/2} \sin(j\omega t/2 - \sigma) , \quad (3.32)$$

where σ is some phase angle. Since σ can be arbitrary, the problem reduces to finding an expression for ν since all other quantities are given. The general theory of Mathieu functions can be used to obtain expressions for ν . Eller and Crum used the results of Hayashi (1964) to obtain approximate solutions for ν . Since the solutions to Hill's equations have regions of stability and instability, the idea is to find the values of ν in the unstable regions and relate these to known parameters, namely, R_0 and P_a . Only the first three unstable regions were considered ($j=1,2,3$). The first three regions correspond to angular frequencies of $\omega/2$, ω , and $3\omega/2$. The values for ν for each j may be summarized as follows: For a given j , the value of ν may be given by

$$\nu_j^2 = S_{1j}^2 - S_{2j}^2 . \quad (3.33)$$

S_{1j} and S_{2j} can be approximated by only considering the coefficients, θ_0 , θ_2 , θ_4 , and θ_6 and by considering only terms up to second order in θ . Thus, the values of S_{1j} and S_{2j} are given by:

for $j=1$:

$$2S_{11} = \theta_2 + \frac{1}{4}\theta_2\theta_4 + \frac{1}{12}\theta_4\theta_6 , \quad (3.34)$$

$$2S_{21} = \theta_0 - 1 + \frac{1}{8}\theta_2^2 + \frac{1}{6}\theta_4^2 + \left(\frac{v_1}{2S_{11}}\right)^2 \theta_2 , \quad (3.35)$$

for $j=2$:

$$4S_{12} = \theta_4 - \frac{1}{4}\theta_2^2 + \frac{1}{6}\theta_2\theta_6 , \quad (3.36)$$

$$4S_{22} = \theta_0 - 4 - \frac{1}{6}\theta_2^2 + \frac{1}{32}\theta_4^2 - \frac{1}{10}\theta_6^2 + \left(\frac{v_2}{4S_{12}}\right)^2 \theta_4^2 , \quad (3.37)$$

and for $j=3$:

$$6S_{13} = \theta_6 - \frac{1}{4}\theta_2\theta_4 , \quad (3.38)$$

$$6S_{23} = \theta_0 - 9 - \frac{1}{16}\theta_2^2 = \frac{1}{10}\theta_4^2 + \frac{1}{72}\theta_6^2 + \left(\frac{v_3}{6S_{13}}\right)^2 \theta_6^2 . \quad (3.39)$$

Next, we obtain a polynomial equation for each v_j :

$$\frac{1}{4}\left(\frac{\theta_2}{2S_{11}}\right)^4 v_1^4 + \left(1 + \frac{e\theta_2^2}{4S_{11}^2}\right)v_1^2 + (e^2 - S_{11}^2) = 0 , \quad (3.40)$$

$$\frac{1}{16}\left(\frac{\theta_2}{4S_{12}}\right)^4 v_2^4 + \left(1 + \frac{f\theta_4^2}{32S_{12}^2}\right)v_2^2 + (f^2 - S_{12}^2) = 0 , \quad (3.41)$$

$$\frac{1}{16}\left(\frac{\theta_6}{6S_{13}}\right)^4 v_3^4 + \left(1 + \frac{g\theta_6^2}{108S_{13}^2}\right)v_3^2 + (g^2 - S_{13}^2) = 0 , \quad (3.42)$$

where

$$e = \theta_0/2 - 1/2 + \theta_2^2/16 + \theta_4^2/12 , \quad (3.43)$$

$$f = \theta_0/4 - 1 + \theta_2^2/24 + \theta_4^2/128 + \theta_6^2/40 , \quad (3.44)$$

$$g = \theta_0/6 - 3/2 + \theta_2^2/96 + \theta_4^2/60 + \theta_6^2/432 . \quad (3.45)$$

Solutions to Eqs. (3.40 - 3.42) will give the values of v for particular values of n and j .

Threshold criteria

Now that we have solutions for the values of v , we must establish a criterion for the location of the unstable regions as a function of bubble radius, R_0 , and applied pressure, P_a . Equation (3.32) represents a solution that will tend to infinity if v is positive (negative values for v will be discarded as will be seen later). This results in describing a shape oscillation whose amplitude exponentially grows in time with angular frequency $j\omega/2$. Due to the absence of viscosity in Eq. (3.12), the shape oscillations, once begun, would increase exponentially in amplitude over time. Experimental results show that this is not the case. To overcome this problem, Eller and Crum followed an example from Lamb (1945) stating that small perturbations to the shape of a pulsating bubble would, in a slightly viscous liquid, damp out exponentially as $e^{-\alpha t}$. The decay constant α is given by

$$\alpha = \frac{(n+2)(2n+1)\mu}{R_0^2}, \quad (3.46)$$

where μ is the liquid viscosity, R_0 is the equilibrium radius of the bubble and $n \geq 2$ is the spherical harmonic mode number. If it is assumed that the onset of a shape oscillation occurs whenever the growth rate of the unstable solutions to Hill's equation ($\nu\omega t/2$) is greater than the damping term ($-\alpha t$), then the instability condition for a given value of n is

$$\nu\omega t/2 > -\alpha t, \quad (3.47)$$

or

$$\nu > \frac{2(n+2)(2n+1)\mu}{\omega R_0^2}. \quad (3.48)$$

Thus, the thresholds for instability are determined by values of ν that satisfy Eqs. (3.40), (3.41) and (3.42) subject to the condition required by Eq. (3.48).

Solutions to the analytical threshold equations

Equations (3.40-3.42) may be numerically solved. The object is to find the minimum value of P_a given values of R_o , j , and n which would satisfy the instability criteria in Eq. (3.48). A computer program was developed to solve this system of equations. Using the program DTHRESH found in Appendix D, these equations were computed to find threshold regions for all combinations of $n=2-4$ and $j=1-3$. This resulted in a description of the minimum instability threshold regions in P_a vs. R_o space. The appropriate constants used were as follows:

Ambient pressure	P_o	1.0 bar
Air density	ρ_g	0.001204 g/cm ³
Ambient temperature	T	293 K
Driving frequency	f	$\omega/2\pi = 22.22$ kHz
Gas constant	R_g	8.314×10^7 dyn/mol K
Kinematic viscosity	μ	0.0091 cm ² /sec
Liquid density	ρ	1.0 g/cm ³
Molecular weight of air	M	28.964 g/mol
Specific heats of air		
at constant volume	C_{vg}	20.8×10^7 dyn/mol K
at constant pressure	C_{pg}	2.91×10^8 dyn/mol K
Speed of sound in water	c	1.5×10^5 cm/s
Surface tension	σ	72.5 dyn/cm
Thermal conductivity of air	K_1	2.5×10^3 dyn/(sec cm K)

The value of R_o ranged from 10 to 100 microns. Pressures greater than 0.8 bar were not treated.

It can be seen that solving Eqs. (3.40 - 3.42) will result in both real and imaginary values for v as well as positive and negative values. All imaginary results were ignored. In calculating the real results, an interesting feature of the solutions was noted. The solution space of v as a function of pressure for a particular radius was normally described by that shown in Fig. 1.

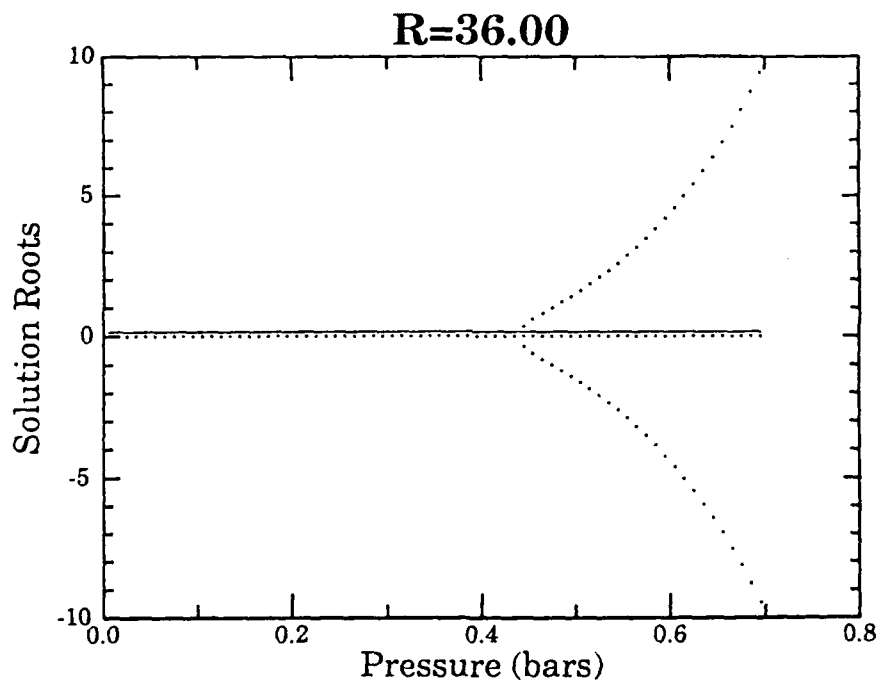


Figure 1: Roots of the analytical solution, exhibiting a single threshold point.

The graph represent all four solutions for v . A value of zero indicates an imaginary solution value. The solution space is actually three dimensional (x =pressure, y =real solutions, z =imaginary solutions). The projection shown is the "real" slice with the "imaginary" slice perpendicular to the

paper. From this figure it is easily seen that as the applied pressure increases, the instability increases once a threshold pressure has been reached. There are regions, though, in which this is not true. Figure 2 shows an example of this behavior.

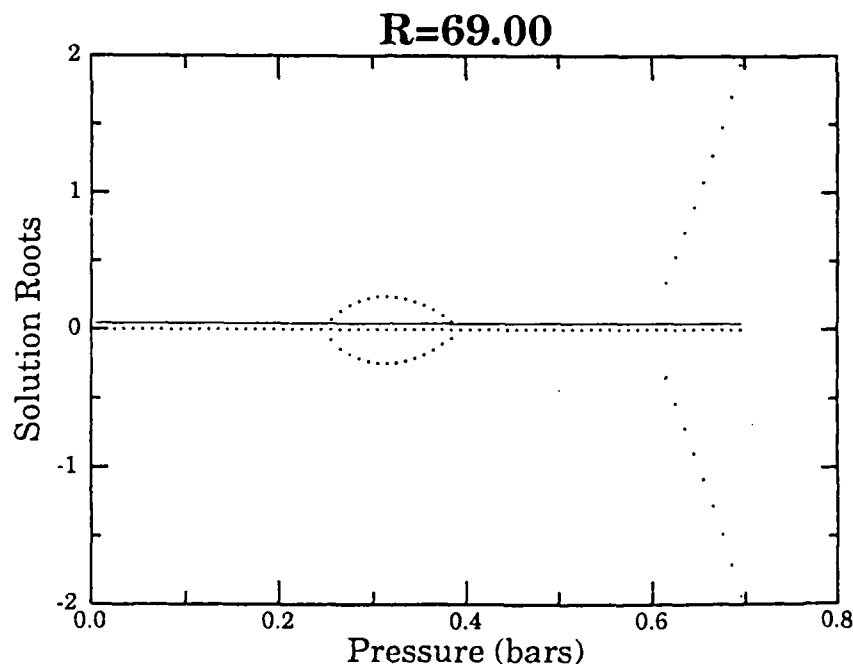


Figure 2 Roots of the analytical solution, exhibiting multiple threshold points.

Here we noticed the appearance of a feature labeled "bubbling". For certain ranges of radii, it was seen that there was a "pocket" of instability preceding the "main" instability point. Whether this bubbling was simply an artifact of the mathematics or a real physical feature was not clear. If we ignored this bubbling feature then an example of the threshold graph is shown in Fig. 3.

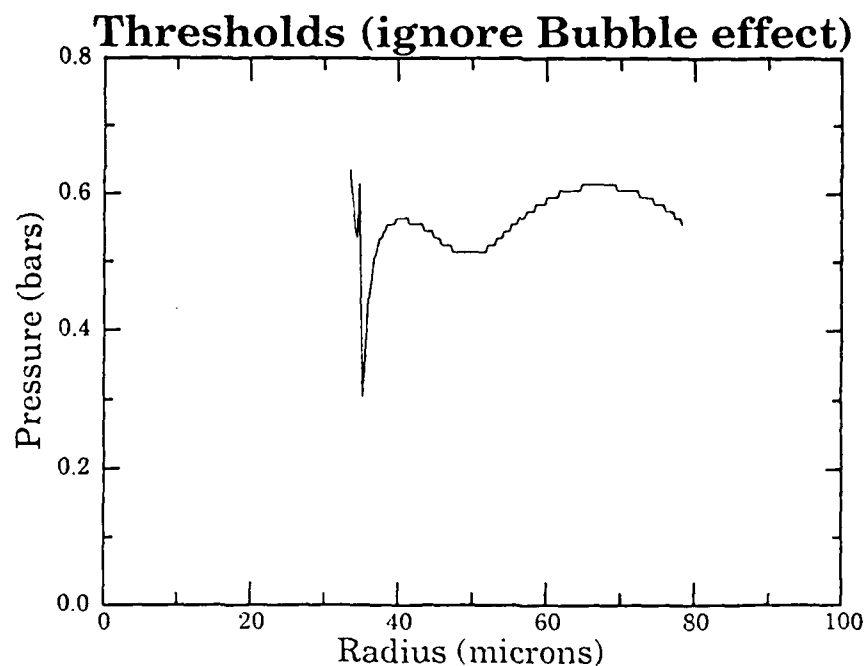


Figure 3: Analytical thresholds ignoring "bubble" feature for $n=2$ and $j=1$.

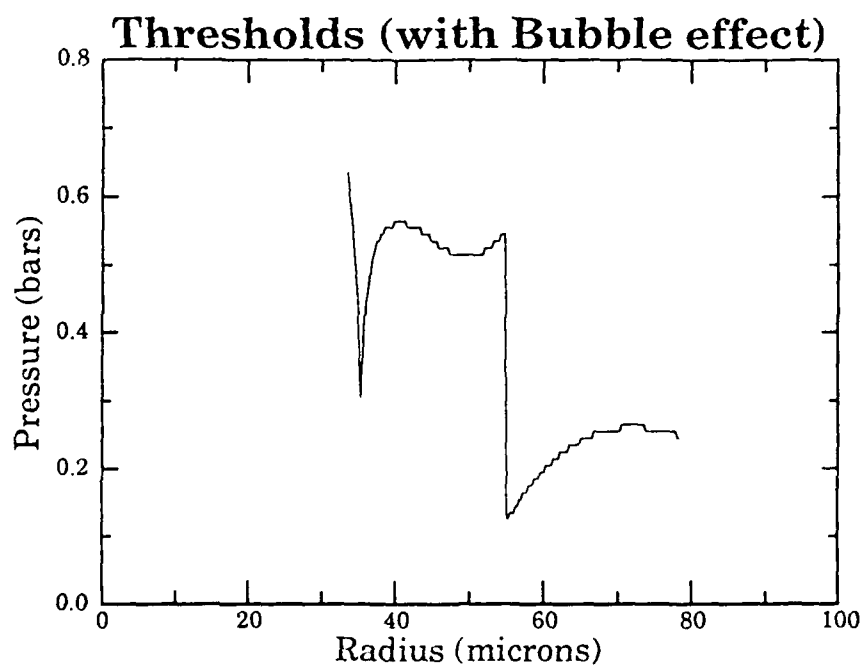


Figure 4: Analytical thresholds for the lowest pressure threshold for $n=2$ and $j=1$.

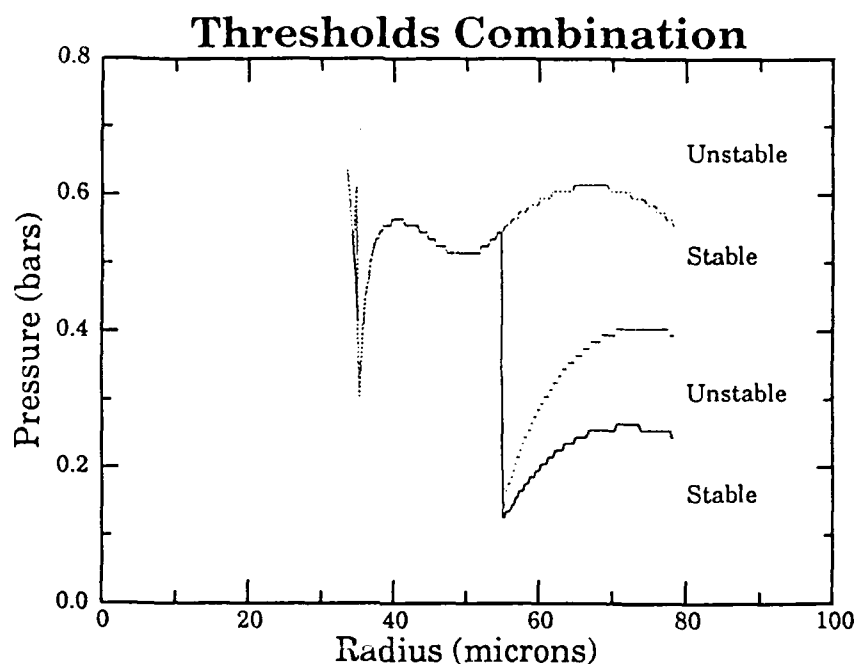


Figure 5: Combination of figures 3 and 4.

Figure 4 shows the minimum thresholds when bubbling is included. A combination of Figs. 3 and 4 is shown in Fig. 5. Notice that theoretically, there should be regions in which a bubble is radially stable at different pressures. Experimentally, this did not make sense. In effect, using this result, it would mean that if we started with a bubble of a particular radius in the bubbling region at a pressure below the first threshold, and then increased the pressure, the bubble should exhibit shape oscillations, and then as the pressure was increased through the unstable region, the bubble should become radially stable again. This experiment was tried repeatedly with no success. This seeming contradiction between theory and experiment might also lead one to suspect that there are "forbidden" regions in the pressure vs. radius curves. These would be regions which

would not contradict the experimental observations but simply that the particular mode was not possible and that another mode would engage keeping the bubble in a shape oscillation. It was also interesting to note that even the numerical integration of the original equations of motion, treated in the next chapter, showed similar (although not identical) behavior as the analytical solution.

If we assume that the radial instability threshold is the lowest pressure which satisfies Eq. (3.48), then the resulting thresholds are shown in Figs. 6, 7, and 8.

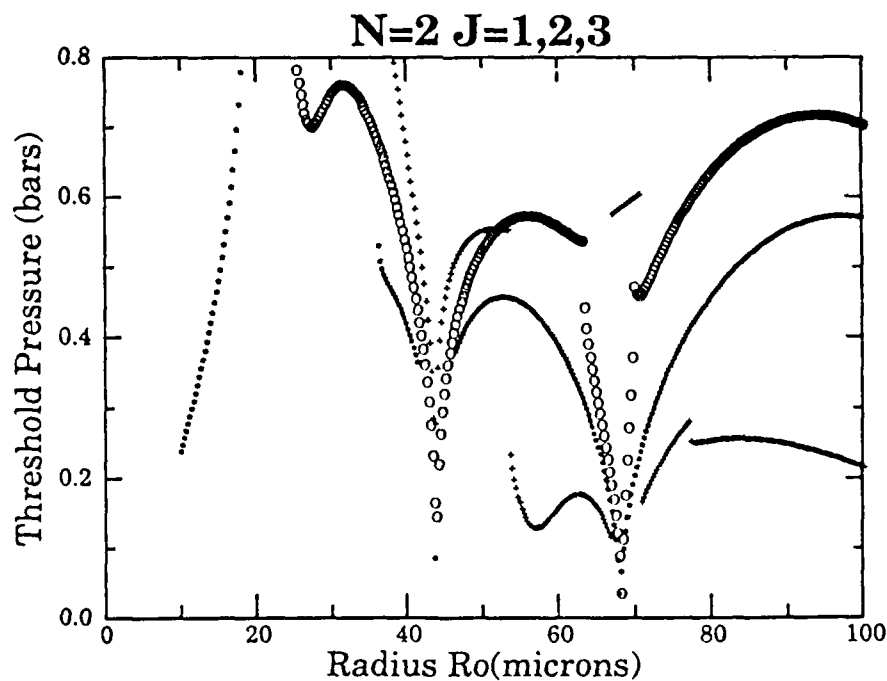


Figure 6: Analytical thresholds for $n=2$; $j=1$ (+), 2 (*), 3 (o).

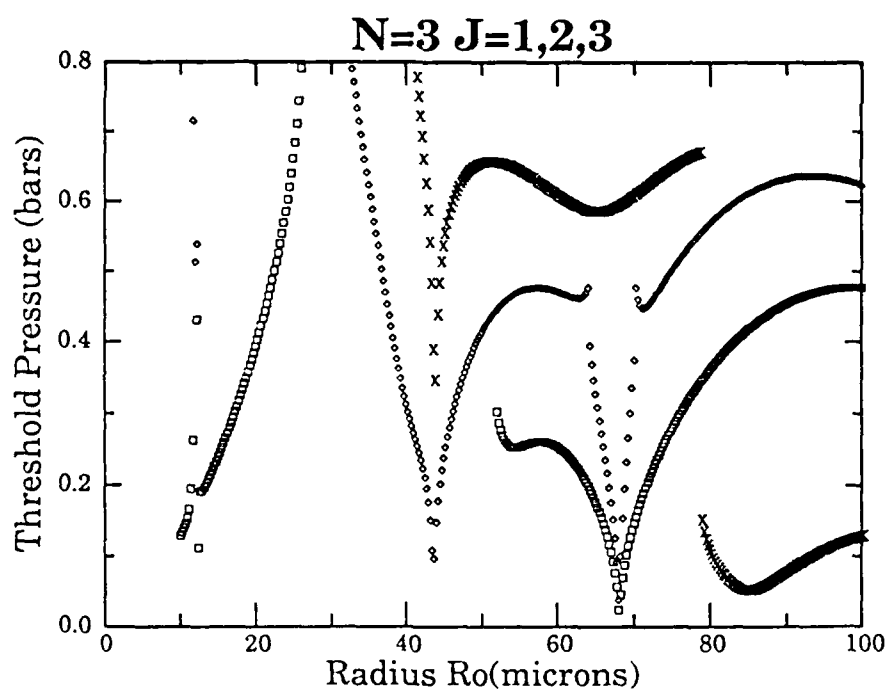


Figure 7: Analytical thresholds for $n=3$; $j=1$ (x), 2 ($\square$), 3 ($\diamond$).

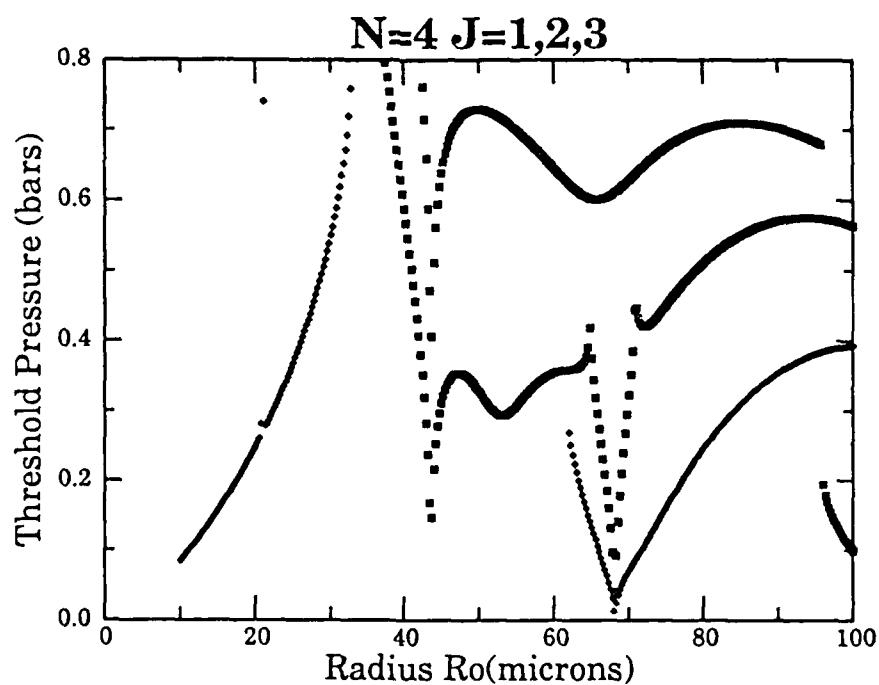


Figure 8: Analytical thresholds for $n=4$; $j=1$ ($\square$), 2 ($\diamond$), 3 ($\square$).

Figure 9 shows the result of combining all permutations of n and j by locating the minimum pressure threshold that would exhibit some sort of shape instability.

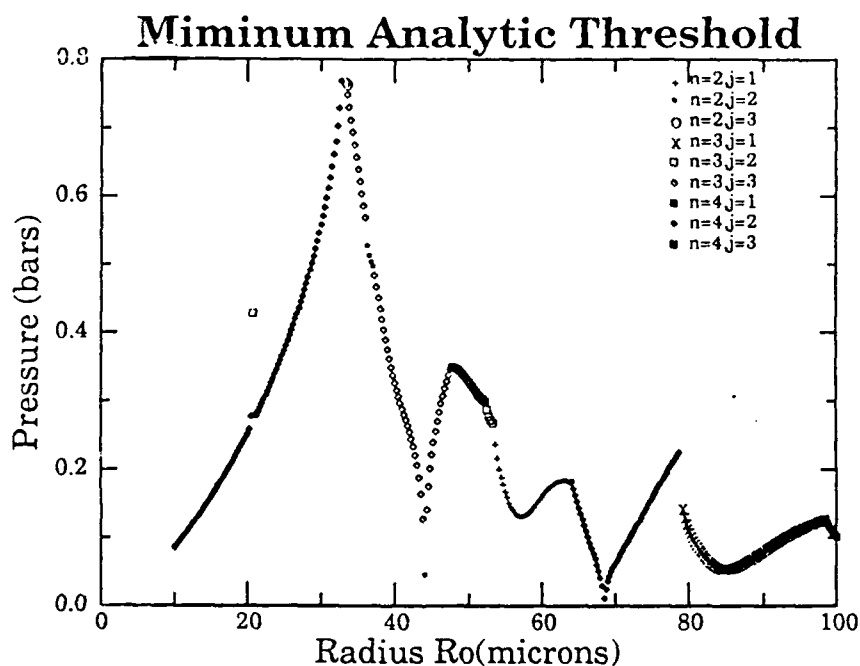


Figure 9: Combination of figures 6, 7, and 8 keeping only the lowest thresholds.

By increasing precision in computing these equations and by using newer numerical computing techniques, it was found that while the current results agree in general with the work done by Eller and Crum, there are regions of differences. It will be shown later that for small bubbles (i.e., $R_0 < 40\mu$), their theory does not predict what happens experimentally (although it is in good qualitative agreement with bubbles greater than

about 40μ). A possible reason for this discrepancy is that Eller and Crum ignored the surface tension term $2\sigma/R$ in Eq. (3.9). For large values of R , this term is negligible, but for smaller radii it cannot be ignored. The polytropic exponent is another parameter in which discrepancies could occur. Figure 10 demonstrates what happens when adiabatic or isothermal assumptions are made.

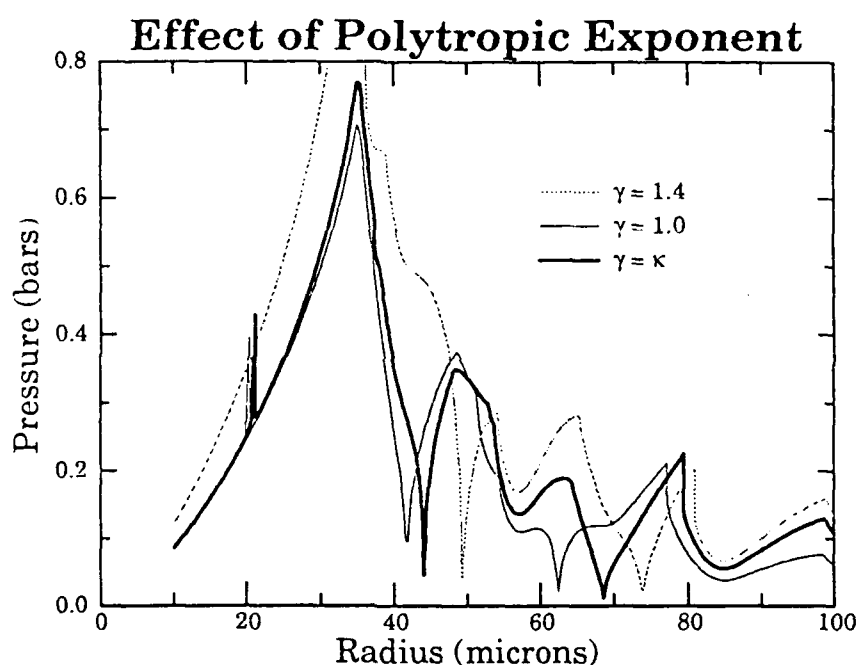


Figure 10: Comparison of analytical thresholds between polytropic exponents of $\kappa = 1.0$ (isothermal), 1.4 (adiabatic), and calculated from equation (3.3).

It is seen that the calculated polytropic exponent gives results somewhere in between adiabatic and isothermal with the calculated value tending toward one or the other at the limits of the bubble's equilibrium radius. In the next chapter, a direct integration technique will be used which does not

ignore either the surface tension or the viscosity terms and uses a calculated polytropic exponent for describing the pressure inside the bubble.

Numerical Solution

Introduction

In the previous chapter, an analytical solution to the coupled equations of motion of a driven, pulsating air bubble in water was derived. An alternate solution utilizing a direct integration technique will be presented in this chapter. The equations of motion (Eqs. (2.60) and (2.61)) are

$$R''R + \frac{3}{2}R'^2 = \frac{1}{\rho} \left[P_i - P_\infty - \frac{2\sigma}{R} - 4\mu \frac{R'}{R} \right] , \quad (4.1)$$

$$a'' + \left[\frac{3R'}{R} - 2(n-1)(n+1)(n+2) \frac{\nu}{R^2} \right] a' + \\ (n-1) \left[\frac{-R''}{R} + 2(n+1)(n+2) \nu \frac{R'}{R^3} + (n+1)(n+2) \frac{\sigma}{\rho R^3} \right] a = 0 , \quad (4.2)$$

where $R=R(t)$ is the bubble radius, ρ is the density of water, P_i is the pressure inside the bubble, P_∞ is the pressure in the water, σ is the surface tension of water, μ is the viscosity of water, $a=a(t)$ is the amplitude of the shape distortion, $\nu = \mu/\rho$, and n is the degree of the spherical harmonic.

Numerical Integration

Numerical integration of these coupled equations, derived earlier, proved to be extremely time consuming. A computer program (SOE, Shape Oscillation Equation solver) was developed to calculate the bubble motion (see Appendix D). Basically, SOE solved Eqs. (4.1) and (4.2) directly by utilizing an IMSL (IMSL, 1987) integration subroutine using Gear's "backward differentiation" technique for $R(t)$ and $a(t)$ for a particular value of n . To solve Eq. (4.1) a polytropic approximation was made for P_i :

$$P_i = P_o \left(\frac{R_o}{R} \right)^{3\kappa}, \quad (4.3)$$

where R_o is the equilibrium bubble radius, $R=R(t)$ is the instantaneous bubble radius, and

$$P_o = P_{amb} + \frac{2\sigma}{R}, \quad (4.4)$$

with P_{amb} being the ambient pressure in the liquid. The value of κ , or the polytropic exponent, has been examined by several workers (Eller (1970), Crum (1983), and Prosperetti (1977)). Their formulations are essentially equivalent and κ can be expressed by:

$$\kappa = \gamma(1+d_{th}^2)^{-1} \left[1 + 3 \frac{(\gamma-1)}{X} \left(\frac{\sinh X - \sin X}{\cosh X - \cos X} \right) \right]^{-1}, \quad (4.5)$$

with the following definitions

$$X = R_0(2\omega/D_1)^{1/2}, \quad (4.6)$$

$$D_1 = \frac{K_1}{\rho_1 c_{p1}}, \quad (4.7)$$

$$d_{th} = 3(\gamma-1) \left[\frac{X(\sinh X + \sin X) - 2(\cosh X - \cos X)}{X^2(\cosh X - \cos X) + 3(\gamma-1)X(\sinh X - \sin X)} \right], \quad (4.8)$$

where γ is the ratio of specific heats of the gas, $\omega=2\pi f$ is the angular driving frequency, K_1 is the thermal conductivity of the gas in the bubble, ρ_1 is the gas density, and c_{p1} is the specific heat at constant pressure of the gas. Next, we will drive the bubble so that P_∞ is given by:

$$P_\infty = P_{amb}(1 - P_a \sin(\omega t)), \quad (4.9)$$

where P_a is the unitless amplitude of the driving pressure. One can write Eqs. (4.1) and (4.2), using the substitution $R=R_0x$, as

$$x'' = -\frac{3x'^2}{2x} + \frac{1}{\rho R_0^2 x} \left[P_0 x^{-3\gamma} - P_{amb}(1 - P_a \sin(\omega t)) - \frac{2\sigma}{R_0 x} - \frac{4\mu x'}{x} \right], \quad (4.10)$$

$$a'' = -\left[\frac{3x'}{x} + 2(n+2)(2n+1) \frac{\mu}{\rho(R_0 x)^2} \right] a' + (n-1) \left[\frac{x''}{x} - \frac{n+2}{\rho(xR_0)^3} ((n+1)\sigma + 2\mu x'R_0) \right] a. \quad (4.11)$$

Solutions

The solution to the above coupled equations will give $a(t)$ for each value of n .

The other parameters are given by:

Ambient pressure	P_o	1.0 bar
Air density	ρ_g	0.001204 g/cm ³
Ambient temperature	T	293 K
Driving frequency	f	$\omega/2\pi = 22.22$ kHz
Gas constant	R_g	8.314×10^7 dyn/mol K
Kinematic viscosity	μ	0.0091 cm ² /sec
Liquid density	ρ	1.0 g/cm ³
Molecular weight of air	M	28.964 g/mol
Specific heats of air		
at constant volume	C_{vg}	20.8×10^7 dyn/mol K
at constant pressure	C_{pg}	2.91×10^8 dyn/mol K
Speed of sound in water	c	1.5×10^5 cm/s
Surface tension	σ	72.5 dyn/cm
Thermal conductivity of air	K_1	2.5×10^3 dyn/(sec cm K)

The procedure that SOE followed was to calculate Eq. (4.10) until the $R(t)$'s reached a steady-state, periodic solution or 2000 driving cycles, whichever came first. The criterion for a steady-state radial oscillation was defined to be no more than 0.001% deviation in the radius and no more than 0.001% deviation in the bubble wall velocity over three driving cycles. (Actually, the velocity criterion was stringent enough so that the actual deviation in the

radius was never more than 0.00001%.) Applying the steady-state criterion over three cycles allowed for detection of steady-state solutions of period one or period two (i.e., solutions which showed cyclic behavior every or every other driving period). If 2000 driving cycles were calculated before the stability criterion was met, then the solution was discarded. A steady-state solution ($t=t_s$) corresponded to a spherically symmetric bubble pulsation. The $R(t)$'s leading up to steady-state were considered transient solutions.

Next, the bubble was given a small perturbation. This meant that the SOE program now solved both Eqs. (4.10) and (4.11). To produce the initial perturbation, $a(t=0)$ was set to 1.0. It is interesting to note that examination of Eq. (4.11) shows that it is independent of the initial value of $a(t=0)$. The value of 1.0 was chosen for convenience. Next, the system of equations was allowed to run from $t = t_s$ to $t = t_s + 20$ driving cycles. SOE then generated statistical and graphical information. Examples of the graphical output is shown in the figures in Appendix C. At this point, the reader may proceed directly to the "thresholds" section on page 59 without loss of continuity. The following descriptions of the numerical outputs are included here for completeness and to demonstrate several interesting numerical features.

The title of each graph is in the following format: SOE_###_###, where the first set of numbers represents the amplitude of the driving pressure in 1/1000 bars, (i.e., 005 means 0.005 bar, and 100 means 0.1 bar). The second

set of numbers represents the bubble radius in microns (i.e., 010 means 10 microns).

The top graph in Figs. C1-C18 represents the driving pressure amplitude in Pascals. The x-axis displays time represented in number of periods of the driving pressure. The minimum x value is t_s , or the number of periods required to attain steady-state oscillation. The second graph represents the solution to the radial Eq. (4.10) (the spherically symmetric term). The periodicity of the volume pulsation can be read from this graph. The third graph represents the spherical deviation coefficient ($a_n(t)$) terms. Recall that the bubble's surface is represented by $r_s = R(t) + \sum_{n=2}^{\infty} a_n(t) Y_n(\theta)$. Thus, the third graph is the n^{th} coefficient of the n^{th} spherical harmonic term. The noteworthy feature of this graph is the increase, decrease, or stability of the $a(t)$'s as a function of time. If the values increase, then the shape oscillation is unstable, meaning that any radial instability will continue to grow in amplitude over time. If the values decrease, then the shape oscillations will damp out in time and the bubble will revert back to spherical pulsations. If the $a(t)$ solutions are stable, then the threshold for radial instability has been found. Actually achieving a stable oscillation proved difficult; as a result the theoretical thresholds were selected by determining the transition point from stability to instability. As with the radial plot, the frequency or periodicity information can be read from the plot.

The bottom two graphs represent stroboscopic Poincaré plots (Cook, 1986). The left one represents the bubble wall velocity vs. bubble radius where each point indicates the instantaneous value of the bubble velocity and radius at a specific time in each driving cycle. This graph shows what happens from time $t=0$ to $t=t_s$ leading up to the steady state oscillation. This is basically the transient information and is an indicator of the stability of the radial pulsation. If the Poincaré phase plot converges to a point, then the oscillation is stable with period one. If it converges to two points, then the oscillation is stable with period two, etc. The other stroboscopic plot represents the same information for the $a(t)$'s from time $t=t_s$ to 20 cycles later. If it converges to (0,0) then the shape oscillation damps out. If it diverges, then the shape oscillation is unstable. Otherwise the shape oscillation is stable and may be either periodic or non-periodic.

Shape Oscillation Equation solver

SOE was run for each value of $n = 2, 3$, and 4 over a range of initial bubble radii and driving pressures. The values ranged from $R=10$ to 100 microns and $P_a = 0.005$ to 0.7 bar. In order to span the space covered by this range, over 5000 graphs were output by SOE over a period of 3 weeks. The collection of graphs compiled in Appendix C represent some typical outputs from SOE.

Figure C1 shows a typical radially stable bubble. This is a 10μ bubble driven at 0.005 bar. Note that this pressure and radius can be read from the title SOE_005_010. The solution reached stability in just 10 driving cycles and when the bubble was perturbed, it damped out in less than a cycle. Figure C2 shows a more typical shape distortion response. This is a 45μ bubble driven at 0.005 bar. Here the shape oscillation damps out in ten driving cycles. Figure C3 shows a 95μ bubble driven at 0.005 bar. Here we see that this combination of pressure and bubble radius is closer to the instability threshold as evidenced by the protracted decay time of the perturbation. Also, the Poincaré phase plots of the radial transient motion show different behavior on each plot even though visually the radial motion is basically the same and any shape distortion is damped out. Figures C1, C2 and C3 show bubbles that are driven at a very low pressure amplitude.

If the driving pressure is increased to 0.1 bar then more interesting bubble behavior is observed. Figure C4 shows a 70μ bubble driven at 0.1 bar. A second harmonic is now observed in the radial oscillation. The shape distortion graph still shows a damping out effect while also showing other transient effects. Figures C5, C6, C7, and C8 show a sequence of plots in which the driving pressure is 0.15 bar. Solutions for bubbles of sizes 54μ , 56μ , 58μ , and 60μ are shown. At 54μ the shape distortion amplitude slowly damps out (Fig. C5). At 56μ (Fig. C6) the shape oscillation is basically stable (graphically the solution looks stable, but actually it is slightly decreasing) with a period of twice the driving frequency. The periodicity (period two)

can also be seen from the accompanying Poincaré graph. For the 58μ bubble (Fig. C7), it can be seen that the shape oscillation amplitude is growing. The appearance of the second harmonic in the radial response is now more pronounced. For the 60μ bubble (Fig. C8) (still at the same driving pressure) the shape distortion amplitude is again damping out. Thus, it was seen that bubbles in this size range went through a threshold with the threshold point being about 57μ at 0.15 bar.

This "threshold crossing" feature can also occur more than once at the same driving pressure. Figures C9-C15 show a sequence of plots representing multiple crossings of the instability thresholds in radius for the same pressure. Here the pressure is kept constant at 0.175 bar. The bubble radii range from 54μ to 66μ . At 54μ (Fig. C9) the shape distortion solution is decreasing in time, at 56μ (Fig. C10) the distortion is increasing in time indicating a threshold crossing near 55μ . At 58μ (Fig. C11) and at 60μ (Fig. C12) the shape oscillation amplitudes are increasing while at 62μ (Fig. C13) the shape distortion initially increases and then damps out. Thus, a second threshold crossing near 61μ has occurred. For a 64μ (Fig. C14) bubble the shape solution again initially increases and then damps out. For a 66μ (Fig. C15) bubble, the solution is again increasing indicating a threshold crossing near 65μ . This "multiple crossing" effect is prevalent throughout the pressure and radius values used.

The preceding figures showed only the first 20 cycles of the solution after steady-state. Actually, the calculations were continued for another 80 cycles for a total of 100 cycles of the driving pressure. Figure C16 shows an example of all 100 cycles. Since it was difficult to see much information in this type of graph, 20 cycles was deemed sufficient to determine graphically the bubble's behavior.

Increasing the pressure to 0.2 bar showed more transient behavior in both the radial and the shape distortion solutions. In fact, some combinations of pressure and radius showed an extremely large (10^{21}) increase in the shape oscillation amplitude. These are obviously non-physical since once the bubble begins large amplitude shape distortions the bubble will break up. The theory does not include any breakup information.

Figure C17 shows an interesting Poincaré plot. The steady-state radial oscillation is a typical oscillation with a strong second harmonic component, but the transient information leading up to the state-state solution has an interesting behavior. Analysis of the transient behavior of the SOE plots, while an interesting project, is beyond the scope of this work.

So far, all of the radial responses have been period one (i.e., they repeat once every cycle of the driving frequency). Figure C18 illustrates a period two response evidencing the emergence of a sub-harmonic. The period two response is easily seen in the Poincaré plot.

Thresholds

SOE generated over 5000 of the graphs outlined above. The type of shape distortions (i.e., the spherical harmonic number, n) were calculated separately for cases of $n = 2, 3$, and 4 . The graphs were then examined to obtain the thresholds for surface oscillation. These thresholds were defined as the transition point from decreasing to increasing a 's as a function of time. Figures 11, 12, and 13 show the pressure thresholds as a function of radius for $n = 2, 3$, and 4 .

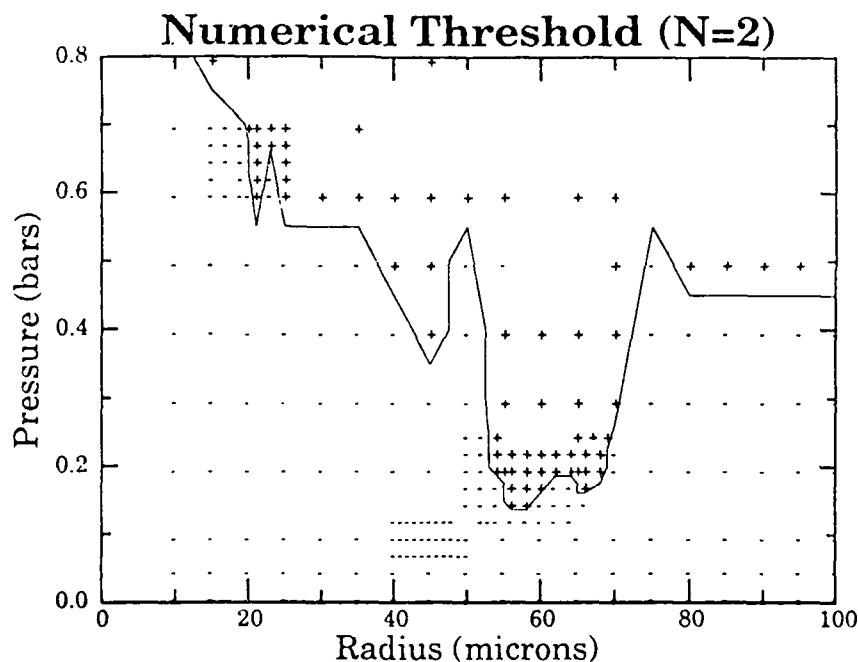
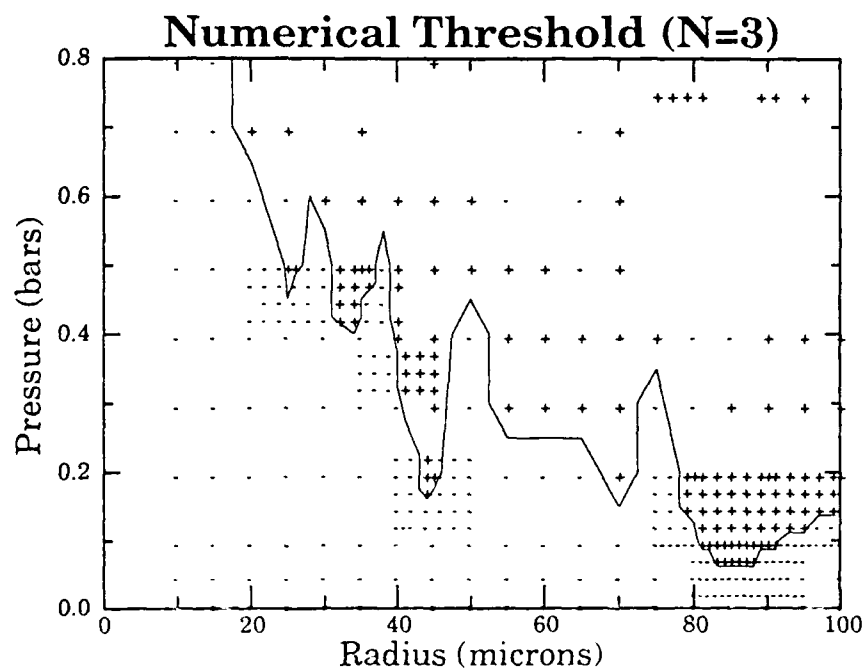
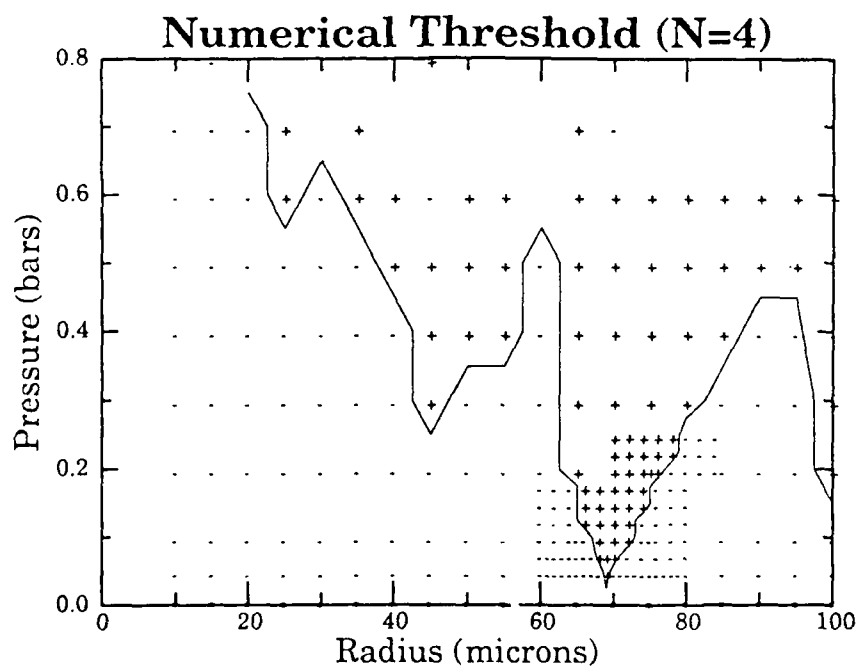


Figure 11: SOE numerical thresholds for $n=2$.

Figure 12: SOE numerical thresholds for $n=3$.Figure 13: SOE numerical thresholds for $n=4$.

The numerical threshold graphs are shown in grid form. The plus signs indicate an increasing solution to Eq. (4.11), while the negative signs indicate a decreasing solution. The points at which the solutions go from decreasing to increasing indicate the transition through the radial instability threshold. An adaptive grid approach was used to find the cross-over point from decreasing to increasing solutions. Initially, a large grid spacing was used and then by manually looking at the resulting plots from SOE, regions of the grid were blocked off and the grid spacing was narrowed. This procedure was repeated until a definite threshold curve could be generated. Figure 14 is a superposition of all the thresholds.

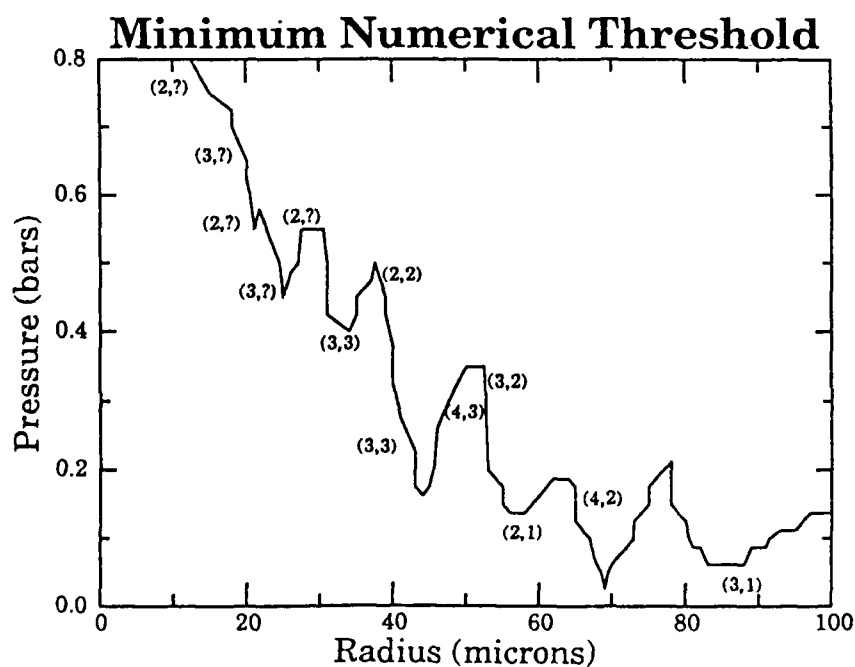


Figure 14: Minimum thresholds of figures 11, 12, and 13.

The solid line represents the minimum pressure threshold for radial instability as a function of bubble radius. The values of the mode, n , and the frequency multiplier, j , (from $e^{j\omega t/2}$) are also indicated.

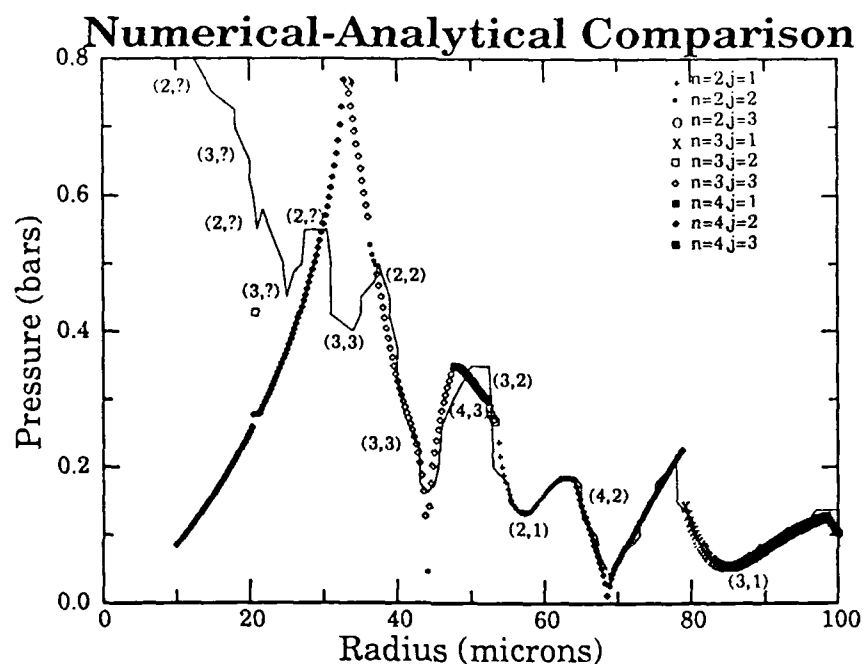


Figure 15: Overlay of figures 9 and 14. Comparison between the analytic and numerical solutions to the equations of motion.

Comparison with analytic solutions

Figure 15 shows a comparison between the analytical results in the previous section and the numerical threshold results obtained in this section. It can be seen that for bubble radii greater than 40μ , the predicted thresholds for both the analytic and numerical solutions agree quite well.

This figure demonstrates one of the important contributions of this study. It shows that by including the surface tension terms, in the equations of motion, and then numerically integrating the resulting coupled differential equations, a different threshold curve is obtained than that obtained using the analytical approach, especially for smaller bubbles.

Below is a table of the mode numbers, n , which represent the type of shape oscillation, determined by the n^{th} spherical harmonic, and the frequency multiplier j determined by $e^{j\omega t/2}$. For $j=1$ the shape oscillation will oscillate at $1/2$ the driving frequency, for $j=2$ the shape oscillation will oscillate at the driving frequency and for $j=3$, at $3/2$ the driving frequency.

Bubble Radius Range μ	Mode number n_a, n_n	Frequency j_a, j_n
97 - 100	4,4	1,1
78 - 97	3,3	1,1
63 - 78	4,4	2,2
53 - 63	2,2	1,1
51 - 53	3,3	2,2
48 - 51	4,4	3,3
36 - 48	3,3	3,3
35 - 36	2,2	2,2
33 - 35	3,3	3,3
$\Rightarrow$ 33 - 33	2,3	3,3
24 - 33	4,2	2,?
20 - 24	4,3	2,?
20 - 21	4,2	2,?
18 - 20	4,3	2,?
10 - 18	4,2	2,?

(subscripts a and n represent analytic and numeric)

Table 1: Analytical/Numerical comparison

As seen from the table, the analytical approximation agrees with the numerical solution quite well from 35μ to 100μ . For smaller radii though, the numerical threshold results are more reasonable. Unfortunately, for these smaller bubble radii, transients obscured the accurate determination of the shape oscillation frequency and are denoted by a (?). It would be expected that as the bubble radius decreases, the stability would generally increase or approach a constant value. The numerical threshold predictions show a general increase in the stability. As was stated in the analytic section, the reason for the non-physical nature of the analytic approximation could be due to the ignoring of the surface tension terms in the equations of motion.

The numerical method used Eqs. (4.1) and (4.2), were derived under the assumption of axisymmetry. In theory, a more general solution, not assuming axisymmetry, could be attempted and might be a subject for future work. Now that the theoretical, analytical and numerical bases have been outlined, the next section will detail the experimental verification of these approaches.

Mie Scattering

Theory

In this experiment, laser light was scattered from an oscillating bubble to detect radial and shape oscillations of the bubble. Scattering laser light off an air bubble in water to quantitatively measure the pulsations of bubbles from about 10-100 microns has been shown to be a very sensitive technique (Hansen 1983, and Holt 1988). The theory which describes the phenomenon of the scattering of electromagnetic waves off small particles ($ka > 10$), is known as "Mie Theory", named after Gustav Mie. The scattering of light by a spherical object has received considerable attention (Mie 1908, Debye 1909, Stratton 1941, Born and Wolfe 1959). This theory deals with the scattering of a plane electromagnetic wave by an isotropic, homogeneous sphere of arbitrary size. The scattered intensity as a function of position is the quantity of interest. In the far field the scattering amplitude functions are given by:

$$S_1 = \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} (a_n \pi_n(\cos\theta) + b_n \tau_n(\cos\theta)) , \quad (5.1)$$

$$S_2 = \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} (a_n \tau_n(\cos\theta) + b_n \pi_n(\cos\theta)) . \quad (5.2)$$

The interested reader is directed to Appendix B for a derivation of these amplitude functions. The functions π_n and τ_n are given by:

$$\pi_n(\cos\theta) = \frac{P_n^{(1)}(\cos\theta)}{\sin\theta}, \quad (5.3)$$

$$\tau_n(\cos\theta) = \frac{d}{d\theta} P_n^{(1)}(\cos\theta), \quad (5.4)$$

where $P_n^{(1)}$ are the associated Legendre polynomials. The scattering coefficients are given by:

$$a_n = \frac{\psi_n(\alpha)\psi_n'(\beta) - m\psi_n(\beta)\psi_n'(\alpha)}{\zeta_n(\alpha)\psi_n'(\beta) - m\psi_n(\beta)\zeta_n'(\alpha)}, \quad (5.5)$$

$$b_n = \frac{m\psi_n(\alpha)\psi_n'(\beta) - \psi_n(\beta)\psi_n'(\alpha)}{m\zeta_n(\alpha)\psi_n'(\beta) - \psi_n(\beta)\zeta_n'(\alpha)}, \quad (5.6)$$

where $m = k_1/k_2$, $\alpha = k_2a = 2\pi m_2a/\lambda_0$, $\beta = k_1a = m\alpha$, k is the wave number, a is the bubble radius, and λ_0 is the wavelength of the incident light in vacuum. Subscripts 1 and 2 refer to the bubble interior and surrounding medium respectively. The functions ψ and ζ are given by:

$$\psi_n(kr) = (\pi kr/2)^{1/2} J_{n+1/2}(kr), \quad (5.7)$$

$$\zeta_n(kr) = (\pi kr/2)^{1/2} H_{n+1/2}^{(2)}(kr), \quad (5.8)$$

where $J_{n+1/2}(kr)$ is the half-integral order Bessel function and $H_{n+1/2}^{(2)}(kr)$ is the half-integral order Hankel function.

The intensity of the scattered, polarized light in the θ and ϕ directions is:

$$I_{\theta} = \frac{\lambda^2}{4\pi^2 r^2} |S_2|^2 \cos^2\phi = \frac{\lambda^2}{4\pi^2 r^2} I_2 \cos^2\phi , \quad (5.9)$$

$$I_{\phi} = \frac{\lambda^2}{4\pi^2 r^2} |S_1|^2 \sin^2\phi = \frac{\lambda^2}{4\pi^2 r^2} I_1 \sin^2\phi , \quad (5.10)$$

where λ is the wavelength in the medium. Note that θ and ϕ represent the perpendicular and parallel components, respectively, to the scattering plane. The scattering plane contains both the scattering and the incident waves.

Calculation

In the experiment, the incident light was polarized with the electric field vector parallel to the $\phi=90^\circ$ plane. The detecting equipment was also in that plane so that the only component detected was I_{ϕ} , where

$$I_{\text{exp}} = I_{\phi} = \frac{\lambda^2}{4\pi^2 r^2} I_1 \sin^2\phi = \frac{\lambda^2}{4\pi^2 r^2} I_1 . \quad (5.11)$$

Since all quantities except I_1 were kept constant, the calculation was performed to render I_1 as a function of scattering angle (θ) and as a function of the bubble radius (r). To calculate these quantities a modified version of the computer program developed by Wiscombe (MIEV1) was used. A listing of the modified program (MIENOS) is given in Appendix D. The programs were initially run on a MicroVAX II, with runtimes on the order of a few minutes for simple cases to over ten hours for integration cases. The programs were then transferred to a CYBER 205 which improved performance considerably. Execution times on the CYBER ranged from one to ten minutes. This allowed for the production of a large number of data sets in a relatively short time.

Scattering intensity as a function of angle

The first set of data concerned the scattering intensity as a function of angle with a constant bubble radius. Both I_θ and I_ϕ were calculated. The scattering angle was from 0 (forward) to 180 (backward) degrees. The bubble radii ranged from 10 to 100 microns. Figures C19 and C20 (found in Appendix C) illustrate the scattering. A representative figure is shown here for a bubble of a 50μ radius.

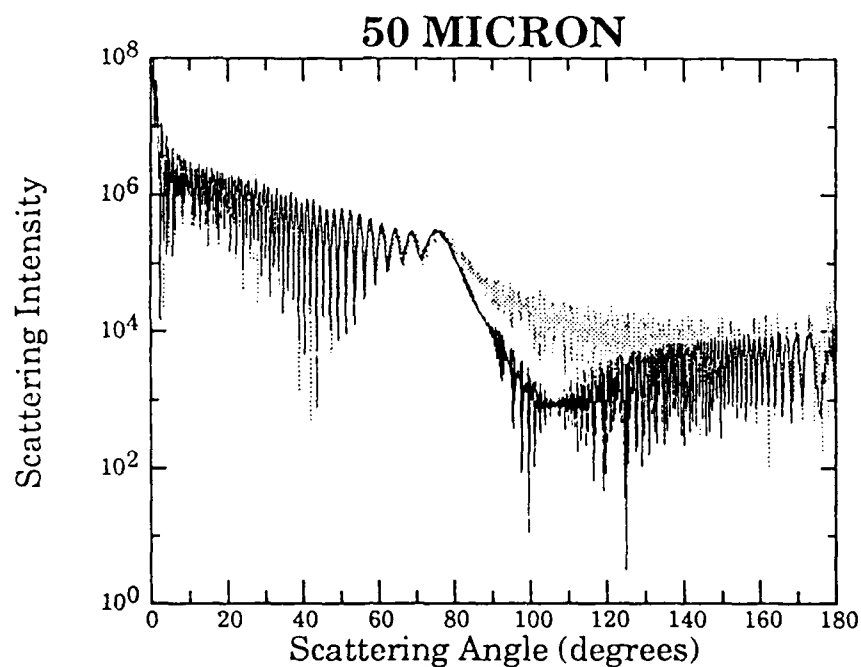


Figure 16: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of scattering angle for a 50μ bubble.

In Fig. 16, the dotted lines represent I_1 and the solid lines I_2 where

$$I_1 = \log_{10} \left(\frac{4\pi^2 r^2}{\lambda^2} I_\phi \right) \quad (5.12)$$

$$I_2 = \log_{10} \left(\frac{4\pi^2 r^2}{\lambda^2} I_\theta \right). \quad (5.13)$$

In general, Figs. C19 and C20 show that I_1 is normally slightly greater than I_2 . Also, from about 75 to 130 degrees, I_1 is much greater than I_2 . The general structure of the curves is essentially the same. Each begins with a very large forward scatter followed by a general decline to a small peak around 75 to 80 degrees, and in the case of I_1 , a leveling out up to 180

degrees, while for I_2 , a sharp decline to about 100 degrees followed by a general rise up to 180 degrees.

With the experimental setup used in this study, these graphs should be reproducible. However in practice, the detection apparatus always has an acceptance angle that must be taken into consideration. In this experiment, an acceptance angle of about 4 degrees was used. To compensate for this angle of acceptance, the scattering programs were rerun and integrated over 4 degrees. Figures C21 and C22 show the results. Figure 17 is a representative graph of a 50 μ bubble.

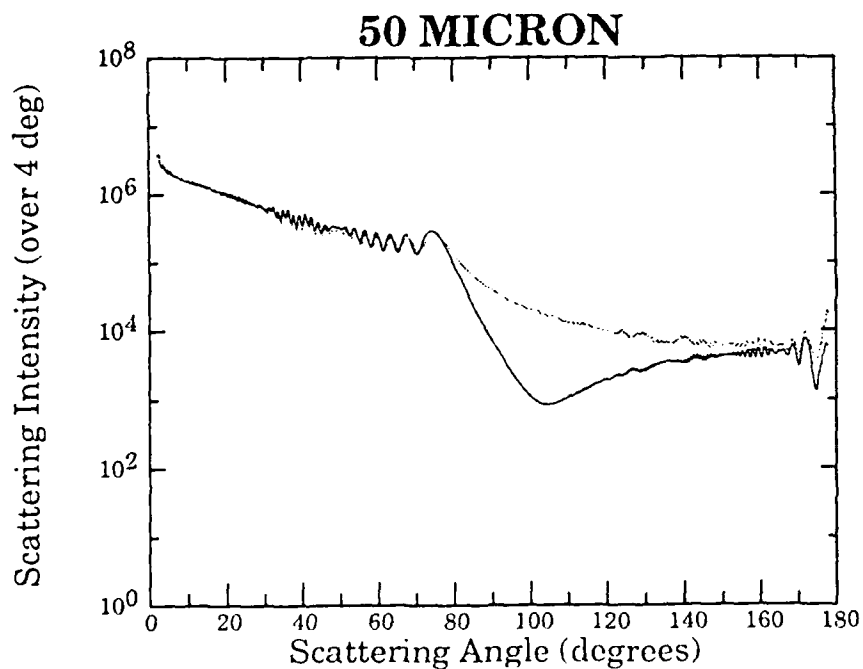


Figure 17: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a solid angle of 4 degrees as a function of scattering angle for a 50 μ bubble.

Nearly all of the fine structure exhibited in Figs. C19 and C20 has been eliminated. Again, I_1 is generally larger than I_2 . Figure 18 is a composite of I_2 , and Fig. 19 a composite of I_1 , for a range of bubble radii.

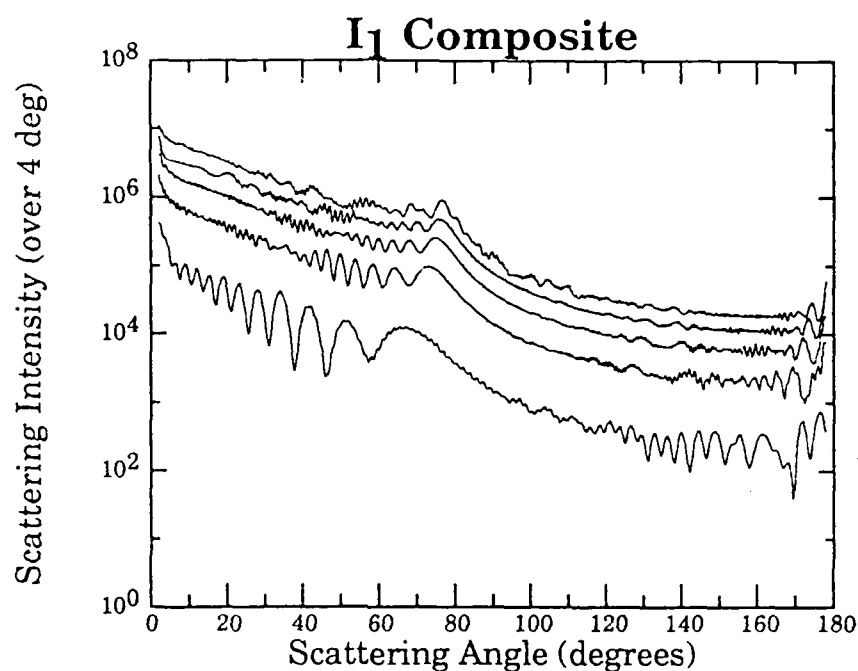


Figure 18: Composite I_1 angular Mie scattering result (over a 4 degree solid angle) for bubble radii: 10μ (lower curve), 30μ , 50μ , 70μ , and 90μ (upper curve).

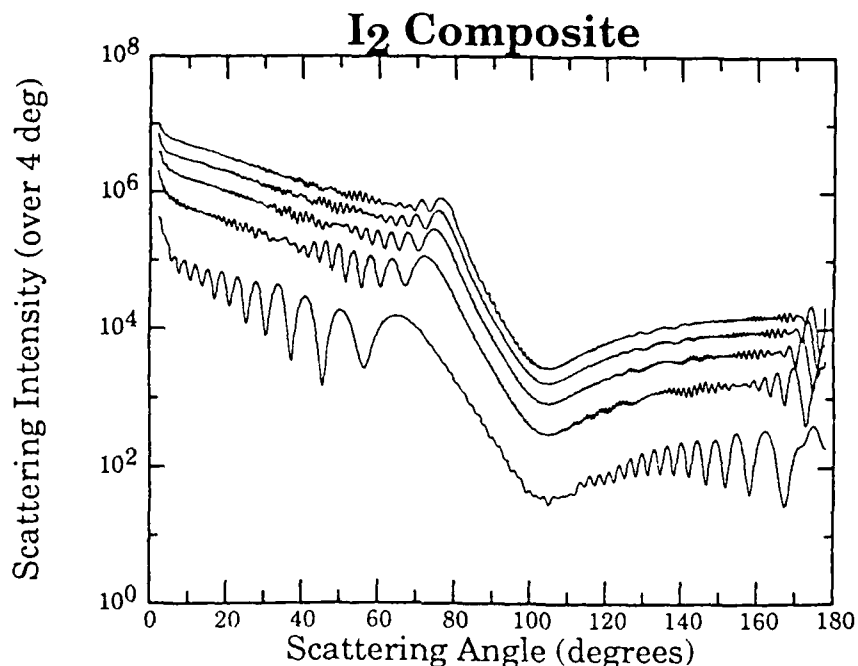


Figure 19: Composite I_2 angular Mie scattering result (over a 4 degree solid angle) for bubble radii: 10μ (lower curve), 30μ , 50μ , 70μ , and 90μ (upper curve).

It is apparent from the graphs that to maximize the detection of scattered light, the detector must be placed at an angle somewhere less than 80 degrees. In this experiment, detectors were placed at +30 degrees, -30 degrees, and 80 degrees. The detectors were sensitive enough so that their positioning was not paramount and their placement was dictated by experimental constraints, not by theoretical considerations. The choice of polarization angle was also dictated by the same considerations. The I_1 polarization was chosen since, at these angles, the detectors output higher voltages than with the I_2 polarization. To determine if these theoretical graphs matched experimental data, an experiment was performed in which a bubble of known size was levitated in a tuned cell and made to

pulsate at 22.22 kHz in a spherically symmetric oscillation. (The experimental apparatus will be described in detail in the next chapter.) The detector was then rotated around the levitation cell from an angle of 45 to 120 degrees. The output from the detector was recorded on a computer. Figures 20, 21, and 22 show the results for 50 μ , 60 μ , and 80 μ bubbles.

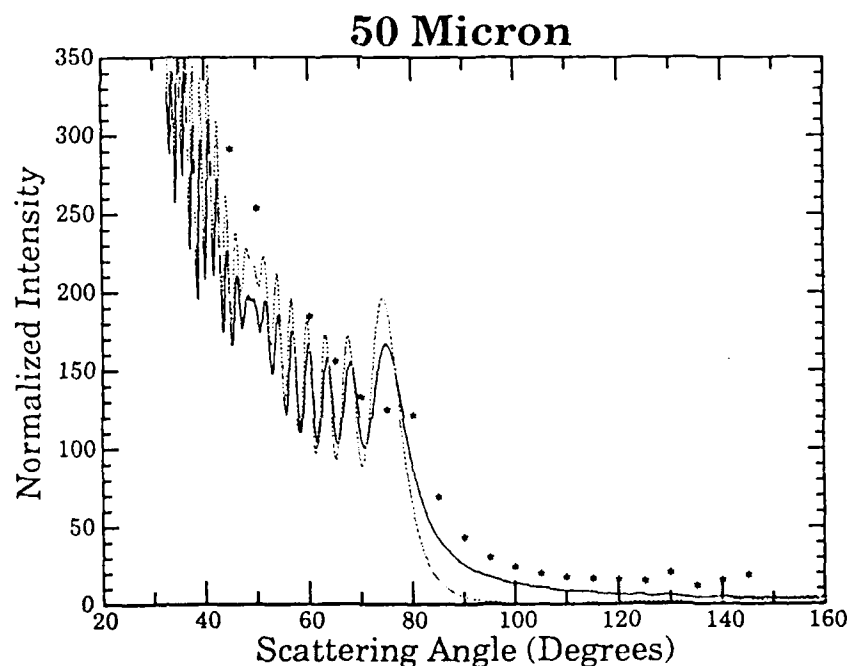


Figure 20: Experimental angular Mie scattering results (asterisks) for a 50 μ bubble, overlayed by I_1 (solid line) and I_2 (dotted line).

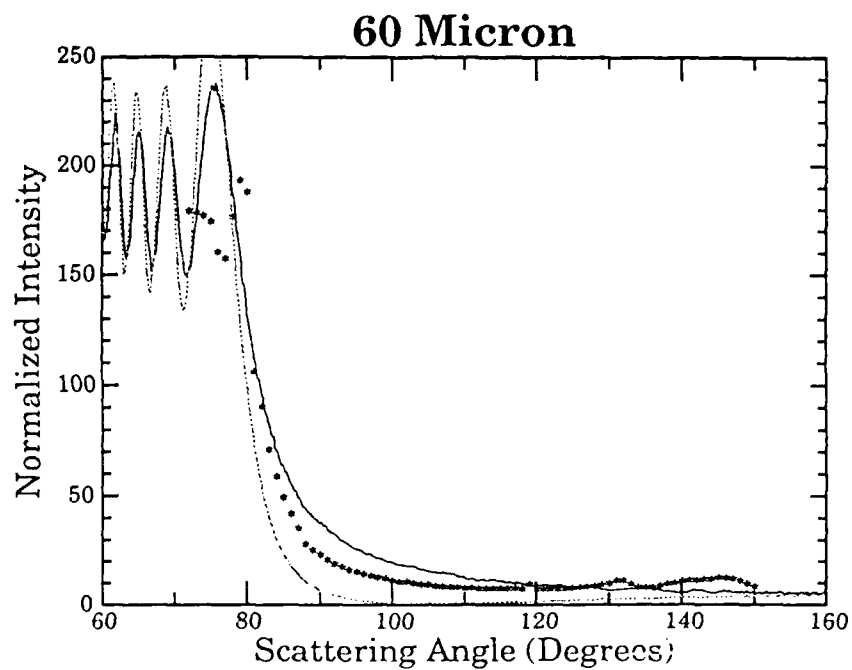


Figure 21: Experimental angular Mie scattering results (asterisks) for a 60 μ bubble, overlayed by I_1 (solid line) and I_2 (dotted line).

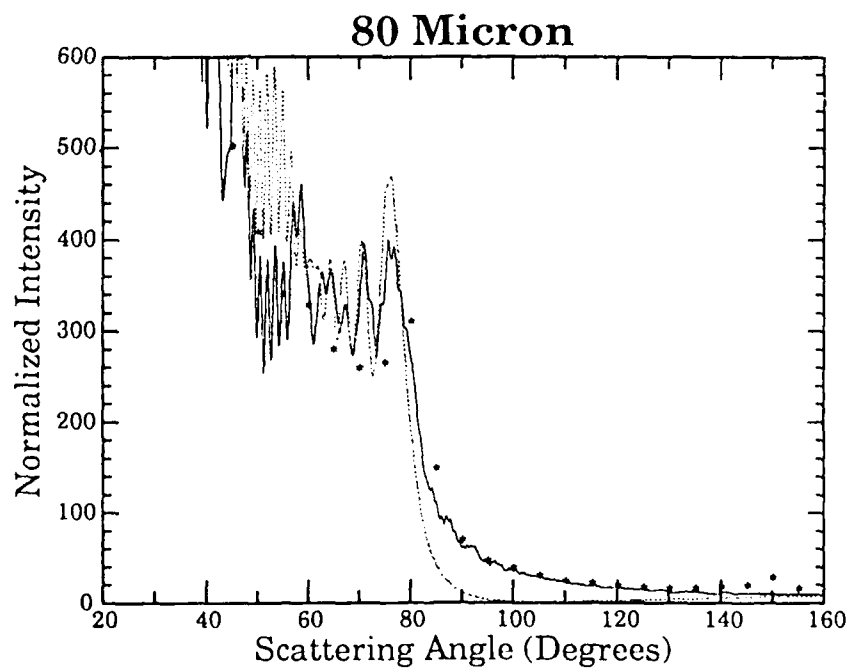


Figure 22: Experimental angular Mie scattering results (asterisks) for a 80 μ bubble, overlayed by I_1 (solid line) and I_2 (dotted line).

The asterisks represent the experimental data while the lines represent the predicted scattering amplitudes from Mie scattering theory. The solid lines indicate I_1 and the dotted lines I_2 . Both I_1 and I_2 have been normalized to compare against the experimental data. (The normalization is described in the next section.) The scattering angle is with respect to the forward scattering direction. It is clear from these graphs that the polarization is in the I_1 plane. For scattering angles above 80 degrees the prediction matches the experimental data quite well. Due to experimental constraints, angles less than 40 degrees were not accessible. For the angles between 50 and 80 degrees, the general shape of the curves are similar but specific differences are also apparent.

Scattering at a specific angle for various bubble radii

The scattering programs were then run keeping the detection angle constant and varying the bubble size. Figures C23-C27 show the scattering intensity as a function of bubble radius for detection angles of 10 to 180 degrees. Figure 23 illustrates the scattering intensity as a function of bubble radius for an 80 degree detection angle.

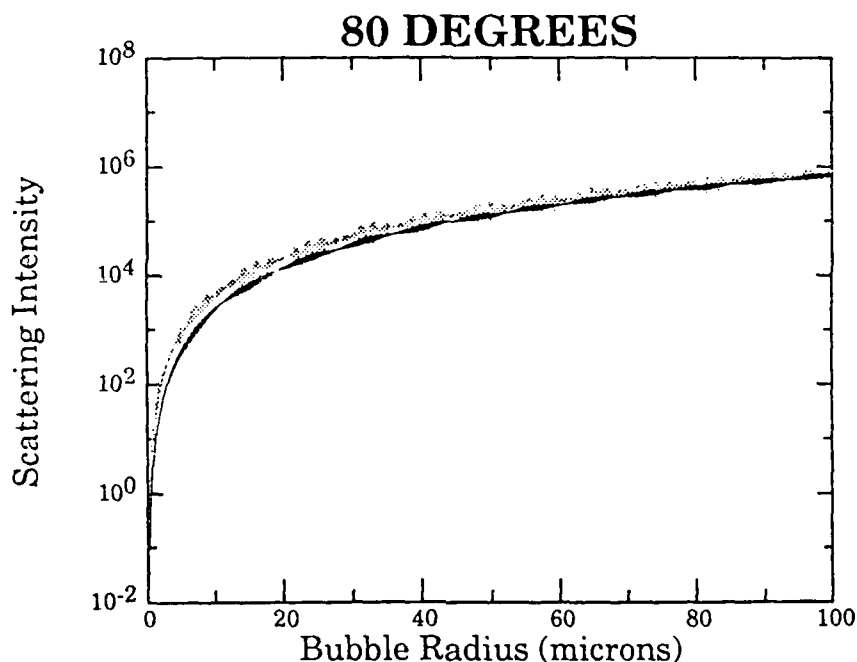


Figure 23: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of bubble radius for an 80 degree scattering angle.

Again, there is noticeable fine structure. To reduce the fine structure, with the aim of using this information for calibration purposes, the same technique of integrating over a solid angle was utilized. This was a very compute-intensive process and we were only able to obtain results in a reasonable period (about one hour) when using a Cyber 205 and fully vectorized computer code. Figures C28-C32 show the results of this integration. Figure 24 illustrates the results for an 80 degree detection angle over a 4 degree acceptance angle.

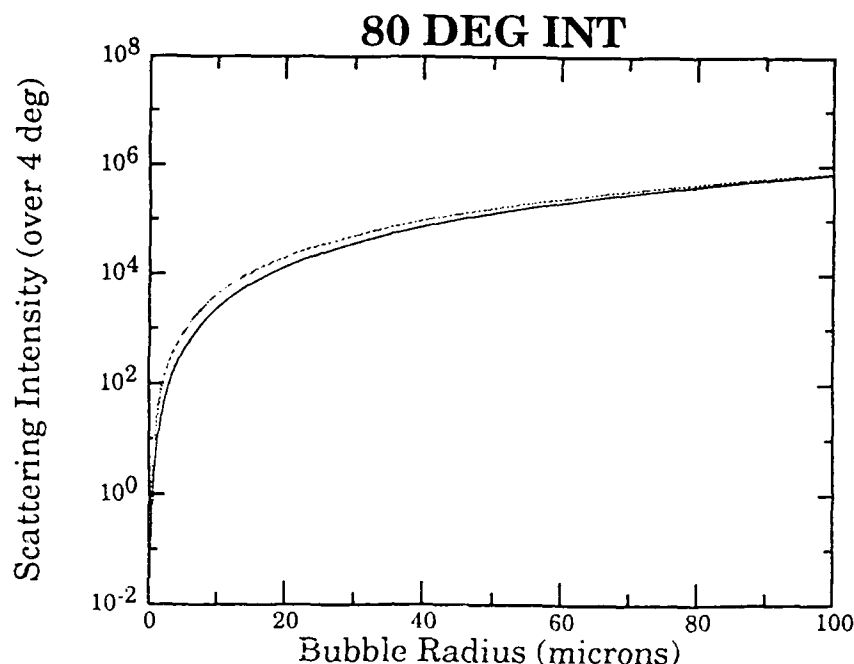


Figure 24: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a 4 degree solid angle, as a function of bubble radius for an 80 degree scattering angle.

Notice that most, but not all of the fine structure is gone. Examination of Figs. C28-C32 show that for most detection angles the scattered intensity is not single valued. The graph at 80 degrees shown in Fig. 24 is single valued over the range of bubble radii considered and as such can be used as a transfer curve to obtain the actual bubble radius given a specific intensity. This fact was the major reason for placing one of the detectors at 80 degrees.

To verify the theory, the detector was positioned at a fixed location (keeping the detection angle constant) and the bubble size was varied. A bubble was created in a levitation cell tuned to 22.22 kHz. A polarized laser beam (488

nm) was scattered off the bubble and detected by a photodiode located 80 degrees from the forward scattering angle at a distance of about 150 mm from the center of the bubble. This gave an acceptance angle of about 4 degrees. Bubbles of various radii were created and the detector output was recorded. Only those bubbles that were spherically symmetric were chosen. Bubble radii around $1/2$ and $1/3$ of the resonance radius ($R_0 = 142 \mu\text{m}$) were not chosen due to inherent non-linearities in their pulsations and their affinity for producing surface oscillations. It was also noticed that bubbles near these sizes grew very rapidly in a matter of seconds. Figure 25 shows the experimental data results of this test.

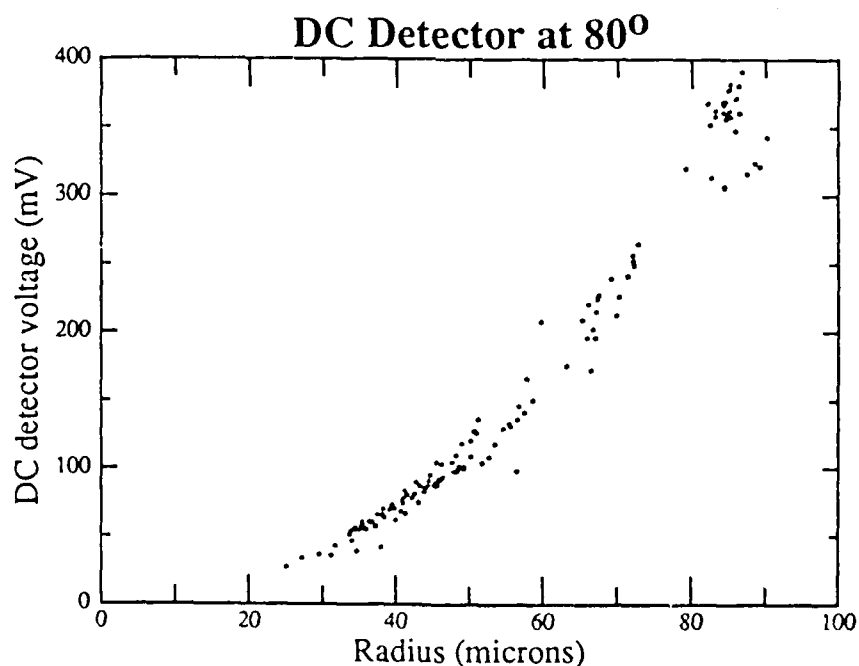


Figure 25: Experimental data from DC detector at 80 degrees as a function of bubble radius.

In order to compare the Mie scattering prediction to the experimental data, the incident intensity on the bubble is required to scale the prediction appropriately. In this study, we were unable to directly measure the laser beam intensity incident on the bubble's surface. To compensate for this a least-squares fit was made between the theoretical and experimental results to determine the multiplicative constant I_0 . Where $I_{\text{exp}} = I_{\text{mie}} / I_0$. Figure 26 shows the results of the comparison between theory and experiment.

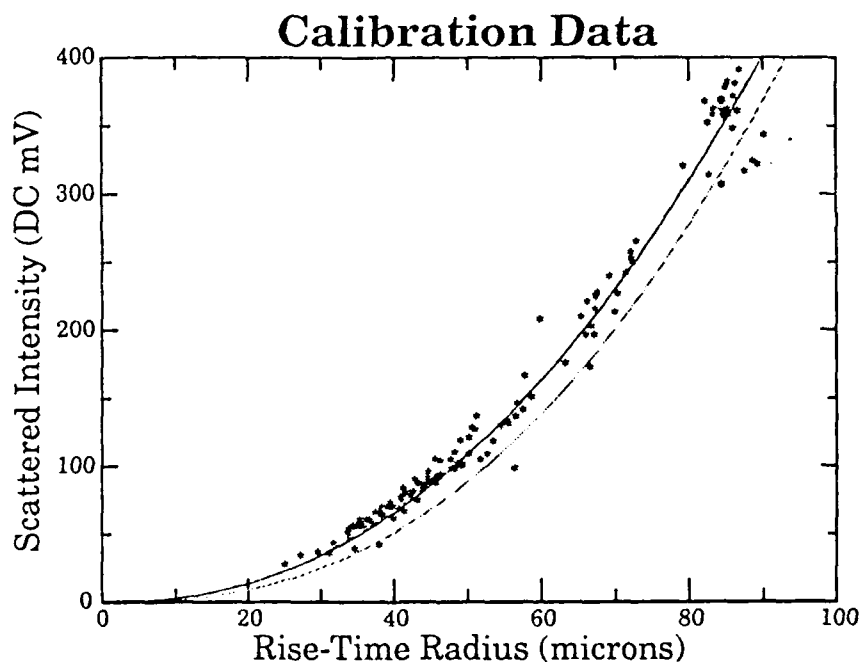


Figure 26: Experimental data from DC detector at 80 degrees as a function of bubble radius. Overlaid by the Mie scattering prediction, I_1 (solid line) and I_2 (dotted line).

It can be seen that the theoretical prediction matches the experimental data very well. The underestimation of Mie theory at smaller radii could be

explained as a systematic error in the experimental data. To test this hypothesis, the experiment was performed under a variety of conditions, but no significant differences were found. Thus, if a systematic error did occur it was not observed. However, differences between the Mie theory fit and the experimental data were about one to two microns which was within the experimental error estimate.

Experiment

Method

In the preceding chapters we presented the theoretical foundation for predicting shape oscillation thresholds, in this chapter we present the experimental results and comparisons with theory. In this experiment an acoustic levitation technique was used to position a single, stably-pulsating air bubble in water at a constant position. In order to observe the pulsation of the bubble, it was necessary to levitate the bubble in water at a specific position so that a laser beam could be scattered off the bubble for long periods of time. Since an air bubble in water experiences a buoyancy force, an opposing force is required to keep the bubble stationary. In this experiment an acoustic stationary wave was produced in a cylindrical cell. The two acoustic radiation forces acting on a bubble in this field are the so-called rigid-sphere force (Nyborg, 1967) and a force due to the compressibility of the bubble. Crum (1970) observed that the rigid-sphere terms were completely dominated by the compressibility terms for a gas bubble and could be neglected. Eller (1967) derived an expression for the compressibility terms of the average acoustic force in the z direction as

$$F_A = \frac{2\pi^2 R_n^3 P_A^2}{3P_o \lambda_z (1 - \omega^2/\omega_o^2)} \sin\left(\frac{4\pi z}{\lambda_z}\right), \quad (6.1)$$

where R_n is the equilibrium bubble radius about which it sinusoidally oscillates, P_A is the acoustic pressure amplitude at the position of the bubble along the z axis, P_o is the hydrostatic pressure, λ_z is the wavelength of the standing wave in the z direction, ω is the driving frequency, and $\omega_o^2 = 3P_o/\rho R_n^2$ is the resonance frequency with ρ being the density of water. When this force balances the buoyancy force, $F_b = (4/3)\pi R_n^3 \rho g$, then the bubble is acoustically levitated. Bubbles driven below resonance have a stable equilibrium position on the z axis just above a pressure maximum. Thus, the sound field draws the bubble toward a pressure antinode opposing the upward buoyancy force. Figuratively, this can be depicted as shown in Fig. 27. During the positive pressure cycle, the bubble's volume is at its smallest and the force due to the pressure gradient is upward. During the negative pressure cycle, the bubble's volume is at a maximum and the force is downward. Thus, the average force is basically the sum of the forces, and since the volume during the negative cycle is larger than the volume in the positive cycle, the total force points downward opposing the upward buoyancy force. Once we had a method for levitating bubbles, an experimental apparatus was designed.

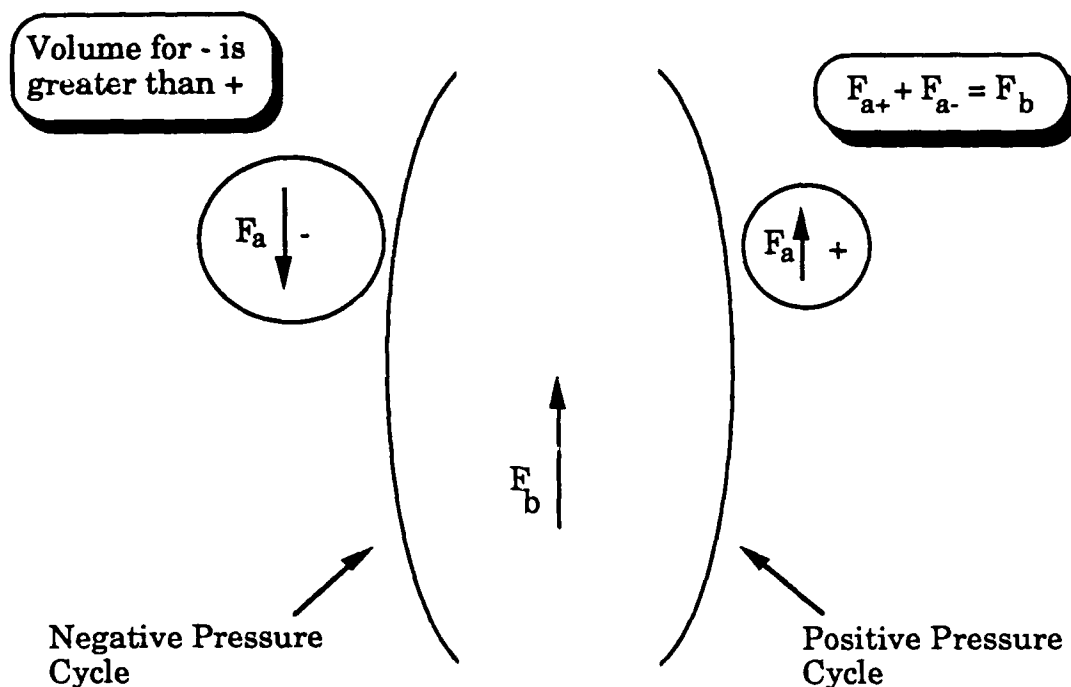


Figure 27: Figurative acoustic levitation technique.

Figure 28 shows a functional diagram of the experimental setup. Basically, a levitation technique was used to trap a single air bubble in water at a specified location in a resonating cell. Using this levitation technique the trapped bubble would linearly pulsate at the driving frequency (at sufficiently low acoustic pressures). Laser light was then scattered off the bubble and detected by one or more photodiodes. The output voltages from the photodiodes were recorded on digitizers and were either displayed on oscilloscopes or stored on a computer.

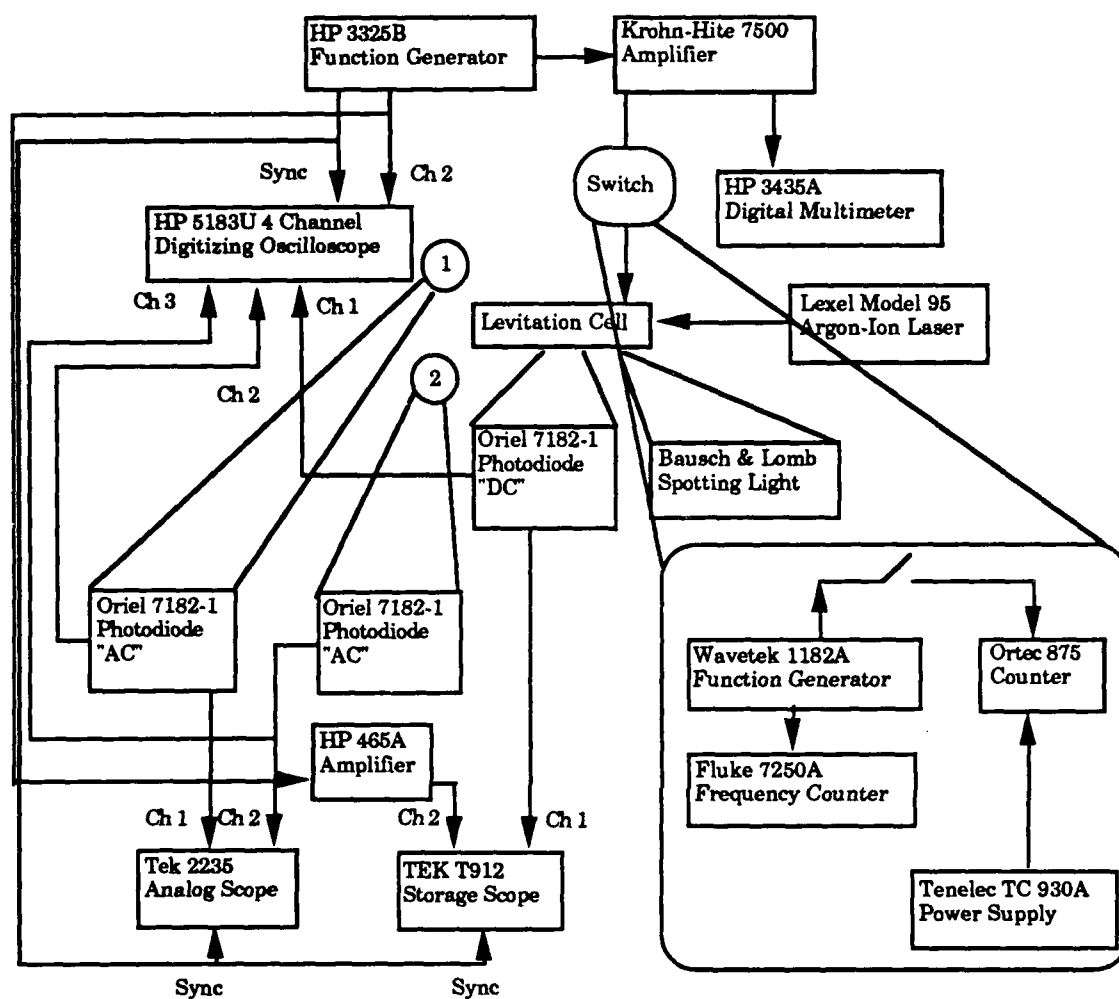


Figure 28: Functional diagram of complete experimental system.

The experimental setup consisted of an acoustic levitation cell filled with water mounted on an y-z translator, a three watt argon-ion laser operating at 440 mW and a wavelength of 488 nm, a maximum of three Oriel photodiode detectors operating with an inline laser-line filter, and associated optics and temperature probes, all mounted on a 4' x 6' optical table with self-leveling pneumatic supports for vibration isolation. In

addition to the equipment on the optical table, two adjoining equipment racks were employed to contain a function generator driving a 75-watt power amplifier used to drive the levitation cell, digitizing and analog oscilloscopes used to observe and record the photodiode output, the power supply/light control equipment for the laser and other support equipment that will be detailed later.

To control most of the experiment, a MNC PDP-11/73 computer was utilized for process control, data acquisition and data storage. The MNC was connected into a VAX network for further data reduction and graphical analysis. With this setup, bubbles in the range of 10 to 100 μ m radius could be generated easily and repeatably. Observations of the bubble's pulsation, either linearly, non-linearly or with shape oscillations, could be made with relative ease. The details of the system are described below.

Apparatus

Levitation cell

Figure 29 shows a detailed drawing of the levitation cell. It consists of two 3" diameter, 1/2" thick, coaxially mounted, cylindrical piezoceramic (PZT-4) transducers separated by a 3" diameter, 1/4" thick, glass cylinder. The glass, transducers and a 3" diameter Plexiglas disc were glued together

with silicone glue to obtain a watertight enclosure. A threaded, cylindrical piece of plexiglass was glued directly to the top transducer with silicone glue. A threaded, watertight lid was fashioned with two "quick-disconnect" connectors, allowing water flow into and out of the cell, maintaining a closed system. Also, a small 6-32 tapped hole was placed in the lid for insertion of a temperature probe.

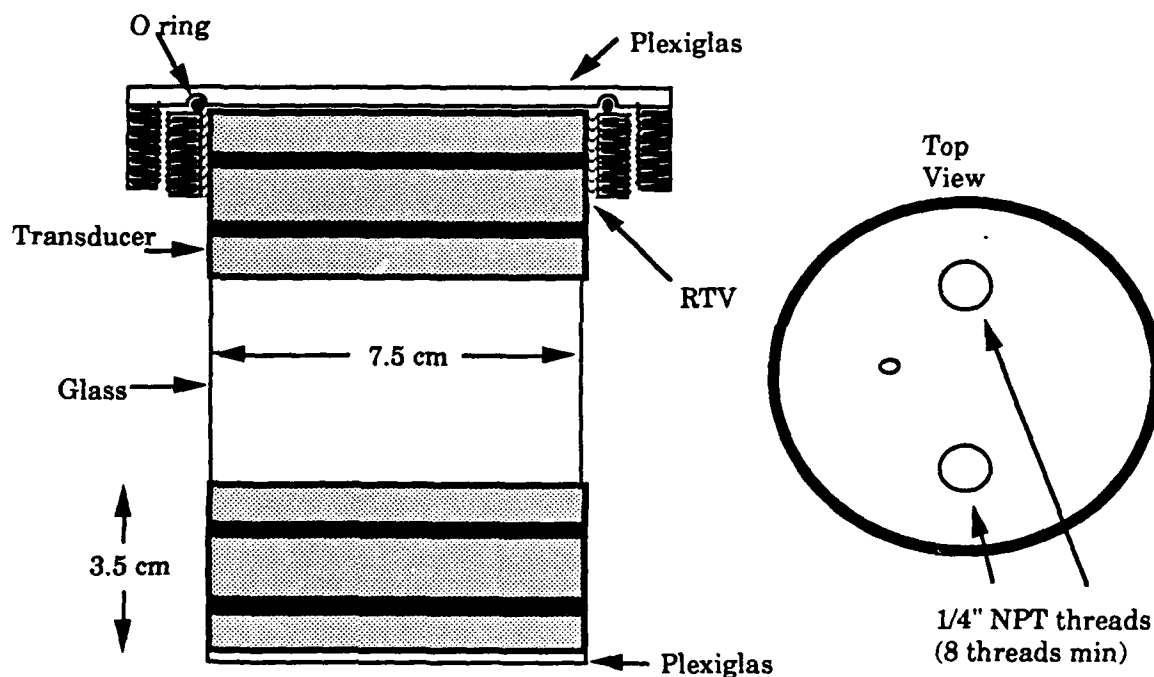


Figure 29: Detail of acoustic levitation cell.

The transducers were operated in parallel and poled to drive the transducers in their thickness mode. The transducers had a resonance frequency of 44 kHz and were driven below that resonance at 22.22 kHz. This choice of frequency was dictated by the geometry of the cell and was

consistently used throughout the experiment. At this frequency, the standing wave generated in the cell was in the (r,θ,z) mode of $(1,0,3)$. The main antinode was located slightly below the geometric center of the cell. This was due to the fact that the cell was not completely filled with water and that the bottom (Plexiglas plate) and top (air) surfaces were not the same.

Extreme care had to be taken to ensure that all surfaces were free of bubbles. Any extraneous bubbles would greatly affect the pressure profile in the cell. The water in the cell was filtered to 0.1 micron and most organics were removed using a Barnstead water-treatment system. Before any experiment, the water was allowed to sit in the sealed cell with low power applied to the transducers for approximately one hour. This allowed the water to reach dissolved gas equilibrium and to be in thermal equilibrium with the cell. Power had to be applied to the cell during this time since the transducers had a warm-up period of about 15 minutes, after which their temperature remained essentially constant. To ensure that the water was saturated with air, the cell was briskly shaken for a few seconds about 5 minutes prior to an experiment.

The cell was mounted on an y - z translation stage, with the x -axis being defined as the direction of the laser beam parallel to the plane of the optical table. The translators provide 0.001" resolution graduations and 1" travel in both the y and z directions. The levitation cell was driven by a 75-watt

Krohn-Hite 7500 power amplifier operated by a Hewlett-Packard 3325B synthesizer/function generator, which provided a 22.22kHz sine wave. The output of the power amplifier was monitored by a Hewlett-Packard 3455A digital multimeter. This arrangement allowed constant monitoring of the power applied to the cell. The function generator was controlled via an IEEE-488 interface by a MNC PDP-11/73 computer. The sine wave produced by the function generator was monitored by a Tektronix T912 10 MHz analog storage oscilloscope after being amplified by 20 dB by a Hewlett-Packard 465A amplifier. With this setup the pressure inside the cell could be precisely monitored and changed as needed.

Optical Scattering System

Figure 30 shows a detailed drawing of the optical scattering system. A 3-watt water-cooled Lexel model 95 argon-ion laser was operated at the 488 nm wavelength producing an effective single line output of 440 mW in the TEM₀₀ mode. A rotating polarizer was attached to the laser providing a 1200:1 linear polarization ratio. The polarizer was positioned such that the electric field vector was parallel to the scattering plane, and consequently in the x-y plane, parallel to the optical table. The laser was mounted on the optical table and the laser beam was raised and reflected 90 degrees by two Oriel 99.8% reflection mirrors to illuminate the levitation cell. The laser beam had a $1/e^2$ width of 1.0 mm nominal and was aligned such that it was

parallel to the optical table and passed through a diameter of the levitation cell. This laser beam width was large enough to completely illuminate all the bubbles in the size range used. The water circulation system, power supply, and light control system were mounted in an adjoining rack.

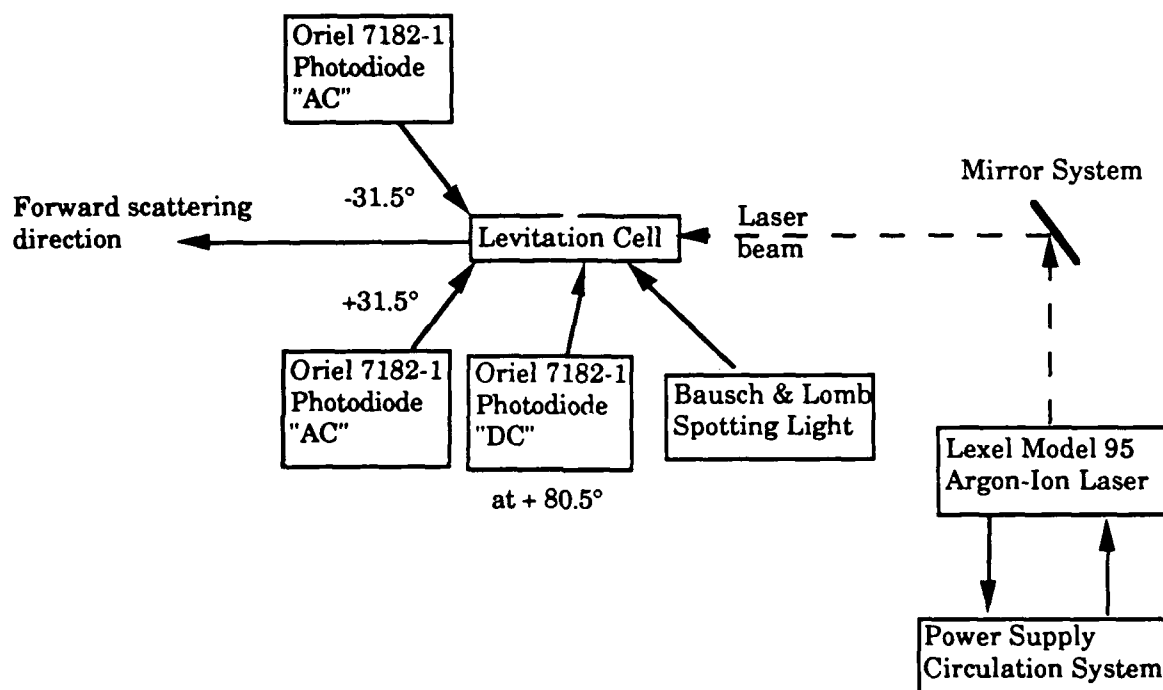


Figure 30: Detail of optical scattering system.

The light detection system consisted of the following: There were a maximum of three Oriol 7182-1 photovoltaic silicone photodiodes with integral preamplifier, each with a detection area of 100 mm². The photodiodes were mounted in Oriol 71910 hoods with an integral 488 nm laser-line transmission filter and circuit board used for mounting the silicone detector. Figure 31 shows the schematic of the detection circuit.

The internal $25\text{k}\Omega$ potentiometer was used to compensate for any dark voltage or bias current. The external resistor controlled the amplification and bandpass characteristics of the photodiode.

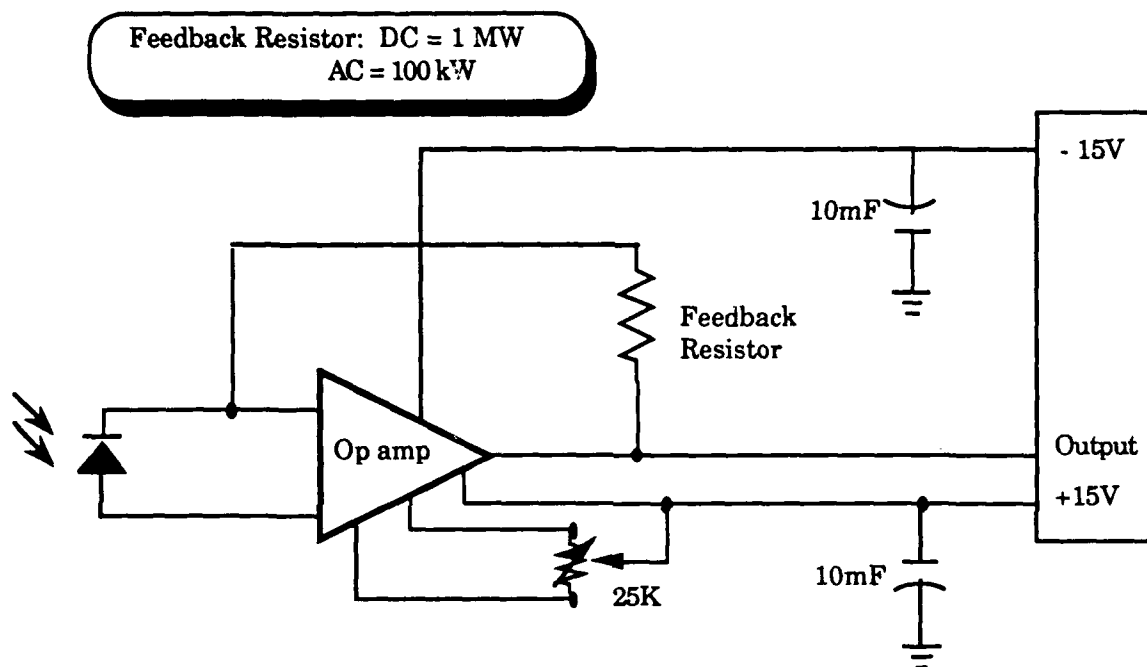


Figure 31: Photodiode detector circuit.

To obtain maximum amplification, bandwidth must be sacrificed. Two different circuits were used in this experiment: The first (or so-called DC) circuit used a $10\text{ M}\Omega$ feedback resistor, which allowed for maximum amplification with a bandpass of DC to 1 kHz . This detector was DC coupled and was used to obtain the average size of the bubble, as will be seen later. The other two detectors were AC coupled and utilized a $100\text{ k}\Omega$ feedback resistor for a bandpass of DC to 100 kHz . By utilizing both types of

detection we could obtain the average size of the bubble while simultaneously viewing the pulsations of the bubble.

The reason for using two different detectors instead of one DC coupled detector was the dynamic range of the digitizers used in this experiment. The change in intensity as a function of time as the bubble pulsated was on the order of 1% to 10% of the average radius, which did not allow for adequate resolution of the pulsation when superimposed on the DC component.

The voltage output by the detectors was proportional to the incident intensity scattered from the bubble, which could be related to the bubble's size. The calibration technique is presented later. The detector output was sent directly to three digitizers in an Hewlett-Packard 5183U waveform recorder/digitizing oscilloscope. Also, the "AC" outputs were connected to a Tektronix 2235 100 MHz analog oscilloscope while the "DC" outputs were connected to a Tektronix T912 10 MHz storage oscilloscope. This allowed for real-time monitoring of the bubble's movement. The "DC" signal was also routed to the MNC computer for threshold triggering measurements. The digital and analog oscilloscopes were triggered by the function generator via a sync signal. The MNC computer also controlled the digitizers as well as the function generator. With this optical detection system, various properties of scattered light from a pulsating bubble could be observed.

Support equipment

In addition to the equipment listed above, various pieces of support equipment were used. To observe the bubbles in the cell without the laser beam activated, a Bausch & Lomb 31-35-30 spotting light was used. This light would backlight the bubble so that it could easily be seen with the eye. To perform the rise-time measurements (discussed later) additional equipment was needed as shown in Fig. 32 below.

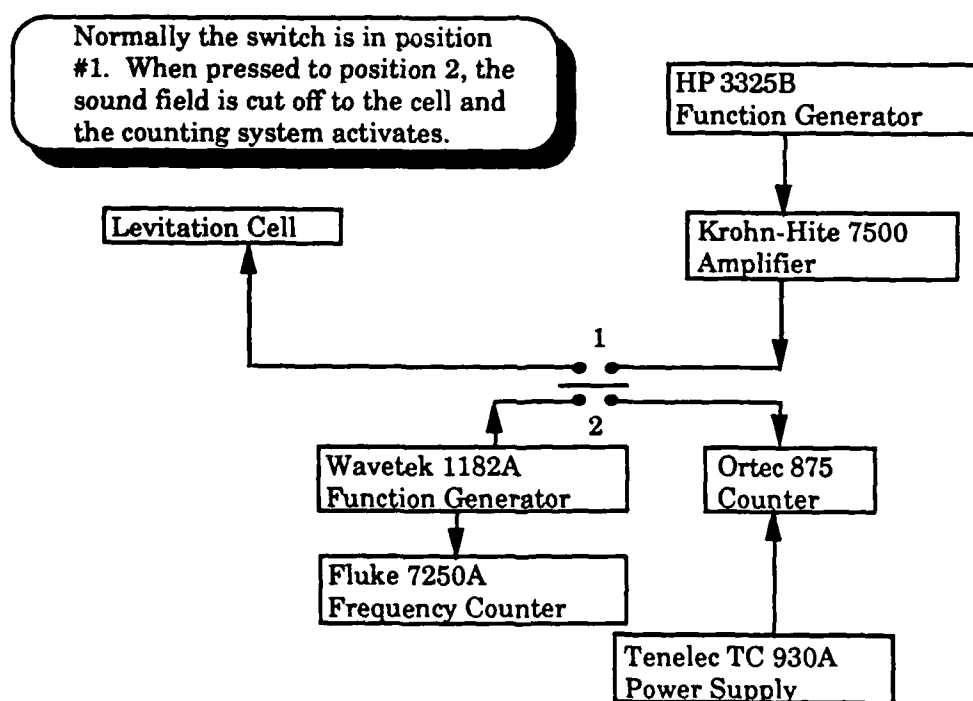


Figure 32: Detail of sound cutoff apparatus.

It was required that the sound field in the cell be intermittently turned off and on while taking rise-time measurements. Also the time that the sound field was off was needed, as well as a means of determining how far the bubble rose while the sound was off. To do the timing, a powered relay was used to turn the sound field off and on using a momentary switch. This switch was powered by a Tenelec TC 930A power supply. The switch also controlled a counting circuit, which consisted of a Wavetek model 1182A 4MHz function generator operating in a square wave mode at 1 MHz, and an EG&G Ortec 875 counter. When the switch was turned off, the function generator drove the counter to count in microseconds. When the switch was released, the counter would stop. Various other optical accessories, mounting hardware and electronic equipment were used at different stages of the experiment and will not be itemized.

With this experimental setup we have improved upon previous work done by Hansen (1983) and by Holt (1988). This system is reliable, repeatable, and relatively easy to use to observe the behavior of a levitated air bubble pulsating in water.

Tuning the levitation cell

In order to maximize the pressure in the cell, the cell must be tuned to its resonance frequency. This allows a minimal voltage to be applied to the

transducers, and since the transducers have a tendency to heat over time, a minimal voltage was desired so as not to affect the temperature of the water in the cell.

During experimental runs, the tuning was accomplished by filling the cell completely with water and then removing 40cc's of water from the cell. Voltage was applied to the cell and a bubble was created (the actual method for bubble creation will be discussed later). Once the bubble was levitated, the generator frequency was altered (while the generator voltage remained constant) so that the bubble would move to its lowest stable position in the cell. If the frequency was higher than 22.22 kHz, water was added to the cell; if lower, then water was removed. As the water level approached resonance height, the applied generator voltage had to be lowered to keep the bubble linearly oscillating. This procedure was continued until the cell was at resonance. Large bubbles (~ 80 microns) were used to calibrate the cell in this way due to the fact that small changes in the cell's resonance characteristics affected large bubbles more than small bubbles.

To check that this was a valid method of calibration, a needle hydrophone was mounted vertically above the cell and immersed in the water. The cell was filled with water and 40cc's were removed. The hydrophone was then placed at the location of the antinode near the center of the cell and the frequency was altered to maximize the signal output from the probe. If the frequency was above 22.22 kHz, water was added to the cell; if less than,

water was removed. When the hydrophone was unable to sense any difference in pressure when the frequency was changed a small amount, around 22.22 kHz ($\sim \pm 20$ Hz), the cell was said to be calibrated to the accuracy of the probe. The probe was then removed, and a bubble was created in the cell. The frequency was again altered around 22.22 kHz to determine if the cell was indeed at resonance. It was determined that the bubble method of tuning the cell was much more sensitive to frequency changes and water level than the probe method. With large linearly pulsating bubbles, one could observe vertical bubble motion when the frequency was changed by as little as 10 Hz.

Keeping the cell tuned over the time period of a data run, which lasted anywhere from one minute to one hour, proved to be a difficult task. The resonance of the cell changed mostly due to temperature fluctuations as the cell did not have any protection from the ambient air. Also, evaporation played a role until a cover was machined to seal the top of the cell. This had an added benefit in that the cell became nearly a closed system, so that dust from the laboratory air did not contaminate the water. The cell was not a completely closed system, however, in that a tiny hole was left in the top for a temperature probe. During data runs, the cell was checked many times for resonance fluctuations. The criterion used was that if the tuning changed more than 20 Hz at any time during the data run, then the data was declared invalid for that run.

Bubble Radius Calibration

The first problem encountered in this experiment was how to accurately determine the size of the air bubbles in the levitation cell. Visual observation was tried, but due to the small bubble sizes and the geometry of the cell, viewing the bubble through a measuring microscope proved futile. A way to determine the bubble size as a function of time in real-time was needed. A technique applied successfully by Hansen (1983) and Holt (1988) employed scattering laser light off the bubble. In effect, the bubble could be "photographed" in real-time as it radially pulsated at frequencies around 20 kHz. In order to use this method, a calibration of the scattered light as a function of bubble radius was needed. To accomplish this, a bubble rise-time technique was employed (Hansen 1983).

Figure 33 shows the experimental arrangement for the rise-time measurements. A levitation cell filled with water was tuned to 22.22 kHz. A bubble was then created and levitated in the cell. The position of the bubble relative to the cell could be altered by varying the voltage applied to the transducers. An argon-ion laser, positioned at a fixed height, illuminated the bubble. The position of the bubble relative to the laser beam could be maintained by vertical (z) and horizontal (y) translators supporting the cell. A photodiode was placed at an angle of 80 degrees from the forward scattering direction. The position of the photodiode was chosen due

to a consideration of Mie scattering theory and convenience (an optical bench with regularly spaced holes was used to mount the apparatus).

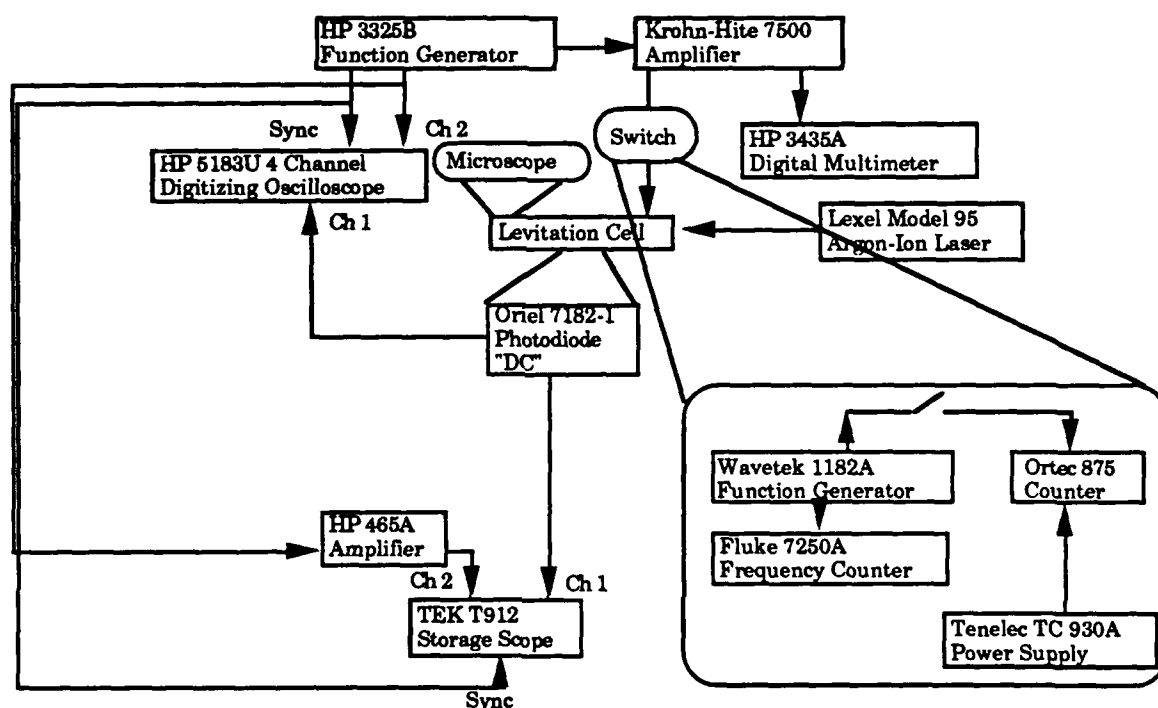


Figure 33: Functional diagram of rise-time measurements experimental apparatus.

A vertically-graduated microscope objective, shown in Fig. 34, was positioned so that the bubble was visible over $\pm 1\text{cm}$ of vertical bubble travel. The graduations in the microscope were calibrated by immersing a small ruler in the water and viewing the ruler through the microscope. Once calibrated, the position of the microscope relative to the laser beam was fixed. In this experiment, the laser beam, microscope and optical detector were attached to an optical bench and not moved. Only the levitation cell had freedom of movement.

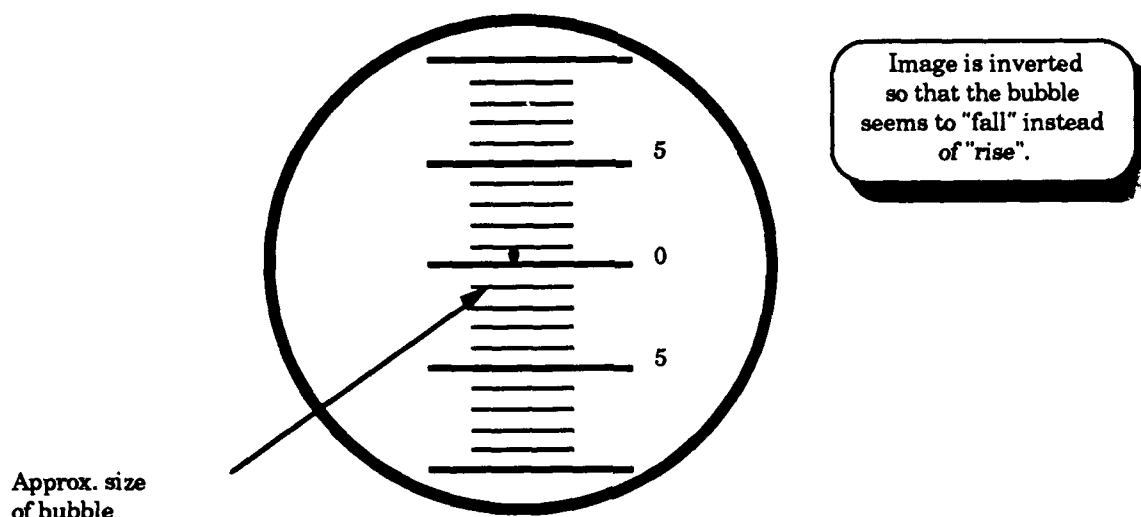


Figure 34: Detail of graduated, viewing microscope.

The rise-time calibration procedure was as follows: First, a bubble was created in the cell and positioned to reside completely in the laser beam. Next, the DC and AC voltage outputs of the photodiodes were recorded. The AC output was viewed on an analog oscilloscope to ensure that the bubble pulsation was linear (i.e., only radial pulsations of a spherical bubble were allowed). The bubble was then moved slightly above the laser beam by adjusting the vertical translator so that the bubble could be viewed by the eye.

Next, while viewing the bubble through the microscope, the sound field in the cell was abruptly turned off which simultaneously activated a timer. Then the bubble was allowed to rise a specified number of microscope graduations. Since the bubble tended to dissolve, this distance was kept to a minimum. To ensure that the bubble did not appreciably dissolve during this measurement, the position of the bubble was noted when the sound field was turned back on. If it did not return to the same location as it originated, or if the DC scattered voltage was different than the original, the data point was declared invalid. The distance the bubbles rose were on the order of 5-15 graduations, depending on bubble size.

When the bubble reached a specified height, the sound field was again activated, thus stopping the timer, and the bubble returned to the center of the cell. The distance risen, the time for the bubble to rise, the relative position of the bubble to a fixed origin, the generator voltage, the ambient temperature, the water temperature, the laser beam intensity, and the AC and DC scattering voltages were recorded. A given bubble was allowed to rise at least five times while all information was recorded. Figure 35 shows the scatter in the rise-time measurements as a function of event. Each individual data point shown represents a separate bubble. The error bars on each point indicate the maximum scatter in the recorded rise-time per bubble. It can be seen that a variety of bubble sizes were considered. The number of rise-time measurements per bubble ranged from five to eight.

From this figure we can estimate the average error in the rise-time measurements to be about 3-5%.

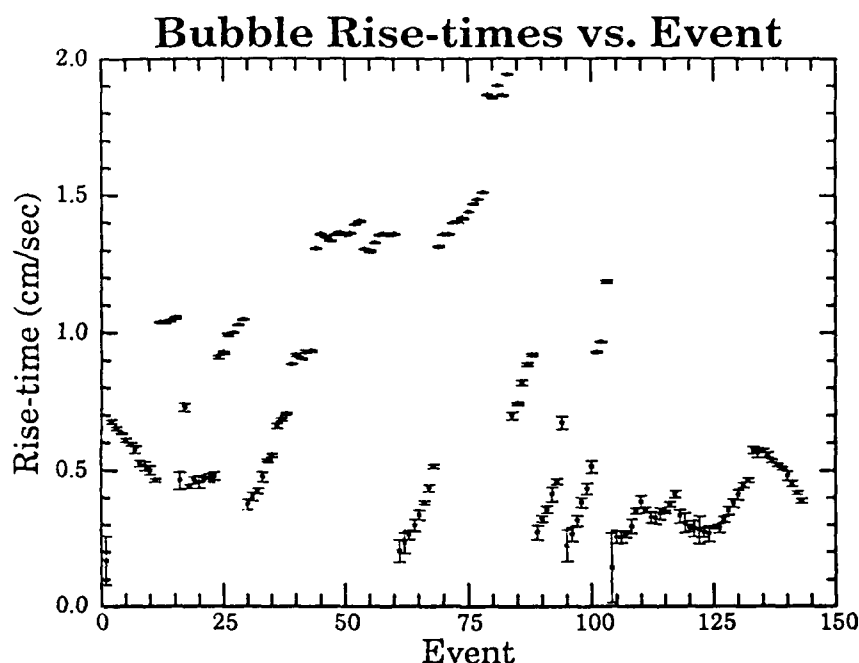


Figure 35: Bubble rise-times vs. event with error bars.

There were three main sources of error. The first was the reaction time of the operator in turning on the sound field after the bubble had risen a certain amount. After many trials, this time was minimized and was reduced to about 0.3 seconds, which remained essentially constant over the data sets. The second error was due to parallax that occurred while viewing the bubble through a microscope. It was determined that these errors tended to increase as a function of the tiredness of the observer. Due to this, the data sets were limited to a maximum of one-hour runs, with no

more than three runs in a single day. The third and most inconsistent error arose from currents in the water, both thermal and pressure related. For bubbles larger than 40 microns in radius these current effects were negligible, but, for bubbles less than 40 microns, the currents could noticeably affect the vertical motion of the rising bubble. When such effects were obvious to the observer, the data set was discarded. In examining Fig. 35 it is obvious that for larger bubble sizes the scatter was greater. This was mainly due to the reaction time error, since the large bubbles were rising very rapidly. To avoid some systematic errors, the data sets were taken two ways. The first was to start with a small bubble, and by forcing the bubble to grow by rectified diffusion, take data sets over a large range of radii for the same bubble. To determine if there was any systematic error, a second method was employed in which bubbles of various sizes were created randomly and only used for one rise-time trial. There was no general trend in the data using either method. Figure 36 shows the average rise-time of the bubbles vs. the scattered DC voltage.

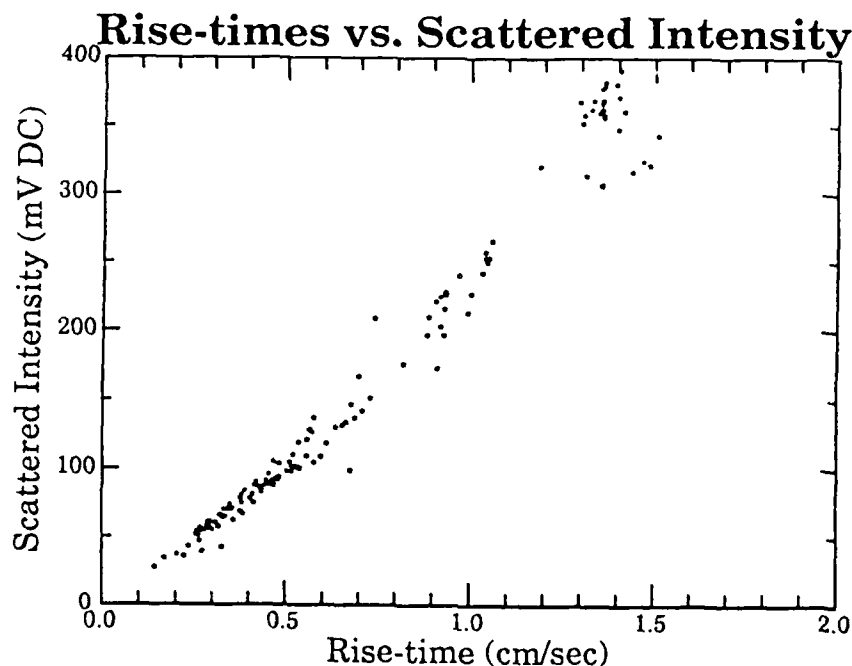


Figure 36: Rise-time vs. DC scattered voltage.

In order to convert the rise-times to bubble radii, a drag law was used. If one has a bubble rising at terminal velocity, U , in water, the drag force is given by (Milne-Thomson, 1968)

$$F_D = \frac{1}{2} \rho U^2 \pi R_0^2 C_D, \quad (7.1)$$

where ρ is the density of water, R_0 is the radius of the bubble, and C_D is the drag coefficient. The forces acting on the bubble are weight, drag and buoyancy, which must be balanced. The resulting equation is

$$F_w + F_d = F_b \quad (7.2)$$

or

$$\frac{4}{3}\pi R_o^3 \rho_a g + \frac{1}{2}\rho U^2 \pi R_o^2 C_D = \frac{4}{3}\pi R_o^3 \rho g, \quad (7.3)$$

where ρ_a is the density of the air in the bubble and g is the acceleration due to gravity. Solving for R_o in Eq. (7.3) one can arrive at

$$R_o = \frac{3U^2 C_D}{8g} \left(\frac{\rho}{\rho - \rho_a} \right). \quad (7.4)$$

Since $\rho_a \ll \rho$ we can write Eq. (7.4) as

$$R_o = \frac{3U^2 C_D}{8g}. \quad (7.5)$$

The drag coefficient, C_D , is normally expressed as a function of the Reynolds number, $Re = 2R_o U / \mu$, where μ is the kinematic coefficient of viscosity of the fluid. For rigid spheres, Stokes calculated $C_D = 24/Re$. Numerous drag laws have been determined empirically. The law which will be used here was determined by Schiller and Nauman (Clift, et al., 1978). This "law", which is actually a fit to a set of experimental data, was chosen so as to be consistent with other recent works (Hansen 1983, and Holt 1988). Equation (7.5) then becomes

$$R_0 = \frac{3U^2}{8g} \frac{24}{Re} (1 + 0.15Re^{0.687}) = \frac{9U\mu}{2gR_0} \left[1 + 0.241 \left(\frac{R_0 U}{\mu} \right)^{0.687} \right], \quad (7.6)$$

where R_0 must be solved for iteratively. Thus, by using this drag law, the rise times can be converted into bubble radii, thereby calibrating the detected intensity of scattered light from the bubble as a function of bubble radius. The calibration result is shown below in Fig. 37.

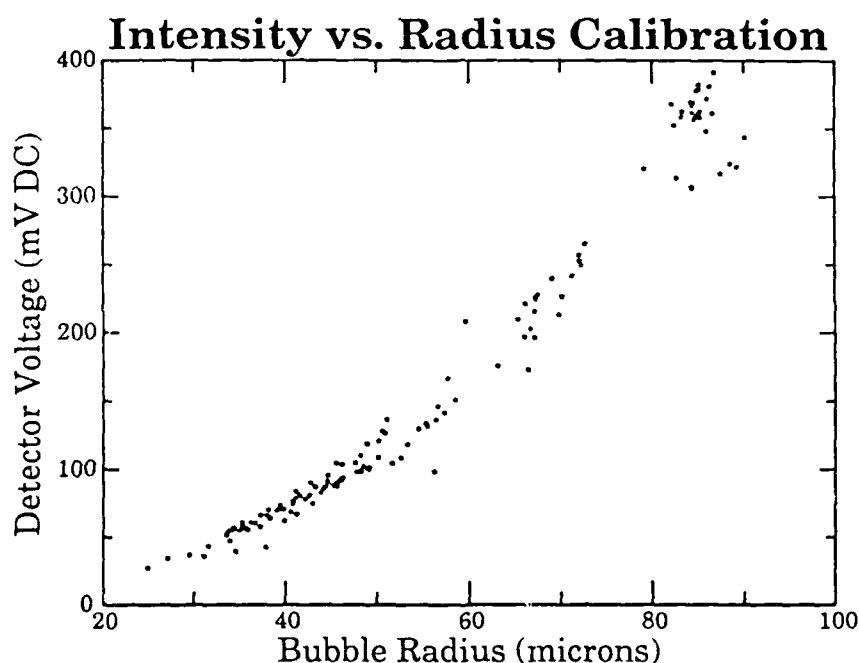


Figure 37: Bubble radius vs. DC scattered voltage.

Mie scattering theory can also be used to test the validity of the rise-time method. Figure 38 shows the predicted relative intensity for various radii at an 80 degree scattering angle (solid line). The symbols represent the

experimental data. Since the incident intensity on the bubble could not be determined, the theoretical prediction was normalized, as in the preceding chapter, and compared to the experimental data. It can be seen that there is general agreement between theory and experiment.

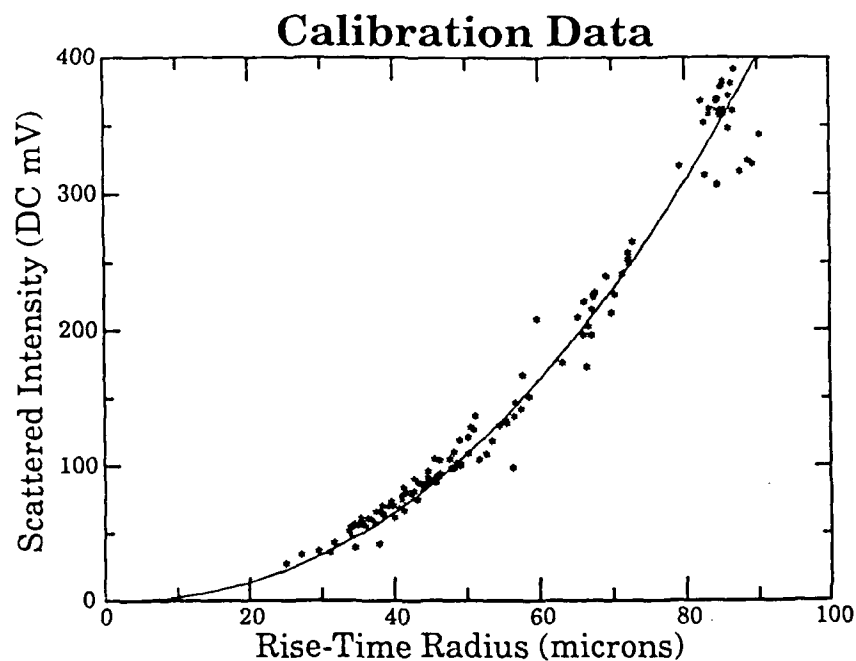


Figure 38: Overlay of Fig. 37 and normalized Mie scattering prediction.

Pressure Calibration

One of the most difficult experimental problems encountered was that of determining the pressure exerted on the bubble in the cell. The experimental apparatus used in measuring the position of the bubble in the

sound field relative to some fixed position as a function of pressure is shown in Fig. 39.

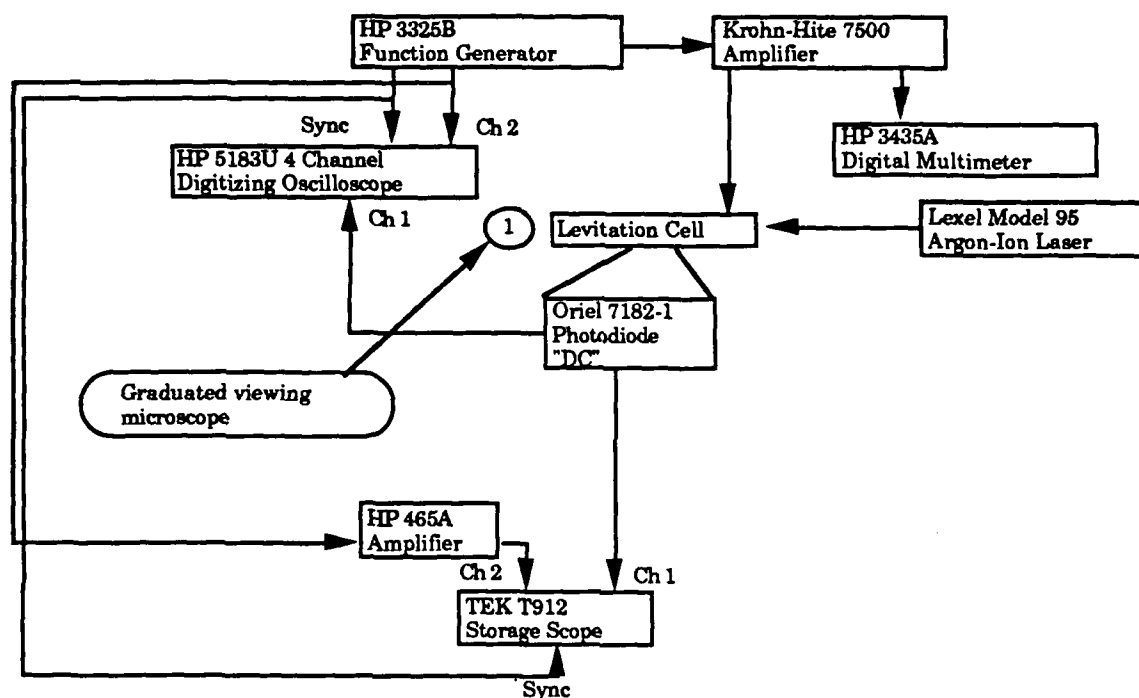


Figure 39: Schematic of pressure calibration technique using bubble position in the levitation cell.

The procedure for pressure field calibration is to generate a bubble of fixed size, and while slowly increasing the pressure in the cell, record its position relative to some fixed point. The results using this technique are shown in Fig. 40.

For an air bubble in water to maintain a stationary position, the applied standing wave acoustic force must balance the buoyancy force, F_b , on the bubble

$$F_b = V\rho g , \quad (8.1)$$

where $V=V(t)$ is the volume of the bubble, ρ is the density of water, and g is the acceleration due to gravity. The time average volume is given as

$$V(t) = \frac{4}{3}\pi R_o^3 \left\langle \left(\frac{R(t)}{R_o} \right)^3 \right\rangle , \quad (8.2)$$

where R_o is the equilibrium radius of the bubble and $R(t)$ is the radius of the bubble as a function of time. The angle brackets represent a time average. Inserting Eq. (8.2) into Eq. (8.1), the average buoyancy force can be written as

$$F_b = \frac{4}{3}\pi R_o^3 \rho g \left\langle \left(\frac{R(t)}{R_o} \right)^3 \right\rangle . \quad (8.3)$$

If the acoustic field is then balanced against the buoyancy force, the bubble will "levitate"; i.e., remain in a stationary position. The acoustic force is given by

$$F_a(r,t) = - \langle V(t) \nabla P(r,t) \rangle , \quad (8.4)$$

where $P(r,t)$ is the acoustic pressure field. We will let the pressure field be represented by

$$P(r,t) = P_\infty - P_a \sin(kz) \cos(\omega t) , \quad (8.5)$$

where P_∞ is the ambient pressure in the liquid in the absence of the acoustic field, P_a is the amplitude of the applied field, ω is the angular driving frequency of the applied pressure, z is the position of the bubble relative to some fixed position on the z axis, and $k = 2\pi/\lambda_z$ is the experimentally measured wave number. The wavelength, λ_z , is determined experimentally by measuring the vertical distance between pressure nodes in the cell using a hydrophone. If we take the gradient of Eq. (8.5) and insert the result into Eq. (8.4), the magnitude of the acoustic force is

$$F_a = \frac{4}{3}\pi R_0^3 (P_a k \cos(kz)) \left\langle \left(\frac{R(t)}{R_0} \right)^3 \cos(\omega t) \right\rangle . \quad (8.6)$$

The condition for the existence of a stationary bubble is found by equating Eqs. (8.3) and (8.6) and solving for P_a :

$$\frac{4}{3}\pi R_0^3 \rho g \left\langle \left(\frac{R(t)}{R_0} \right)^3 \right\rangle = \frac{4}{3}\pi R_0^3 (P_a k \cos(kz)) \left\langle \left(\frac{R(t)}{R_0} \right)^3 \cos(\omega t) \right\rangle , \quad (8.7)$$

which gives:

$$P_a = \frac{\rho g \left\langle \left(\frac{R(t)}{R_0} \right)^3 \right\rangle}{k \cos(kz) \left\langle \left(\frac{R(t)}{R_0} \right)^3 \cos(\omega t) \right\rangle} . \quad (8.8)$$

If we use only linearly pulsating bubbles and assume a small amplitude approximation, then the radius of the bubble as a function of time can be given by

$$R(t) = R_0(1+x(t)) . \quad (8.9)$$

Next, performing a power series expansion in x of Eq. (8.8) and retaining only the first order terms in x we have

$$\left\langle \left(\frac{R(t)}{R_0} \right)^3 \right\rangle \approx 1 , \quad (8.10)$$

$$\left\langle \left(\frac{R(t)}{R_0} \right)^3 \cos(\omega t) \right\rangle \approx 3 \langle x \cos(\omega t) \rangle , \quad (8.11)$$

so that Eq. (8.8) is approximately

$$P_a = \frac{\rho g}{3k \cos(kz) \langle x \cos(\omega t) \rangle} \quad (8.12)$$

The linearity of a bubble oscillation can be determined experimentally by detecting light scattered from the pulsating bubble by a photodiode and displaying the output on an oscilloscope or spectrum analyzer. In theory, all of the quantities in Eq. (8.12) can be measured experimentally. The values were determined as follows: $\rho = 1.0 \text{ g/cm}^3$, $g = 980.0 \text{ cm/s}^2$, $\omega = 2\pi f$, $f = 22.22 \text{ kHz}$, and $k = 2\pi/\lambda_z$. The value of λ_z was experimentally determined by using a hydrophone probe. A needle hydrophone was mounted vertically over the center of the levitation cell, which was filled with water and tuned to 22.22 kHz. The output voltage of the hydrophone was sent to a computer through an analog-to-digital converter. The hydrophone probe was immersed in the water and voltage readings were taken at specified intervals along the z axis. Figure 41 shows an example of a data set obtained in this manner. $\lambda_z/2$ was then determined from the data and λ_z was found to be 53 cm.

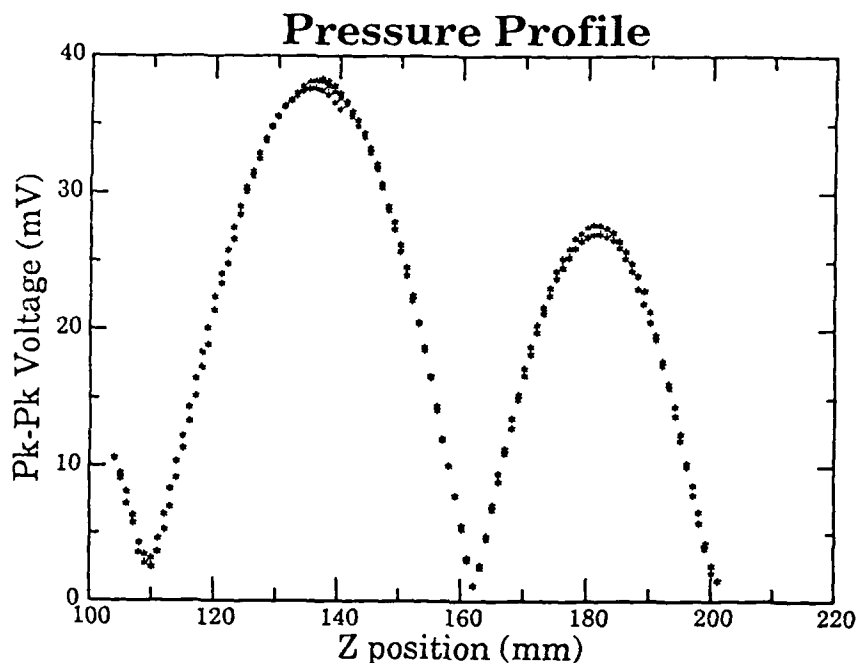


Figure 41: Pressure amplitude (given in hydrophone voltage units) in the cell as a function of height at 0.1 bar.

The only quantity left to determine was $\langle x \cos(\omega t) \rangle$, which is a convolution of the amplitude of the linear oscillation of the bubble as a function of time and a cosine function of the driving frequency. This value was theoretically calculated by Crum and Prosperetti (1983). The value of $\langle x \cos(\omega t) \rangle$ could also be determined experimentally. The only difficulty, which is a substantial one, is to determine the phase relationship between the driving frequency and the bubble pulsation frequency. This information was not available in this experiment. It was reported by Holt (1988) that the theoretical method used was not sufficient to determine the pressure in the cell and therefore he used a "best-fit" resonance approach to determine the pressure. Gaitan (1989) has indicated that an improved theoretical analysis

can be done, which he compared against a "free-field" method which gave more consistent results.

A more direct approach to measuring the pressure in the cell is to immerse a hydrophone into the cell and measure it directly. The experimental problem is how to measure the pressure in the cell without the hydrophone affecting the pressure profile. To do this we used a long needle hydrophone mounted vertically along the z-axis of the levitation cell. The hydrophone was mounted to allow three degree of freedom of movement. The hydrophone could travel in the x, y, or z plane with a 0.1 mm resolution and with 50 mm of travel in the x and y directions, and along the entire cell height in the z direction. In this way the entire pressure profile in the cell could be measured. The probe's movement was independent of the cell's movement. Data was taken at 1.0 mm intervals. This extremely time-consuming task gave the pressure profile in the cell in three dimensions.

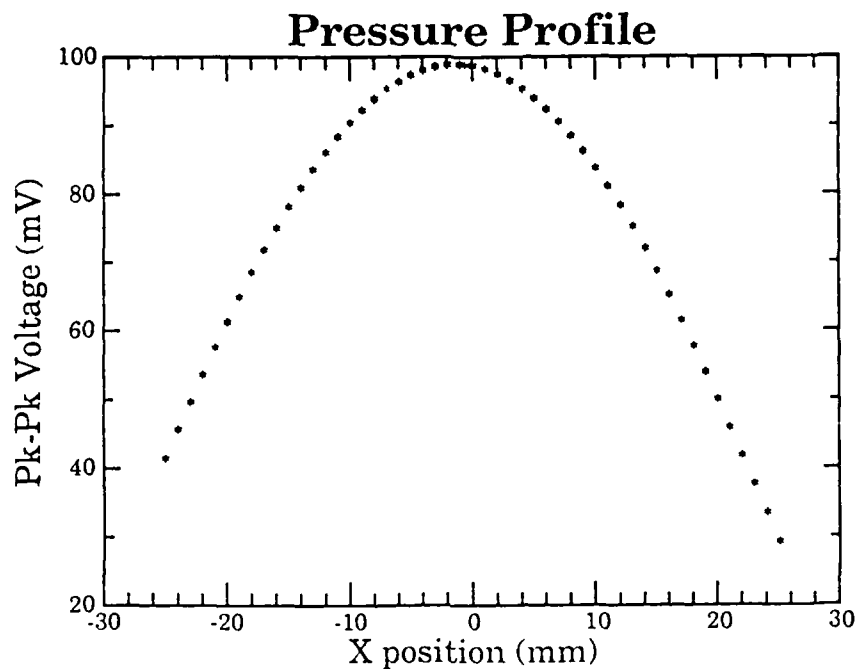


Figure 42: Slice of the pressure in the x direction along the $y=z$ =center axis.

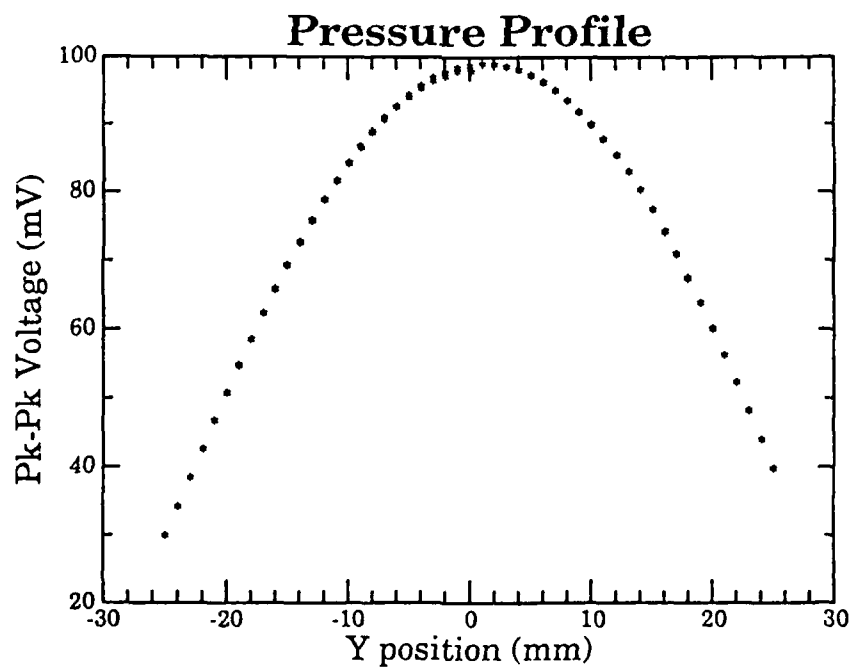


Figure 43: Slice of the pressure in the y direction along the $x=z$ =center axis.

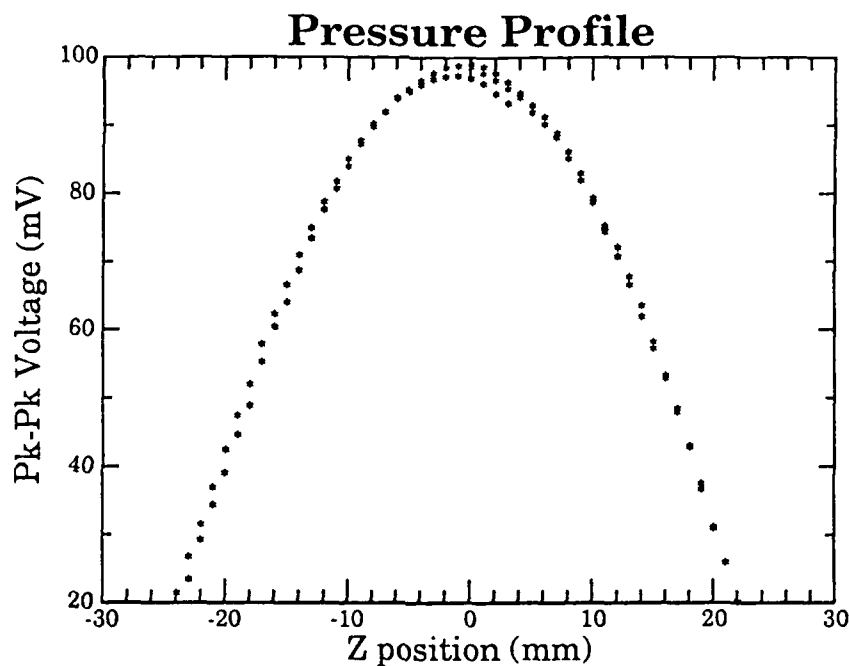


Figure 44: Slice of the pressure in the z direction along the $x=y$ =center axis.

Figures 42, 43, and 44 represent the data obtained along the x, y, and z axes respectively. Due to the long period of time that elapsed between measurement of the first and last data point in a data run, the following procedure was used: First, the cell was tuned to 22.22 kHz, then the probe was moved along a particular direction starting in the center of the cell. The probe was moved in 1.0 mm increments and the voltage from the probe was recorded. When the probe reached a cell wall, its direction was reversed and data was taken until the probe reached the opposite cell wall. The the probe direction reversed once again and was incrementally brought back to the center of the cell while taking data. The scatter in the data of Figs. 42, 43, and 44 may be due to evaporation or changes in the ambient

temperature which affects the cell's resonance characteristics. Only data sets in which the ambient temperature varied less than 0.2°C were declared valid.

To ensure the uniformity of the pressure profile in the cell various readings of the pressure at many locations in the cell were made. Once the data was obtained, it was entered into a computer. A program (SLICE) was written to view the data in 2 dimensional slices (see Appendix D). The output of SLICE consisted of an axial plane slice in any direction. The pressures were color coded and output on a color printer. The pictures shown in Figs. 45 and 46 were reproduced using a gray scale instead of color.

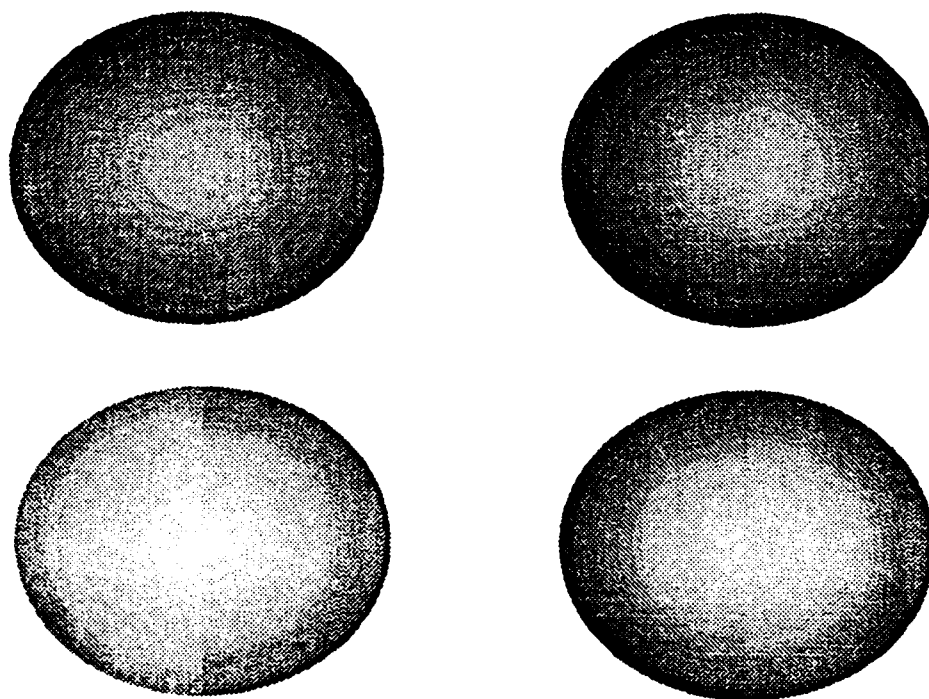


Figure 45: Results from the SLICE program.

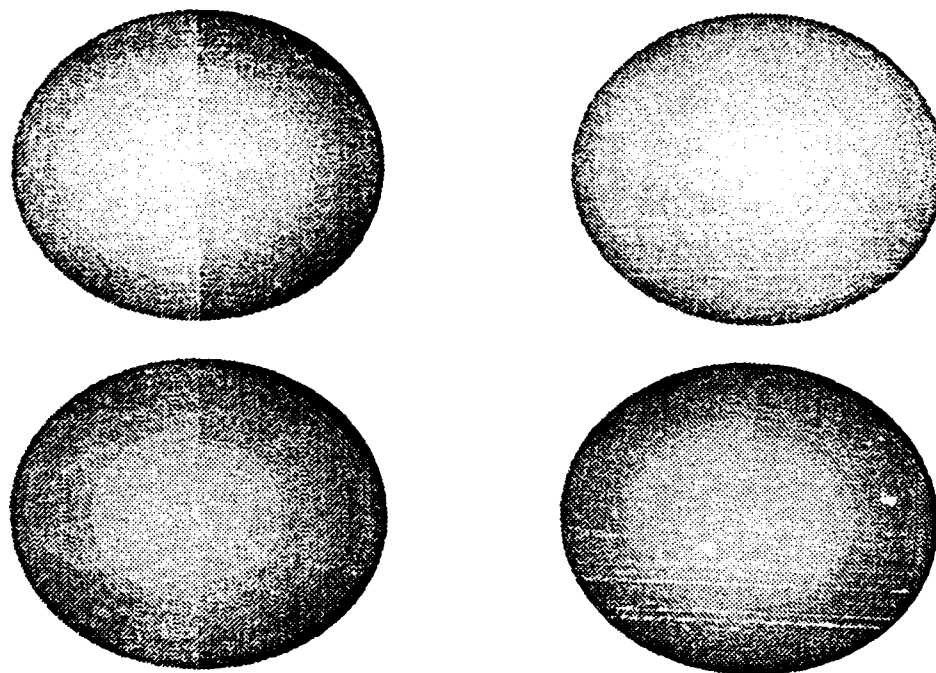


Figure 46: Results from the SLICE program.

The results confirm the supposition that at 22.22 kHz, the cell is operating in a pressure mode (r,θ,z) of $(1,0,3)$, which is basically 53 mm ($1/2$ wavelength) in the z direction with an antinode near the center of the cell and nodes near the top and bottom.

To assess the effect of the hydrophone on the pressure field seen by the bubble, the following procedure was used: First, a bubble was created in the cell and levitated at a pressure slightly above its minimum trapping pressure. This is the minimum pressure necessary to keep the bubble from rising to the surface. Then the hydrophone was placed in the cell and moved toward the bubble along the z axis directly above the bubble while the

position of the bubble was viewed through a microscope. It was found that the bubble did not move perceptibly until the probe was within 3-5 bubble diameters of the bubble. Next, the bubble was levitated at a pressure comparable to the experimental pressures used in detecting shape oscillations (approx. 0.2 bar). The probe was again inserted into the water and moved toward the bubble. This time the probe was able to come as close as one bubble diameter before perceptible bubble movement was observed. This procedure was repeated with various size bubbles ranging from 20 to 70 microns in radius. (It was noticed that the larger bubbles were more affected by the probe than the smaller bubbles.) It was thus assumed that the hydrophone probe did not appreciably affect the pressure field.

The next task was to calibrate the needle hydrophone probe. This was accomplished by using a Brüel & Kjær (B&K) 8103 calibrated hydrophone attached to a B&K type 2635 charge amplifier. The B&K probe's diameter of 9.5 mm was too large to use directly in the cell without greatly affecting the sound field. To calibrate the needle probe against the B&K probe a large tank filled with water was used as shown in Fig. 47.

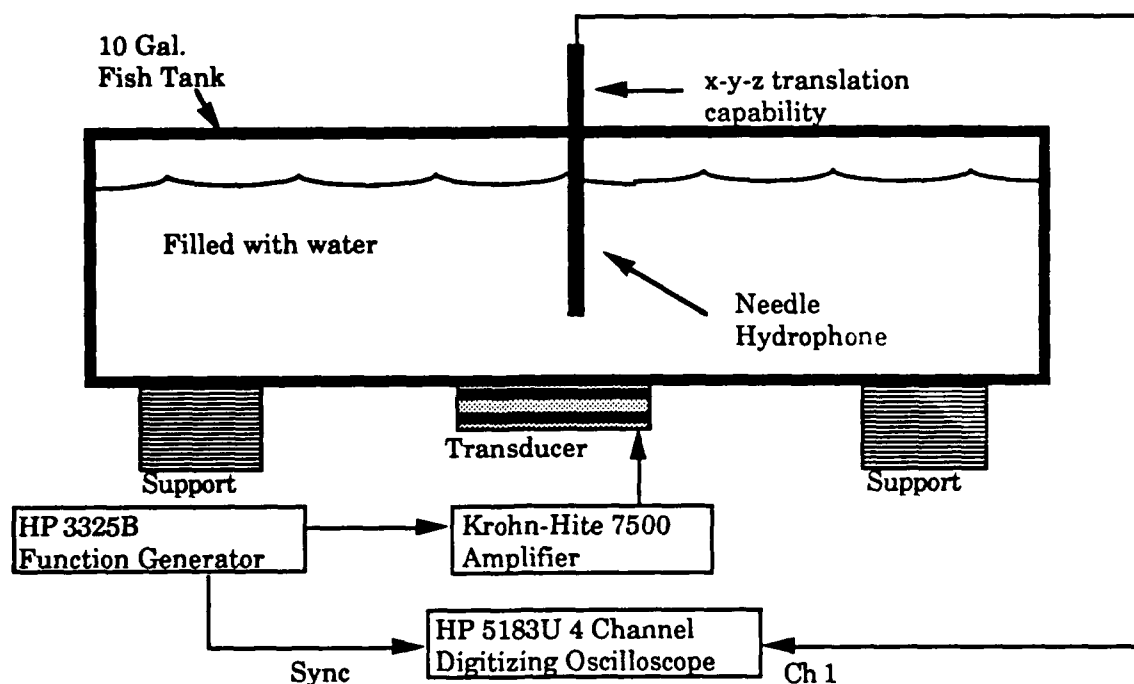


Figure 47: Figure of large tank calibration apparatus.

A transducer similar to that used on the cell was attached to the bottom of the tank and was made to oscillate at 22.22 kHz. Using this apparatus, pressures similar to those observed in the cell were created in the tank. The voltages produced by the needle probe were used as a measure of the similarity. First the needle probe, and then B&K probe were immersed into the tank at the same location, and the pressure was varied over the range indicated by the cell measurements. Figure 48 shows the voltage produced by the needle probe and the B&K probe as a function of voltage applied to the transducer.

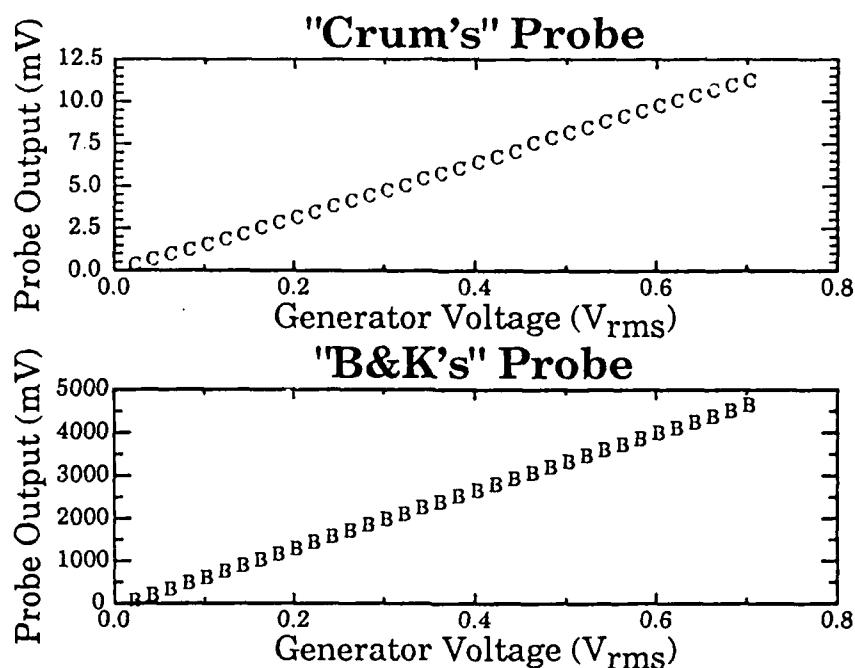


Figure 48: Needle and B&K probe voltage vs. generator voltage.

Using the known calibration constant for the B&K probe, the B&K output voltages were converted to pressures. The needle probe voltages were then compared to the B&K pressure values and the ratio was taken to obtain the calibration constant for the needle probe. Once the needle probe was calibrated, the pressure in the cell could be related to the voltage applied to the cell as shown in Fig. 49. Thus, we now had a way to determine the pressure inside the cell based on an external measurement.

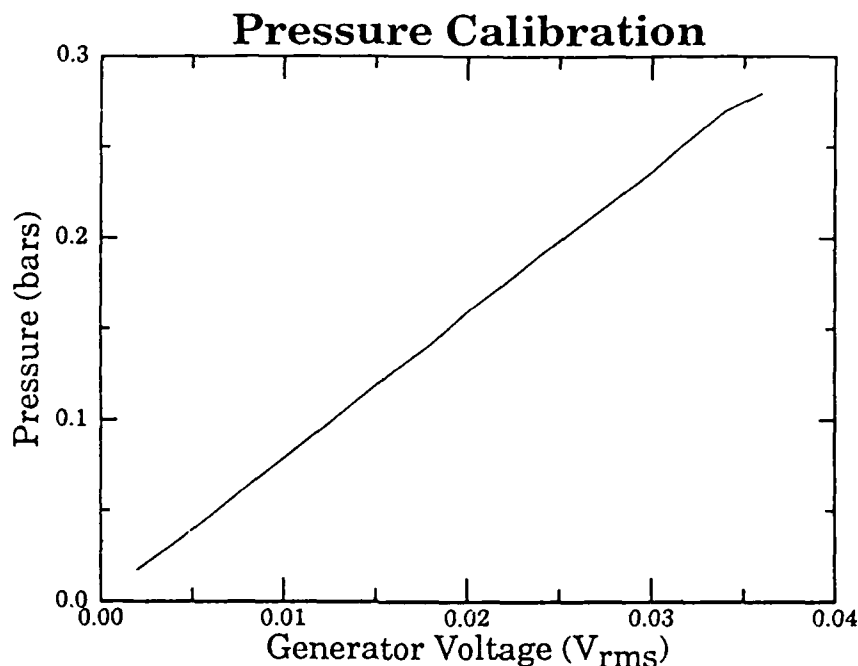


Figure 49: Pressure in the cell vs. applied generator voltage.

Laser radiation pressure effects

One of the questions that arose when dealing with a powerful laser to scatter light off a pulsating bubble is: how much effect does the laser beam have on the bubble? Qualitatively, the incident radiation from the laser will move the bubble away from its acoustic equilibrium position in the direction of the beam. If the bubble had a constant radius, then it would move a distance, x , in the direction of the beam until it reached a new equilibrium position, provided that the beam was not powerful enough to completely move the bubble out of the antinode region. If the bubble was

pulsating radially, then the pressure exerted on the bubble by the beam would also vary and the bubble would oscillate along the beam path. To see if these effects were significant in this experiment, the problem was examined quantitatively.

The radiation pressure can be defined as

$$P = \frac{F_r}{\pi a^2} = \frac{IQ}{v}, \quad (9.1)$$

where $v=c/n$ is the velocity of the light in the medium, $Q \approx 0.3$ (Unger and Marston, 1988), P is the radiation pressure, F_r is the radiation force, a is the bubble radius, and I is the irradiance. If we let

$$P' = I\pi\rho^2, \quad (9.2)$$

where P' is power and ρ is the beam radius then

$$I = \frac{P'}{\pi\rho^2}. \quad (9.3)$$

Inserting Eq. (9.3) into Eq. (9.1) we have

$$P = \frac{0.3P'}{\pi\rho^2v}. \quad (9.4)$$

Given that $\rho = 6.25 \times 10^{-8} \text{m}^2$, and $v = 2.25 \times 10^8 \text{m/s}$, then the pressure is

$$P = (6.79 \times 10^{-3} \text{ s/m}^3) P' . \quad (9.5)$$

In this experiment the maximum amount of power the laser produced was about 0.5 watts. Assuming that all of the power from the laser impinged on the bubble (in practice less than 0.3 watts were actually incident on the bubble), the maximum pressure would be

$$P = 3.4 \times 10^{-3} \text{ N/m}^2 , \quad (9.6)$$

which gives a force of

$$F_r = 0.0107 a^2 . \quad (9.7)$$

As an example, the force on a bubble of 50 microns would be

$$F_r = 2.67 \times 10^{-11} \text{ N} . \quad (9.8)$$

To compare this value to the force exerted on the bubble due to the levitation pressure, we may assume that the acoustic force is equal to the buoyancy force. The buoyancy force can be written as

$$F_b = \frac{4}{3} \pi a^3 g \rho_w , \quad (9.9)$$

where $g = 9.8 \text{ m/s}^2$ is the acceleration due to gravity, and $\rho_w = 1000 \text{ kg/m}^3$ is the density of water. From this relation one finds that $F_b \approx 5.13 \times 10^{-9} \text{ N}$ which is about 200 times greater than the pressure due to the laser radiation.

Next, if we allow the bubble to radially pulsate by say 20% of its radius at a frequency of 20 kHz, then a 50 micron bubble will grow to 60 microns and shrink to 40 microns. At 40 microns, the radiation force is

$$F_r(40) = 1.71 \times 10^{-11} \text{ N} , \quad (9.10)$$

and at 60 microns

$$F_r(60) = 3.845 \times 10^{-11} \text{ N} . \quad (9.11)$$

The change in force is given by

$$\Delta F = 2.136 \times 10^{-11} \text{ N} . \quad (9.12)$$

To determine how far the bubble will move we let

$$F = ma \approx m \frac{\Delta v}{\Delta t} , \quad (9.13)$$

$$\Delta v = \Delta F \frac{\Delta t}{m} = \frac{\Delta F \Delta t}{\frac{4}{\rho_w 3 \pi a^3}} , \quad (9.14)$$

or

$$\Delta x = \Delta v \Delta t = \frac{3 \Delta F \Delta t^2}{4 \pi a^3 \rho_w} . \quad (9.15)$$

Letting $\Delta t = 1/20 \text{ kHz} = 5.0 \times 10^{-5} \text{ sec}$ then

$$\Delta x = 1.02 \times 10^{-10} \text{ m} \approx 0.0001 \text{ microns.} \quad (9.16)$$

Thus, the effects of bubble movement due to the laser radiation are negligible in this experiment.

Shape thresholds

Introduction

Once all the calibrations and other experimental preliminaries had been performed, it was possible to examine the question of when radial instabilities (i.e., shape oscillations) occurred. It has been shown earlier that shape oscillations exhibit a threshold behavior. The next step was to perform experimental verification of the thresholds predicted by the analytical and numerical calculations. The experimental arrangement is shown in Fig. 28.

A levitation cell, tuned to 22.22 kHz, was used to trap a single pulsating air bubble in a cylinder filled with water. The bubble was illuminated by an argon-ion laser beam operating at 440 mW @ 488 nm. The scattered light from the bubble was detected by three silicon photodiodes, converted to voltages, digitized, and stored on a computer.

There were two basic experimental methods to determine when the bubbles reached the shape oscillation point: First, a small bubble was created in the cell and the pressure was kept constant at a value slightly above the

rectified diffusion level (Appendix A). The bubble would then grow until it reached a critical radius, defined by the pressure, where it became radially unstable. The pressure and radius were recorded and the procedure was repeated. The second method (later referred to as "fast thresholds") employed a technique in which once a spherically symmetric bubble was obtained, the pressure in the cell would be rapidly increased (keeping the bubble in the laser beam) until the bubble became radially unstable. Again, the pressure and radius were recorded. Both methods had strengths and weaknesses which are detailed later. Approximately 75 thresholds were obtained using the first method and over 200 thresholds were obtained using the second method. What follows is a detailed description of the apparatus, methods and results.

Apparatus

The apparatus used in determining the shape oscillation thresholds has been detailed earlier. In both methods of determining the shape oscillation threshold three silicon photodiodes were placed around the cell. The "DC" or bubble sizing photodiode was placed at 80.5° from the forward scattering angle. The two "AC" photodiode detectors were placed at $+31.5^\circ$ and -31.5° from the forward scattering angle. The AC detectors were placed at symmetric angles about the forward so that when the bubble was spherically pulsating, the output from the detectors would be identical. If

the detectors showed any significant difference, either in phase or in amplitude, a shape oscillation was assumed.

The first step in the experimental methods was to create a bubble inside the levitation cell. The phenomenon of acoustic cavitation was used for bubble creation (Atchley, 1985). A bubble was created in the cell by increasing the standing wave pressure until cavitation was evident. Once this occurred, the pressure was quickly decreased by about 20% until a formation of streamers was visible. The streamers consisted of about five or six "lines" of micro-bubbles which formed, moved toward the center of the cell, coalesced, and then disintegrated. The streamers were from one to two centimeters in length and were fairly straight. This process happened in a repetitive cycle. The process from creation to destruction of the micro-bubbles happened over a period of about one second with a repetition rate of about three seconds. Visually, there was a uniform, wide-spread micro-bubble field (about 4 cm²) which, at first, slowly condensed. Then the condensation rate increased until enough bubbles coalesced at the vertex to form a central bubble and the remaining micro bubbles formed five or six lines of bubbles leading in toward the center bubble. This central bubble underwent rapid growth while gyrating wildly. The bubble mass then reached a point when it would collapse and form a field of micro bubbles which began the process again. It was possible to carefully lower the sound pressure field in the cell during the rapid growth stage to obtain a stably oscillating bubble. This took some practice and was perfected over time.

The bubbles formed this way were normally around 20 to 30 microns in radius. The pressure required to keep these small bubbles from dissolving was on the order of 0.5 bar.

First experimental method

There were two experimental methods used in determining when the bubble exhibited radial instability.

- 1) Constant pressure, variable bubble size
- 2) Constant bubble size, variable pressure

The first method is described as follows: First, the levitation cell was tuned to 22.22 kHz and was allowed to reach thermal equilibrium. Next, the photodiodes were activated and the laser beam illuminated the levitation cell. Next, a background measurement of the photodiode outputs was taken. This background essentially defined the smallest resolvable bubble detectable by the photodiodes. The measured background corresponded to a bubble whose radius was about 20 microns. Bubbles with radii smaller than 20 microns could not be resolved out of the background noise. Next, a stably pulsating bubble was formed, as described above. The acoustic pressures in the cell were about 0.4 bar immediately following the bubble's creation. At these pressures the bubble would grow fairly rapidly, about 5 microns per minute, since this pressure exceeded the rectified diffusion

threshold. Next, the bubble was allowed to grow while maintaining a constant pressure in the cell. As the bubble grew, it would rise in the cell. Thus, since the laser beam was kept at a constant height, the bubble was constantly repositioned in the laser beam, by moving the cell vertically, so that the entire bubble was illuminated by the beam. To determine when the bubble was completely in the beam, the output from the "DC" detector was maximized by altering the vertical and horizontal position of the cell. Once a bubble had been positioned horizontally, the horizontal position was fixed. (The x-y position of the bubble should not change due to changes in the pressure or changes in the bubble size.)

Periodically, about once every 20 seconds, during the bubble growth the following information was recorded: the time, the ambient temperature, the water temperature and the DC scattered voltage (voltage output by the "DC" photodiode). The driving frequency and voltage applied to the cell were kept constant. The AC detectors were monitored visually on an analog oscilloscope. A computer digitized the AC and DC signals, recorded the time and the DC value, subtracted the AC signals, computed a "shape threshold" criterion value and waited for a shape oscillation.

As the bubble grew, the rms output signals from the photodiodes increased so that the computer had to adapt itself to dynamic conditions and thus continually redefined the shape oscillation threshold criterion. This criterion was normally defined to be >20% deviation from background of the

difference of the two AC signals. Since this criterion did not always work, an override feature was built in that allowed the operator to force an initiation of the recording process.

The recording process, initiated by either the computer or operator, consisted of triggering all four channels on an HP 5183U waveform recorder/digitizing oscilloscope, recording the frequency and amplitude of the driving voltage applied to the cell, the time at which the event occurred, and, if computer initiated, whether the operator considered the event to be a true shape oscillation or simply a random fluctuation. Since the computer's reaction rate was much faster than the human rate, the data acquired by computer trigger showed more precisely the actual transition from spherical to non-spherical pulsation of the bubble.

In most cases, the threshold, once reached by this method, usually resulted in a stable shape oscillation lasting from a few seconds to many minutes. The bubble either remained stable or disintegrated. If it remained stable, the pressure was lowered until the pulsation became spherical and the process was repeated. At this point the bubble either grew slowly or dissolved depending upon the driving pressure.

This method was applied to either growing or dissolving bubbles. No distinction was made in the threshold measurements. If the bubble could not be made stable or it disintegrated, then a new bubble was formed and

the procedure was repeated from the beginning. This method had an advantage over the second method (described later) in that rectified-diffusion rate measurements could be made simultaneously since the time and bubble size were recorded during the growth stage of the bubble. Appendix A outlines these observations.

Transfer of the data from the HP to the computer for storage was not speedy. This, coupled with the fact that the bubbles were either growing or dissolving very slowly, meant that, at most, ten data sets could be taken per day. Examples of these data sets are shown in Figs. 50-55.

Figures 50-55 are described as follows. Each figure contains eight graphs. The title of each figure contains the date the data was taken and the data set number (described later). The four graphs on the left side of the page represent the actual data as a function of time. In the right is the corresponding amplitude spectrum in frequency space. The frequency has been normalized to the driving frequency of 22.22 kHz. The top graph on the left is simply the driving signal applied to the levitation cell. Note, that all "y" axis units have been normalized to the amplitude of the driving frequency. The next three graphs show the output from the three photodiode detectors, DC, AC#1, and AC#2, respectively.

Figure 50 is a sample background data set. This set the limits on the bubble size resolution. Figure 51 is a example of a bubble in shape oscillation in

which the amplitude of the oscillation seen by the two AC detectors is different. Here the spectrums are similar and have harmonics of the fundamental and first sub-harmonic. Figure 52 is an example of a bubble undergoing a complex shape oscillation. Both the phase and amplitudes seen by the detectors are different. Also, a broad frequency spectrum with many harmonics and sub-harmonics are evident. Figure 53 represents one of the more typical shape oscillations. The phase difference between the detectors has not changed, while the amplitude of one detector is much larger than the other detector. There is also an absence of any sub-harmonics. Figure 54 represents a stable shape oscillation. This particular bubble exhibited the same shape oscillation both in phase and amplitude for a period of about 5 minutes. Figure 55 represents another stable shape oscillation, although in this case the phase between the detectors slowly changed over time, indicating a precession. For most of the stable shape oscillations generated, a precession behavior was normally observed. It is interesting to note that the rate of the precession could be controlled by the operator by simply raising or lowering the applied pressure by a very slight amount.

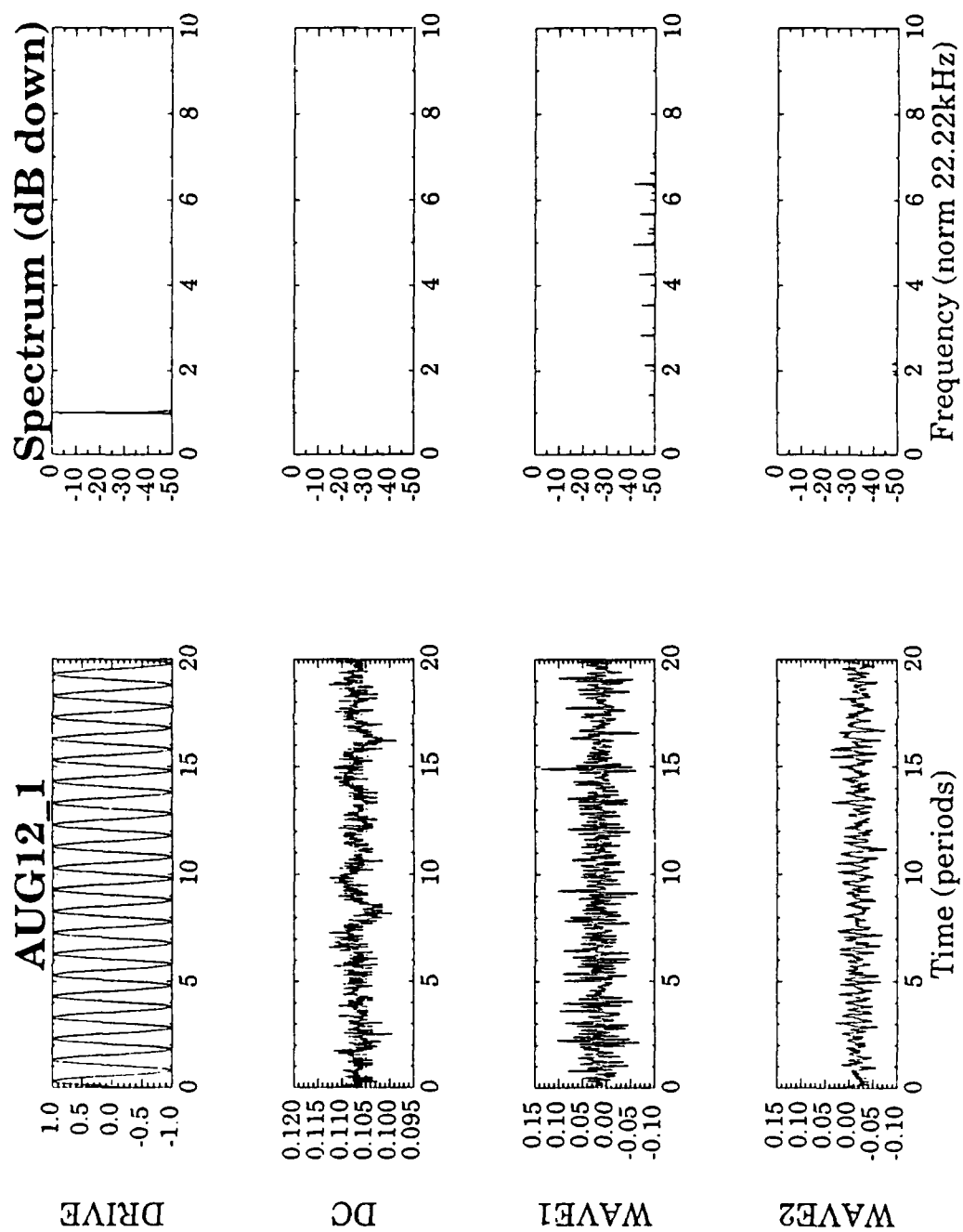


Figure 50: Experimental data set #1 (background)

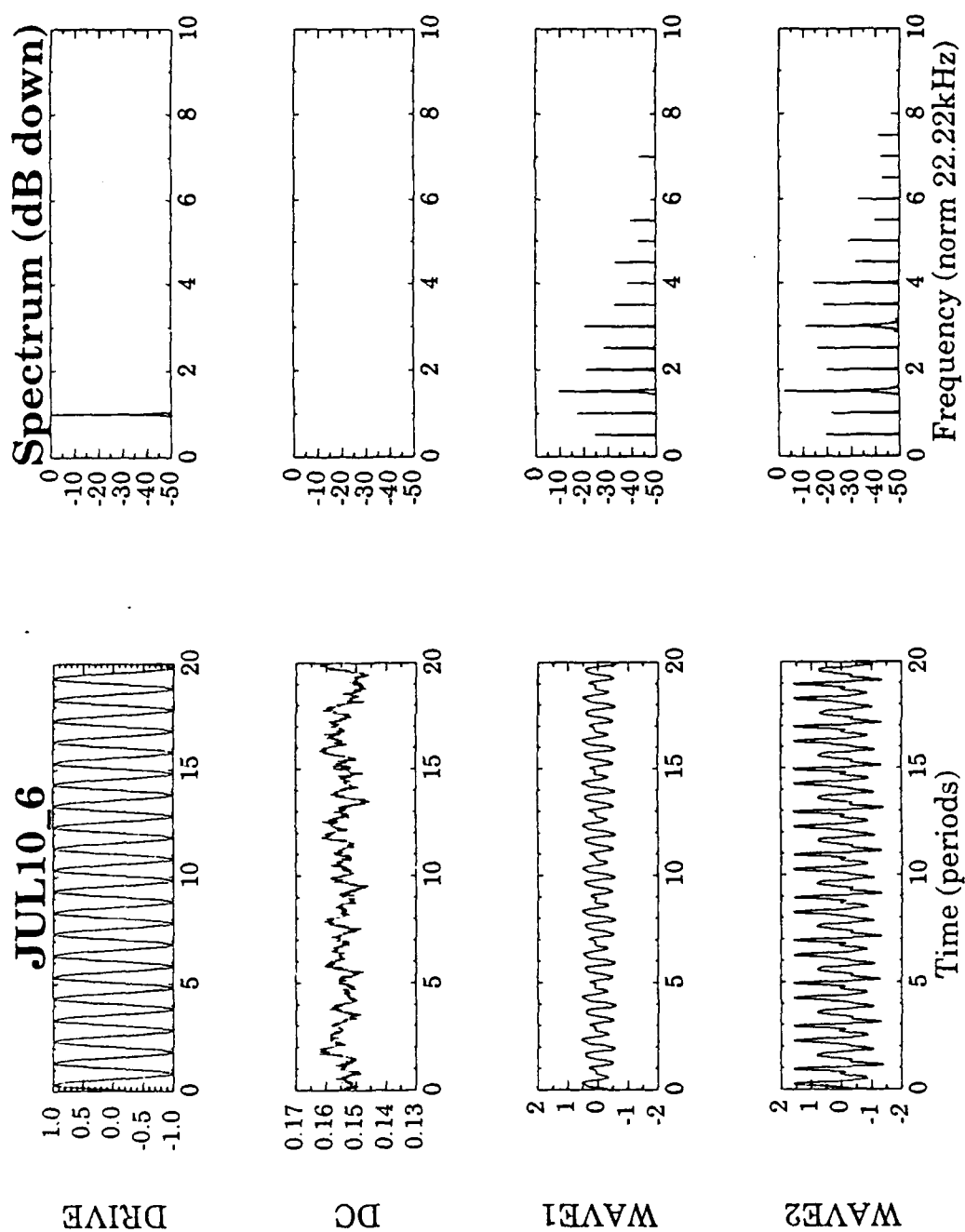


Figure 51: Experimental data set #2

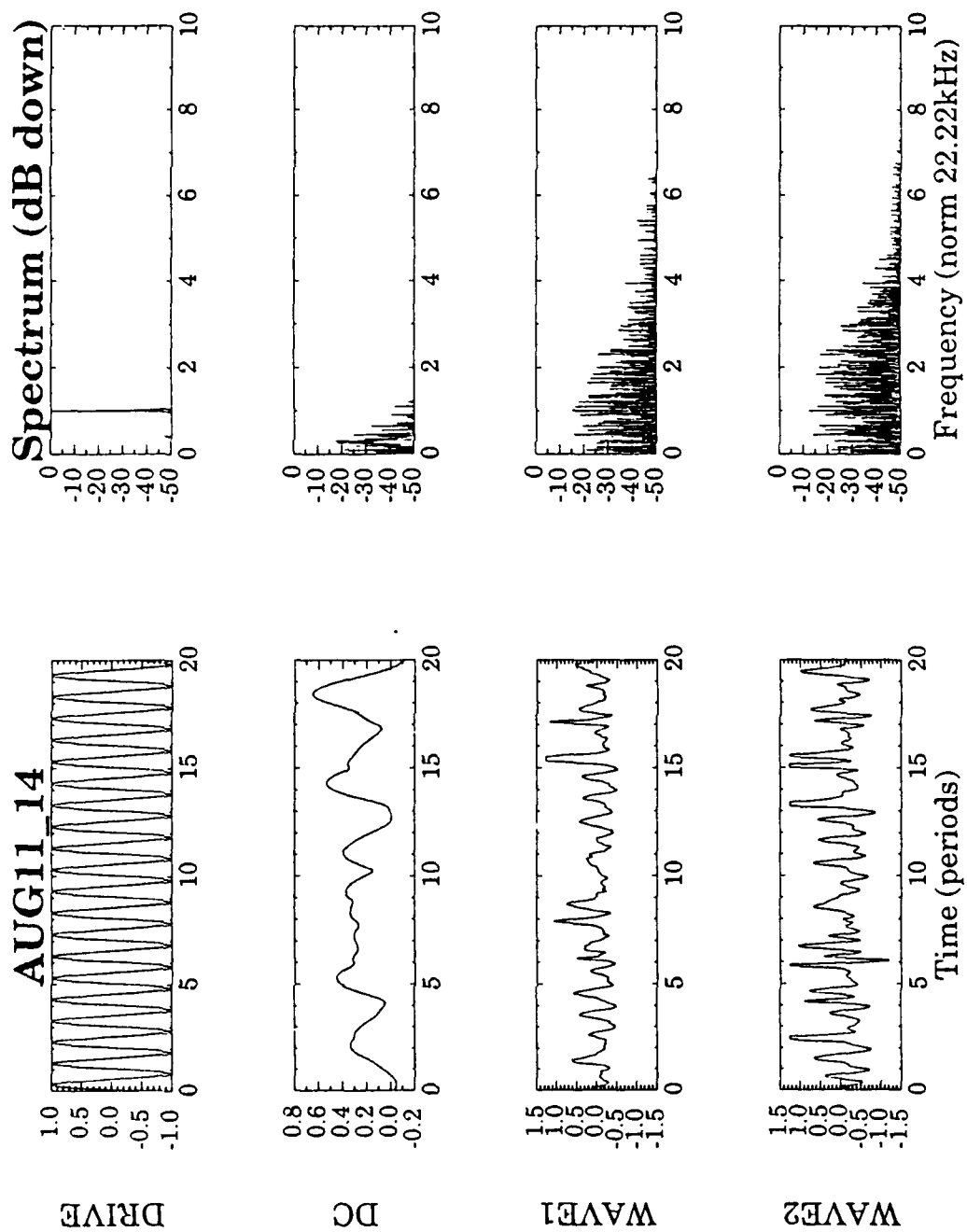


Figure 52: Experimental data set #3

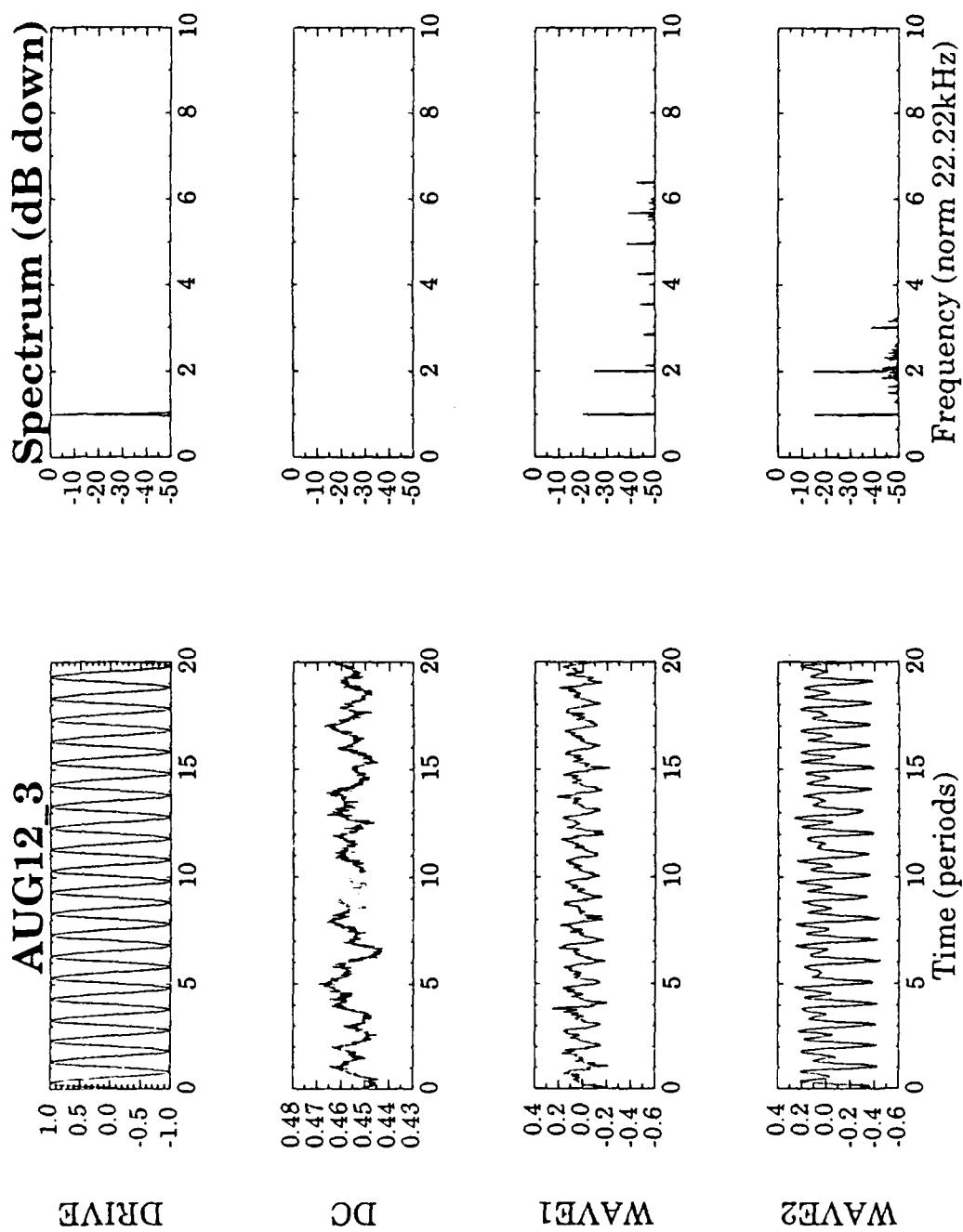


Figure 53: Experimental data set #4

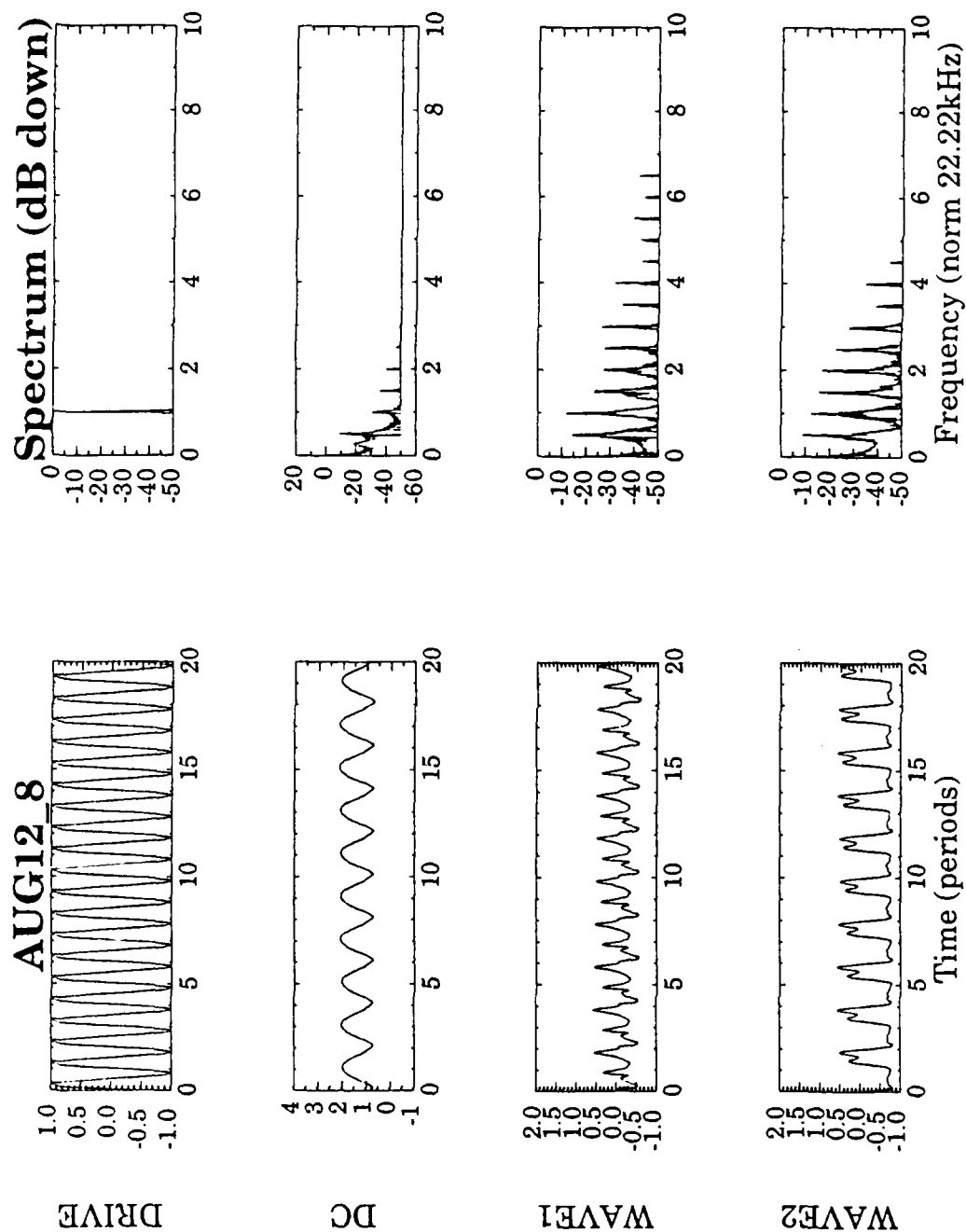


Figure 54: Experimental data set #5

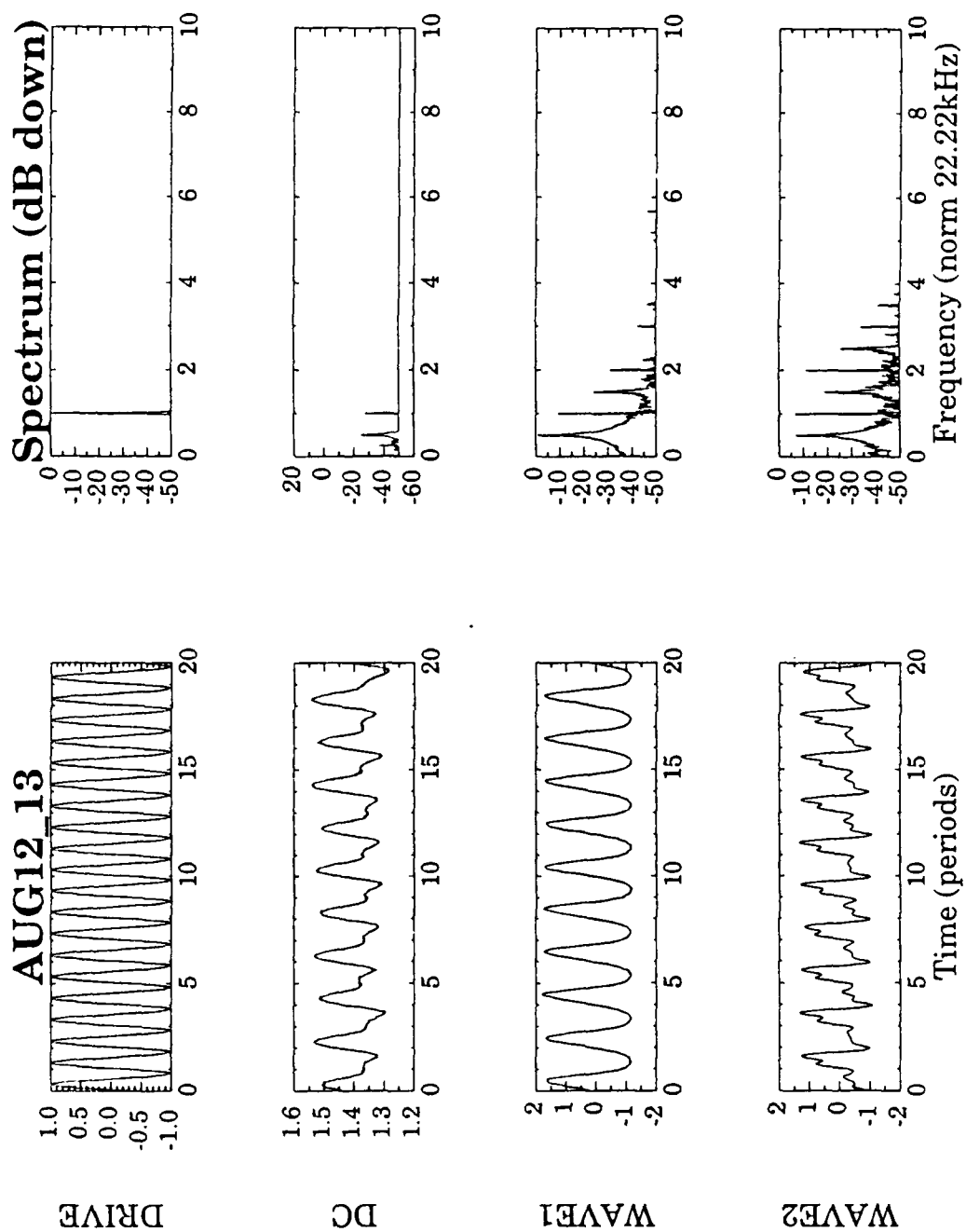


Figure 55: Experimental data set #6

For each data run (usually one per day) approximately ten data sets were taken. The first data set was taken to determine the background readings for the photodiodes. The variability during all the data runs was slight. The second data set was taken to determine if the photodiode detectors and all associated electronics were operating properly. To compare the data runs on a day-to-day basis, a linear spherically pulsating bubble was placed in the laser beam and a data set was taken. Acquiring this data set had the added benefit of automatically calibrating any amplitude or phase differences between the two AC photodiodes. It was found that the variability in phase between the detectors was insignificant. The amplitude differences were detectable over time but were easily compensated for. The rest of the data sets during a data run were taken when a shape oscillation threshold had been reached. The table below shows an example of the relationship between data runs, data sets, and data descriptions.

Data Run	Data Set	Description
AUG18	#1	Background calibration
AUG18	#2	Linear calibration
AUG18	#3	Shape oscillation set #1
AUG18	#4	Shape oscillation set #2
AUG18	#n	Shape oscillation set #n
AUG20	#1	Background calibration
AUG20	#2	Linear calibration
etc.	etc.	etc.

Normally only four or five sets were kept per day, as some had to be discarded for the following reasons: large temperature fluctuations ($>\pm 1^\circ\text{C}$), bubble contamination, or missed thresholds due to too rapid threshold approaches that the computer did not catch. This first method was used throughout the experiment. To increase the number of threshold measurements taken per day an alternate method was needed.

Second experimental method

The second method employed, the so-called "fast-transition method", involved essentially the same apparatus as the first method. Here the bubble radius was kept constant, while the pressure was rapidly increased to initiate a shape oscillation. A bubble was created in the manner described above and was made to pulsate spherically. Next, the bubble was positioned completely in the laser beam. The DC scattered voltage (i.e., the bubble size), the ambient temperature, the water temperature, the driving frequency, and the applied voltage (i.e., the pressure amplitude) were recorded. Then, while adjusting the cell to keep the bubble in the laser beam, the pressure was increased as rapidly as needed until the bubble exhibited non-radial motion as evidenced by the AC detectors. To determine when the bubble went into shape oscillation, the outputs of the AC detectors were input into an analog oscilloscope which had the ability to invert the signal in channel B and add channels A and B (Tektronix 2235 100 MHz).

The resulting trace only showed a difference from zero when the bubble exhibited non-spherical motion.

The difficulty involved in this method was keeping the bubble positioned correctly in the laser beam during the pressure increase. Since any change in the pressure would move the bubble downward, the cell had to be raised to compensate. These tasks were very difficult to perform rapidly enough. Thus, a computer program was written to perform most of the data acquisition tasks. The program SHAPE (Appendix D) controlled the function generator which set the driving frequency and voltage applied to the levitation cell. It also recorded the time, frequency, voltage, ambient temperature, water temperature and DC scattered voltage. SHAPE was set to perform the following tasks each time the spacebar on the terminal keyboard was pressed:

- Record the time
- Record the driving frequency
- Record the driving voltage amplitude
- Record the ambient temperature
- Record the water temperature
- Record the DC scattered voltage
- Increment the applied voltage by a given amount.

The operator would then reposition the bubble in the beam and the process would repeat. When the bubble went into shape oscillations, another key

was pressed (<return>), the above items were recorded, and control of the function generator was returned to the operator who immediately lowered the pressure in the cell until the bubble returned to a spherical pulsation. The process was then repeated. Since no actual waveforms were recorded using this method, and since there was a very short time interval between the start and end of each threshold measurement, it was possible to take many threshold measurements each hour. As before, measurements were discarded if the temperatures changed more than $\pm 1^\circ\text{C}$, if the bubble became contaminated or if a false threshold was obtained. The majority of the threshold data was taken using this method.

Experimental observations

Type 1-5 behaviors

Before we present the threshold results, some experimental observations will be discussed. In observing the traces on the analog oscilloscopes of the output of the AC photodiode detectors, some interesting phenomena was made apparent. There seemed to be five different ways a bubble could make the transition into a surface oscillation. As noted before, the analog trace representing the difference between the two AC detectors was essentially zero when the bubble was spherically pulsating. The first type (type 1) of behavior associated with non-radial motion was apparent when the analog

trace deviated from zero into a small amplitude sine wave as shown in Fig. 56.

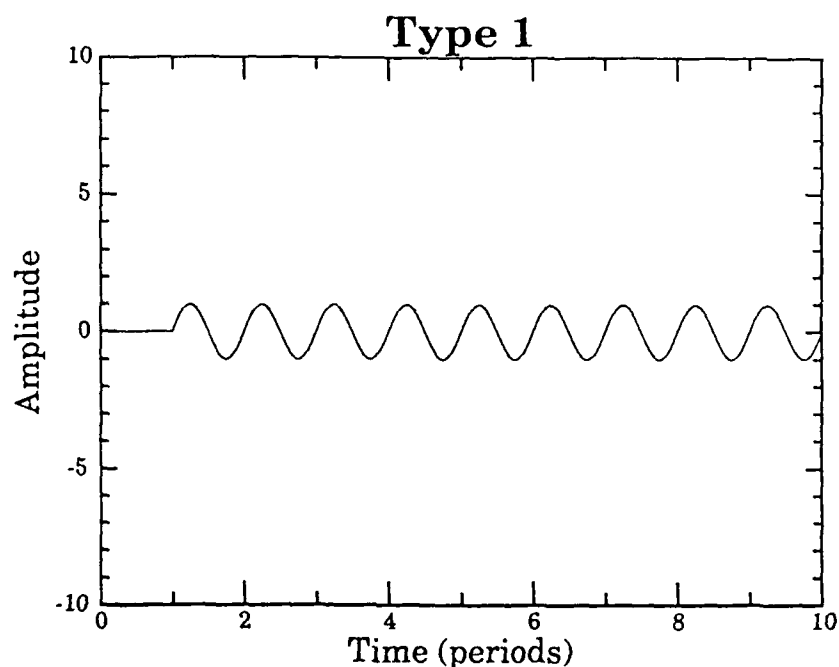


Figure 56: Type 1 trace.

The frequency of this wave was always the same as the driving frequency and the amplitude was an order of magnitude smaller than the other types of traces. Bubbles which exhibited this type of behavior were on the order of 60 microns in radius.

The second type (type 2) of behavior was observed for the majority of the bubbles. This behavior is illustrated in Fig. 57.

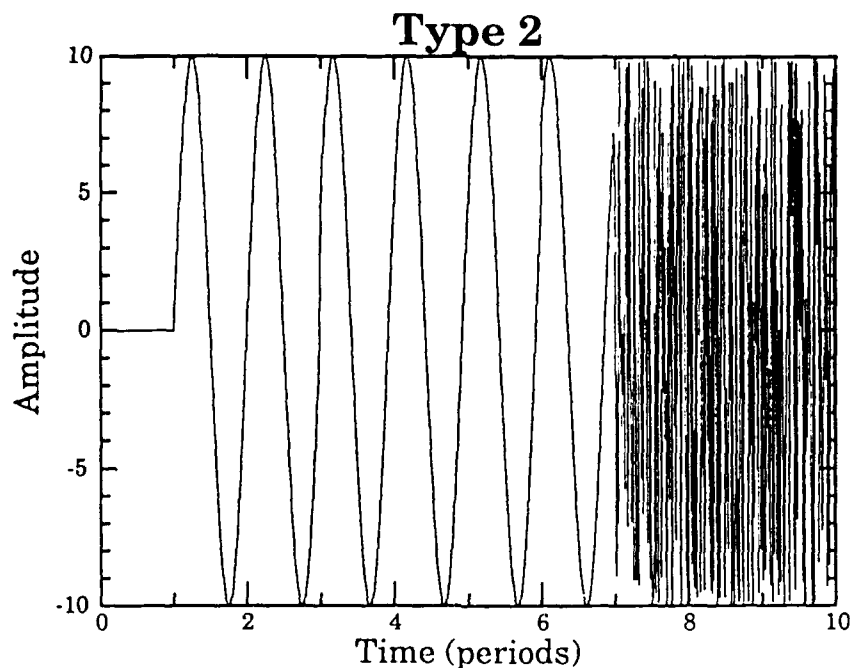


Figure 57: Type 2 trace.

This type was characterized by a large amplitude (at least an order of magnitude larger than type 1) and a sharp demarcation, in applied pressure, between radial stability and instability. The phase difference usually began as a constant and then became more and more unpredictable. Bubbles of this type could normally be brought back to spherically stable radial motion by simply lowering the pressure.

The third type (type 3) of behavior was rarely observed. Figure 58 is an example of type 3 behavior.

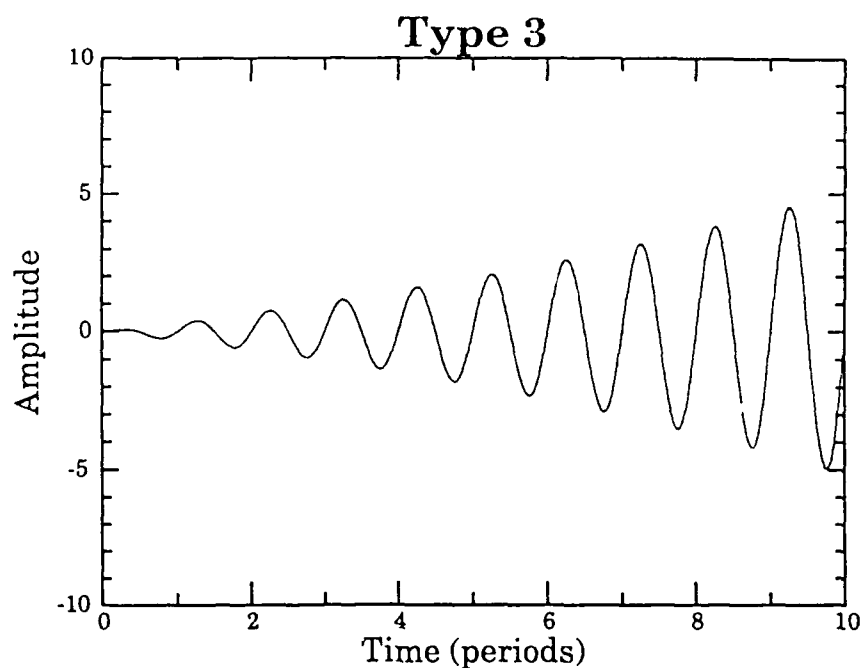


Figure 58: Type 3 trace.

The analog trace exhibited a constant phase and medium amplitude appearance and would normally be long lived (on the order of minutes). It was difficult to class this as a shape oscillation as changes in pressure did not seem to affect the trace to any great degree. Normally, bubbles of this type would transfer into a type 2 bubble by increasing the pressure a sufficient amount.

The fourth type (type 4) of behavior was the most interesting. An example is shown below in Fig. 59.

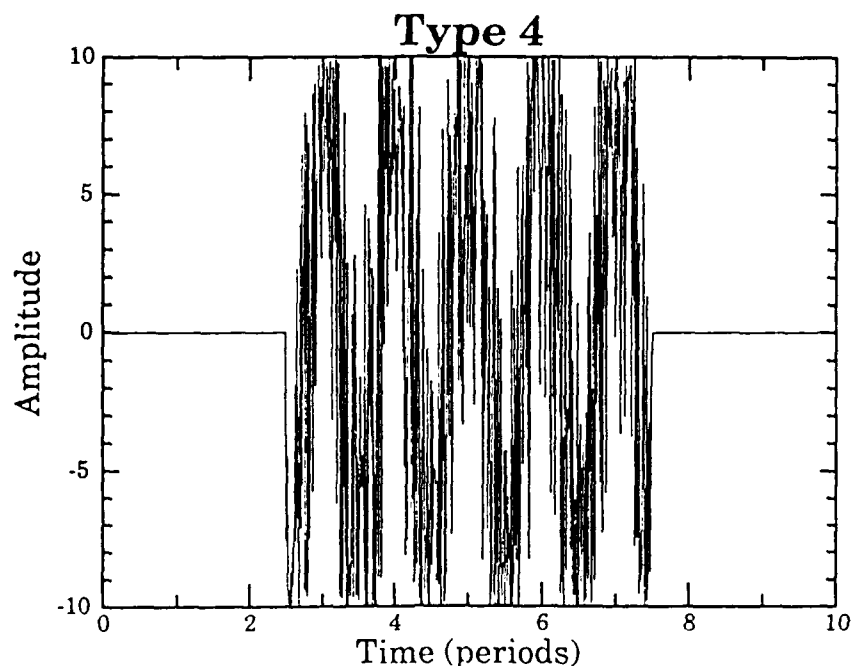


Figure 59: Type 4 trace.

The analog trace would exhibit a "flickering" effect. The trace would be zero, indicating a spherically pulsating bubble and then, with no pressure change, it would spontaneously jump into a large amplitude, seemingly random phase shape oscillation lasting from 0.5 to 2 seconds before spontaneously jumping back to a zero trace. This was quite repeatable in that once a bubble exhibited this behavior, it would continue to do so for many minutes. There did not seem, however, to be any predictability for which bubble size it would occur. Bubbles of all sizes would exhibit this behavior at some time.

The fifth type (type 5) of behavior was the second most probable. Figure 60 is a representative example.

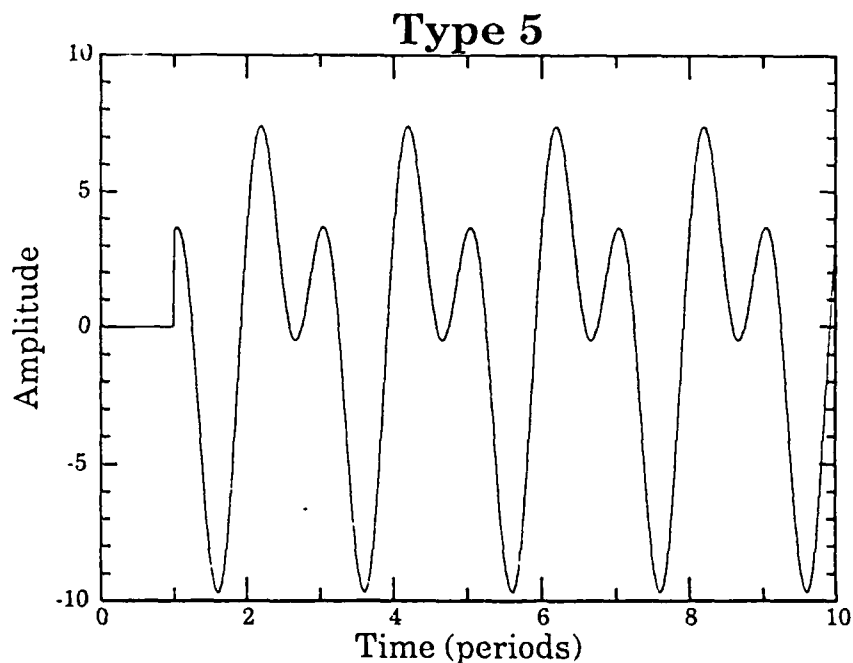


Figure 60: Type 5 trace.

Type 5 bubbles exhibited a large-amplitude, constant-phase, long-lived, stable shape oscillation which could last for up to five minutes. One reason for the longevity of this shape oscillation could be the fact that the rectified diffusion threshold was very near and possibly slightly higher than the shape oscillation threshold. Bubbles around 50 microns tended to exhibit this type of behavior more often than other bubble sizes. These types of traces basically described all of the observed shape oscillations. Most were repeatable with types 2 and 5 being the most prevalent.

Pressure Hysteresis

Another experimental observation was the evidence of a pressure hysteresis. Beginning with a spherically pulsating bubble, the pressure was increased until a shape oscillation occurred. To recover the spherical pulsation, the pressure had to be lowered well below the pressure required for the onset of the shape oscillation. During this time the bubble's equilibrium radius did not change significantly and thus could not be an explanation for this fact. A possible explanation is that there was not a sufficient number of perturbation mechanisms present to cause shape oscillations until the pressure gradient itself excited the non-radial motion. Normally, random fluctuations in the cell environment were sufficient perturbations.

Experimentally this hysteresis effect was tested by forcibly perturbing the bubble by momentarily turning off the sound field. Use of this technique did narrow the hysteresis but did not completely remove it. The average hysteresis width was about 0.05 bar. Studying this hysteretic effect led directly into the next observed effect, what we will call the "super-stability" of a bubble.

Super-stability

When the system was nearly free of perturbation sources, such as thermal currents, dirt particles, micro-bubbles, etc., then it was possible to increase the pressure exerted on a spherically pulsating bubble beyond its radial instability threshold. This fact was not discovered until late in the progress of this study. Earlier it was assumed that there would always be sufficient perturbation sources in the levitation cell to excite a surface oscillation when the threshold had been reached. To make sure that this was indeed a result of lacking perturbations, when such a bubble was observed, the data was retaken by forcibly perturbing the bubble before and near the expected threshold. In each case the bubbles which exhibited this super-stable effect would revert to normal thresholds. These effects were normally easy to observe, but there were cases where the relative stability of the bubble (or more precisely, the width of the radial instability transition region) seemed to be larger than usual.

Instability width

This transition region, can be thought of as a "broadening" of the stability transition region. Observation showed that, for a given bubble radius, the point below which the applied pressure would not be sufficient to induce a shape oscillation, even in the presence of perturbations, that would last more than two or three driving cycles, and the pressure point at which the

bubble would become radially unstable, could vary in width (ΔP). Thus, for some bubble radii, ΔP could be on the order of 0.1 bar where for other bubble radii, ΔP could be less than 0.01 bar. Future experimental studies might show that there is a correlation between this ΔP effect and radial resonances.

Beaconing effect

One of the most interesting observed phenomenon of the surface oscillations was a "beaconing" effect. When a bubble exhibited a stable or long lived (> 2 minutes) shape oscillation the bubble had a tendency to slowly precess. The precession rate was on the order of a few hertz and could be altered by slightly changing the pressure in the cell. Visually, this precession looked like a light house beacon. It was a very obvious effect and when the beacon crossed a photodiode, the output voltage increased by at least an order of magnitude. This beaconing effect was not visible in the data records obtained, since the radial pulsations were on the order of 20 kHz and the records were taken for only approximately 100 cycles. The beaconing effect on this time scale was not significant and could not be observed in the data. Also, the beacon effect normally did not occur until a few seconds after the threshold effect.

Results

The results of both methods for determining the shape oscillation thresholds are shown in Fig. 61.

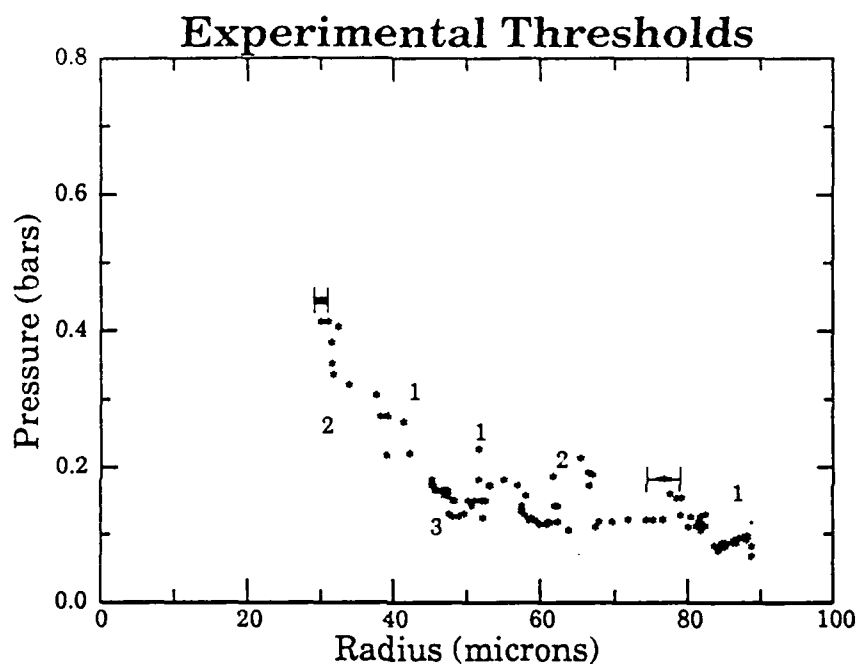


Figure 61: Experimental thresholds for radial instabilities. Representative error bars are shown. (Pressure ~1% error, Radius ~ 3% error)

The figure represents an applied acoustic pressure vs. bubble radius plot which shows the point at which the bubble exhibited non-spherical pulsations. The numbers near the threshold curve represent the frequency of the shape oscillation.

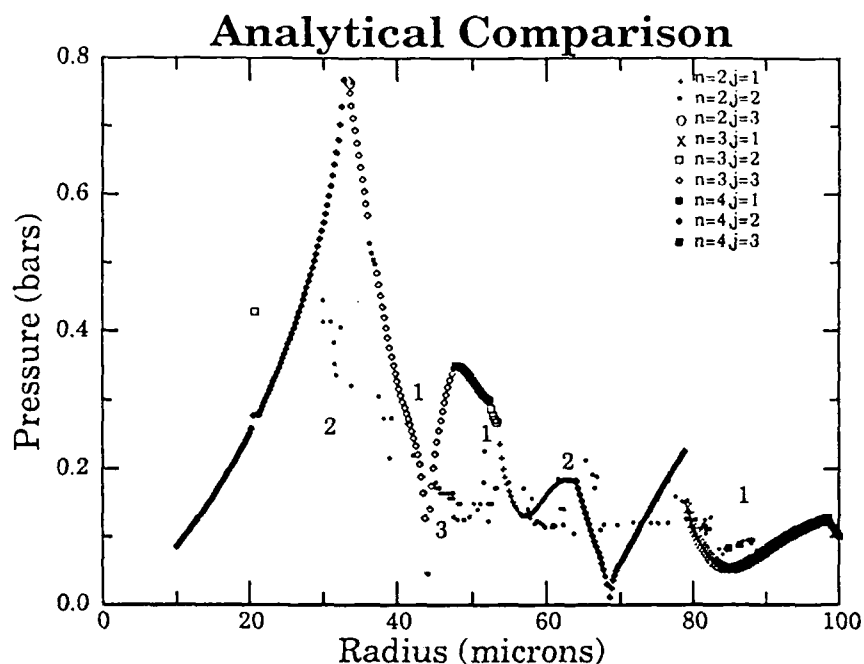


Figure 62: Experimental thresholds vs. analytic prediction.

Figure 62 shows the comparison between the experimentally obtained thresholds and the analytic solution to the theoretical prediction. Note that for bubbles larger than about 40 microns, the order of magnitude of the fit is fairly good. (It is interesting that the shape of the curve is similar but seems to be shifted in radius.) It can also be seen that the predicted frequencies for the shape oscillation match well with the experimental data. For the smaller bubbles, the analytic-solution fit is poor and could be attributed to the ignoring of surface tension terms in the analytic solution, as discussed in Chapter IV.

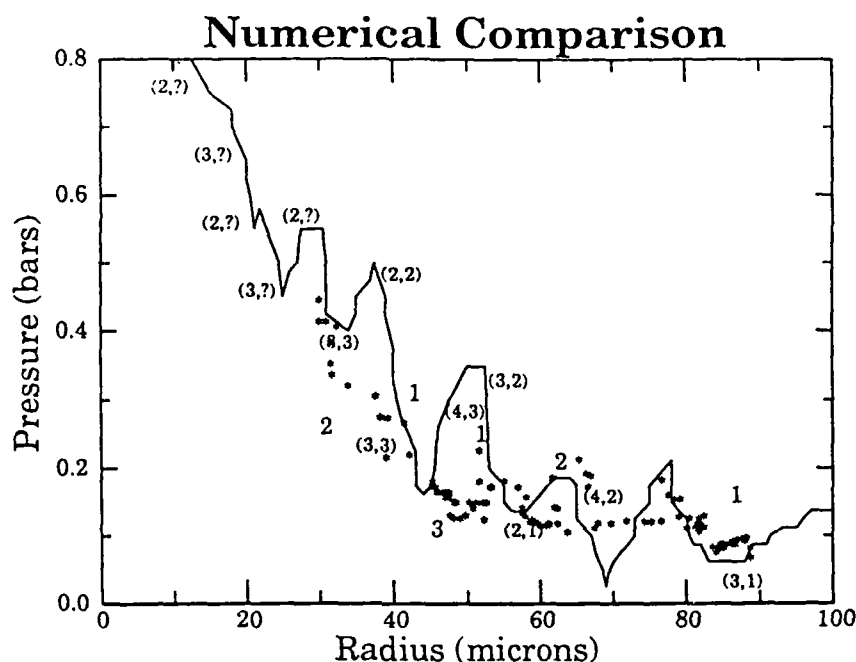


Figure 63: Experimental thresholds vs. numerical prediction.

Figure 63 shows a comparison between the experimentally-obtained thresholds and a numerical solution to the theoretical prediction. Here again the order of magnitude of the fit is quite good. Also, the general trends are similar but are shifted in radius.

Thus, we are confident that the theoretical prediction is at least qualitatively correct in determining where the thresholds for surface oscillations occur. Differences between theory and experiment could be due to assumptions made in the theory as outlined earlier.

Summary

The aim of this dissertation was to analyze the surface-wave threshold phenomenon exhibited by an acoustically driven air bubble in water. To accomplish this task, a theoretical development leading to a simplified set of equations describing the pulsating motion of an air bubble was given in early part of the dissertation.

Next, presentation and refinement of an analytical solution to the equations of motion was discussed and presented in graphical form. Then, in an attempt to improve upon the analytical solution, a direct numerical integration was performed on the equations of motion. It was shown that the numerical integration, which contained terms ignored by the analytical solution, produced a different threshold curve for the smaller bubbles.

Then, experimental results of the shape oscillation thresholds were given and compared to the the theoretical predictions. It was seen that while the theoretical results predicted general trends, there were regions where the peaks and valleys, in the graphical representations, did not exactly match. Possible explanations for these small discrepancies were presented and

discussed. The major cause, in my opinion, to the discrepancies between theory and experiment is due to the modeling of the pressure inside the bubble by a polytropic approximation. More exact theories have been recently developed to model the pressure inside the bubble using thermodynamic considerations and could be employed in the future to generate other threshold curves. This study also provided experimental verification of the theory of Mie scattering, thus affording a precise means of studying the oscillating behavior of a small (10-100 μ) bubble. Further, by using the techniques developed in this study, an extension of this study to examine the actual character of the shape oscillations, radial transients, and chaotic behavior could be more easily undertaken.

Finally, in conclusion, this study has shown that in order to obtain an adequate theoretical treatment of the surface wave threshold phenomenon, a direct integration technique, including surface tension terms, is needed. It has also been shown, in general, that the theoretical predictions of the surface wave thresholds are in general agreement with the experimental data.

Appendix A

Rectified Diffusion Growth Rates

Whenever an experiment is done, there is always extra information obtained that is not directly related to the original experiment. This experiment was no exception. When using the "first" method (described in the section on shape thresholds), growth rates of a spherically pulsating, driven air bubble in water were obtained. The mechanism whereby a pulsating bubble increases its size in a saturated liquid is known as rectified diffusion. Rectified diffusion is a process whereby the average volume of a bubble either grows by pumping air from the liquid into itself or decays by losing air to the surrounding liquid. Qualitatively, this can be described as follows: When the pressure inside the bubble is slightly less than the pressure in the surrounding liquid, then air will diffuse into the bubble. When the converse is true, air will diffuse from the bubble into the liquid. For a pulsating bubble, during the positive pressure driving cycle, the bubble is at its smallest volume, and therefore its smallest surface area. During this phase, the pressure inside the bubble is greater than the

pressure in the liquid, so air will diffuse from the bubble out into the liquid. During the negative pressure cycle, the bubble is at its maximum volume, hence its largest surface area, and the pressure inside the bubble is less than the pressure in the liquid, so air will diffuse into the bubble. Since the amount of air diffused into or out of the bubble depends on the surface area of the bubble, it is seen that the bubble will gain more air than it loses over each cycle (assuming the diffusion rate to be constant). The driving pressure threshold for this effect to occur is known as the growth by rectified diffusion threshold. The rate at which the bubble grows is known as the rectified diffusion growth rate.

We were able to measure the rectified diffusion growth rate of a variety of bubbles. Figures A1, A2, A3, and A4 show the average radii of four different bubbles growing by rectified diffusion as a function of time. The asterisks represent the data and the two curves represent two different rectified diffusion theories (Crum and Hansen, 1982, and Church, 1989). As the actual theories will not be explained here, the interested reader is directed to the references. The Crum and Hansen theory is a linearized theory while the theory of Church includes nonlinear effects such as streaming. The figures show that for slow growth rates (i.e., near the rectified diffusion growth threshold) and low pressures, both the linear and the more general theory predict the experimental results. For higher pressures and for faster growth rates, however, the linear theory works only when the bubble is not near a radial resonance. Church's theory, with

or without considering the streaming phenomenon, more precisely predicts the experimental results. These results are simply presented here. A more detailed analysis may be made in a later study.

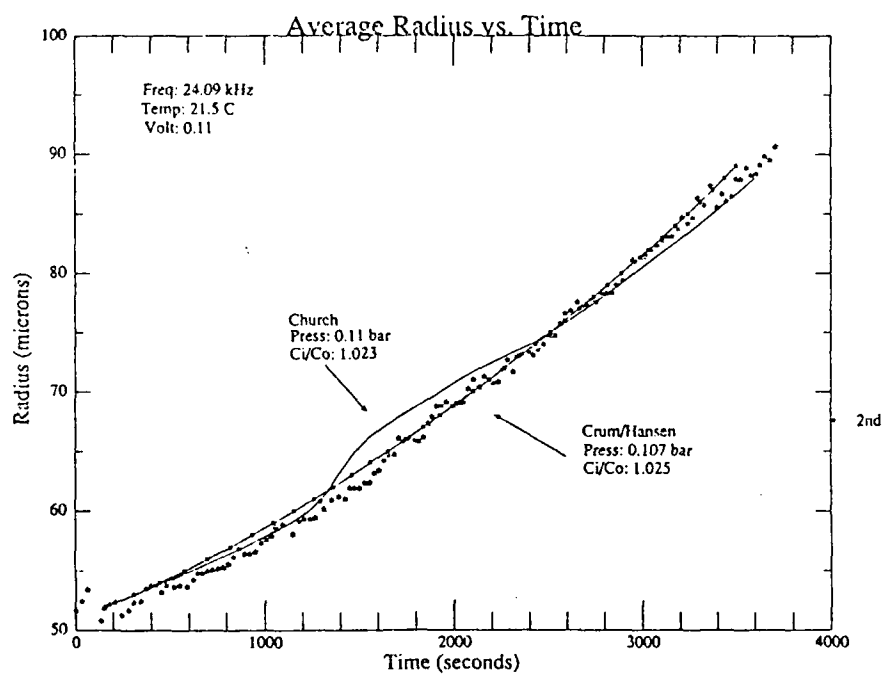


Figure A1: Rectified diffusion case #1

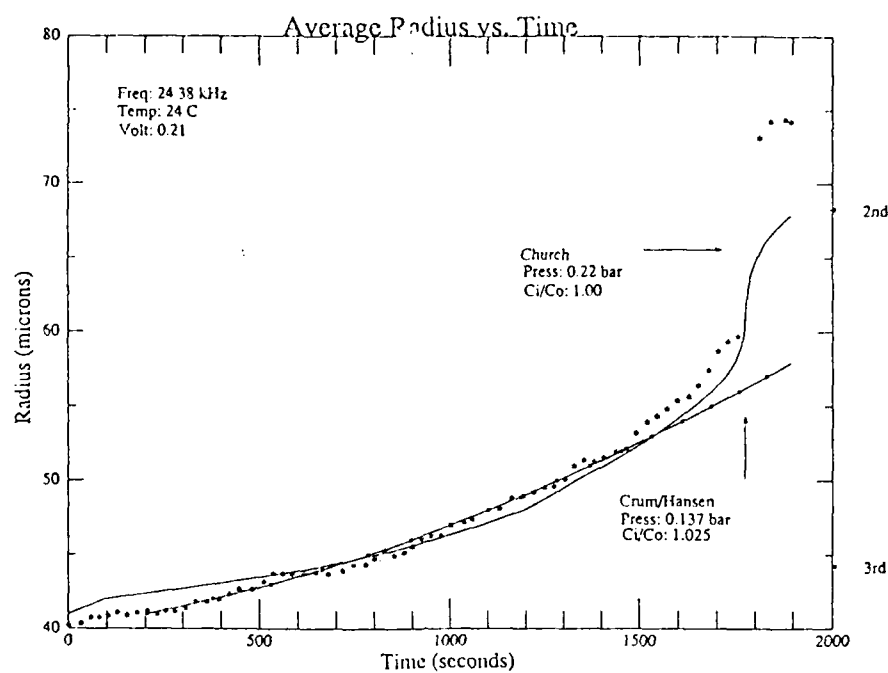


Figure A2: Rectified diffusion case #2

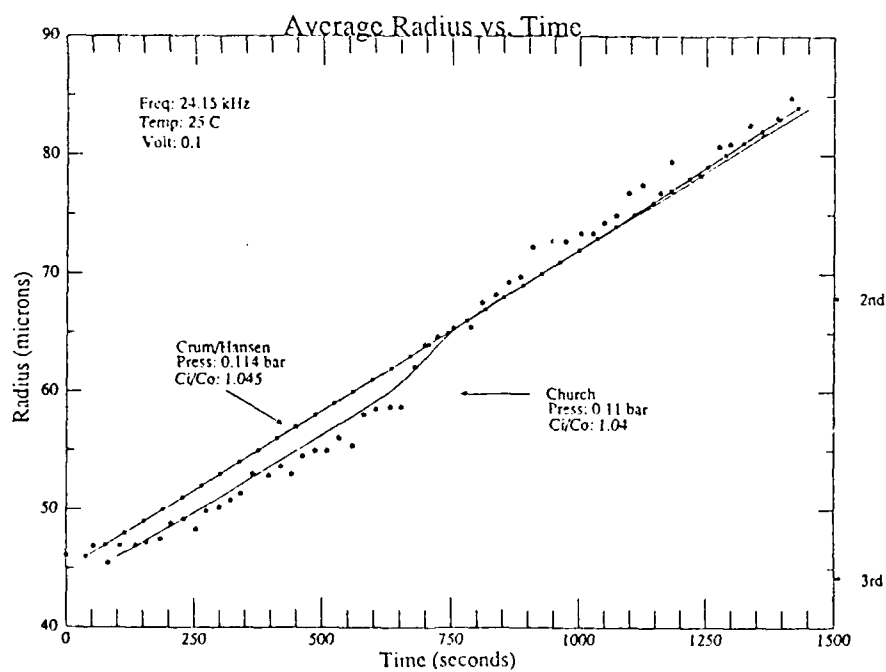


Figure A3: Rectified diffusion case #3

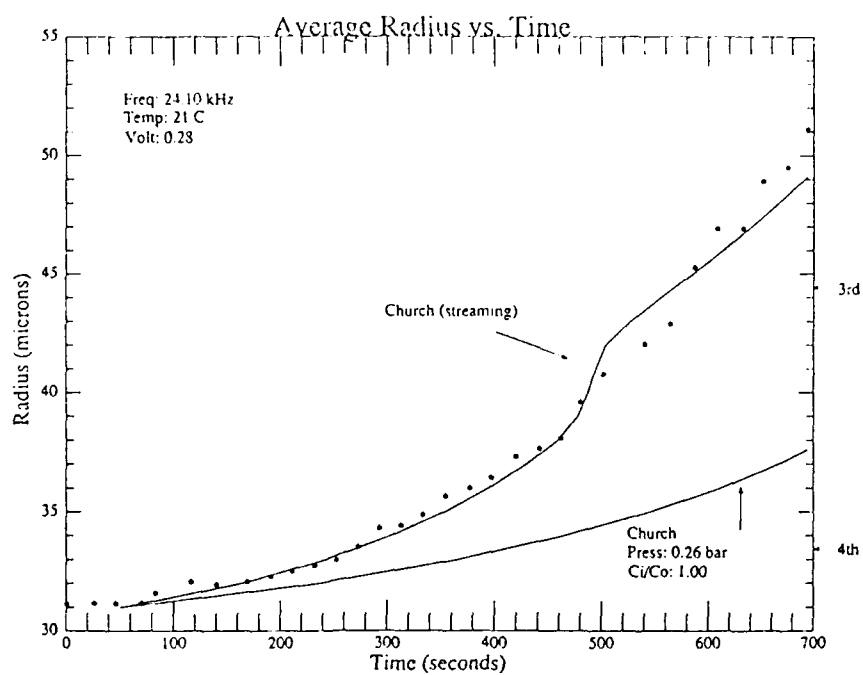


Figure A4: Rectified diffusion case #4

Appendix B

In this appendix, a derivation of the scattering amplitudes is given utilizing Mie theory. This theory deals with the scattering of a plane electromagnetic wave by an isotropic, homogeneous sphere of arbitrary size. Briefly, the theory may be derived as follows.

Beginning with Maxwell's equations for a plane wave in a homogeneous, isotropic medium (water) impinging on a non-conducting sphere (bubble) with no currents or free charges, one may write the equations as

$$\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = 0 \quad (\text{B.1})$$

$$\nabla \times \mathbf{H} - \frac{\partial \mathbf{D}}{\partial t} = 0 \quad (\text{B.2})$$

$$\nabla \cdot \mathbf{D} = 0 \quad (\text{B.3})$$

$$\nabla \cdot \mathbf{B} = 0 \quad (\text{B.4})$$

where $\mathbf{D} = \epsilon \mathbf{E}$ and $\mathbf{B} = \mu \mathbf{H}$.

Since these equations can be written in terms of a vector potential, $\mathbf{A}$, and a scalar potential, ϕ , if we assume that $\mathbf{A}$ is proportional to a time-varying potential (named the electric Hertz vector, Π_1)

$$\mathbf{A} = \mu\epsilon \frac{\partial \Pi_1}{\partial t} \quad (\text{B.6})$$

we find that

$$\mathbf{B} = \mu\epsilon \nabla \times \frac{\partial \Pi_1}{\partial t} \quad (\text{B.7})$$

and

$$\mathbf{E} = -\nabla\phi - \mu\epsilon \frac{\partial^2 \Pi_1}{\partial t^2} . \quad (\text{B.8})$$

Since ϕ must satisfy the wave equation, $\phi = -\nabla \cdot \Pi_1$. An alternate solution, Π_2 , (named the magnetic Hertz vector) to Eq. (B.6) may be chosen as well, so that the vector quantities can be written as

$$\mathbf{B}_1 = \mu\epsilon \nabla \times \frac{\partial \Pi_1}{\partial t} \quad (\text{B.9})$$

$$\mathbf{E}_1 = \nabla(\nabla \cdot \Pi_1) - \mu\epsilon \frac{\partial^2 \Pi_1}{\partial t^2} \quad (\text{B.10})$$

$$\mathbf{D}_2 = -\mu\epsilon \nabla \times \frac{\partial \Pi_2}{\partial t} \quad (\text{B.11})$$

$$\mathbf{H}_2 = \nabla(\nabla \cdot \Pi_2) - \mu\epsilon \frac{\partial^2 \Pi_2}{\partial t^2} \quad (\text{B.12})$$

Since $\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P}$, the wave equations for Π_1 and Π_2 can be written as

$$\nabla^2 \Pi_1 - \mu\epsilon_0 \frac{\partial^2 \Pi_1}{\partial t^2} = -\frac{\mathbf{P}}{\epsilon_0} \quad (\text{B.13})$$

$$\nabla^2 \Pi_2 - \mu_0\epsilon \frac{\partial^2 \Pi_2}{\partial t^2} = -\mathbf{M} \quad (\text{B.14})$$

where $\mathbf{M}$ is the magnetization and $\mathbf{P}$ is the polarization.

The vector potentials can now be written in terms of scalar potentials (referred to as the Hertz-Debye potentials) such that

$$\Pi_1 = -\nabla \cdot \Pi_1 \quad (\text{B.15})$$

$$\Pi_2 = -\nabla \cdot \Pi_2 \quad (\text{B.16})$$

Equations (B.15) and (B.16) are solutions to the scalar wave equation

$$\nabla^2 \Pi - \mu\epsilon \frac{\partial^2 \Pi}{\partial t^2} = 0 \quad (\text{B.17})$$

In terms of these scalar potentials, the components of the field vectors in spherical coordinates are given by (Kerker, 1969)

$$E_r = \frac{\partial^2(r\Pi_1)}{\partial r^2} + k^2 r \Pi_1 \quad (\text{B.18})$$

$$E_\theta = \frac{\partial^2(r\Pi_1)}{r \partial r \partial \theta} + \frac{\kappa_2}{r \sin \theta} \frac{\partial(r\Pi_2)}{\partial \phi} \quad (\text{B.19})$$

$$E_\phi = \frac{1}{r \sin \theta} \frac{\partial^2(r\Pi_1)}{\partial r \partial \phi} - \frac{\kappa_2}{r} \frac{\partial(r\Pi_2)}{\partial \theta} \quad (\text{B.20})$$

and

$$H_r = \frac{\partial^2(r\Pi_2)}{\partial r^2} + k^2 r \Pi_2 \quad (\text{B.21})$$

$$H_\theta = -\frac{\kappa_1}{r \sin \theta} \frac{\partial(r\Pi_1)}{\partial \phi} + \frac{1}{r} \frac{\partial^2(r\Pi_2)}{\partial r \partial \theta} \quad (\text{B.22})$$

$$H_\phi = \frac{\kappa_1}{r} \frac{\partial(r\Pi_1)}{\partial \theta} + \frac{1}{r \sin \theta} \frac{\partial^2(r\Pi_2)}{\partial r \partial \phi} \quad (\text{B.23})$$

where the propagation constant is given by

$$k^2 = -\kappa_1 \kappa_2 = \omega^2 \epsilon \quad (\text{B.24})$$

with

$$\kappa_1 = i\omega\epsilon \quad (\text{B.25})$$

$$\kappa_2 = i\omega. \quad (\text{B.26})$$

Solution to the scalar wave equation

If we assume a sinusoidal time dependence of $e^{i\omega t}$ in Eq. (B.17), then the scalar wave equation becomes

$$\nabla^2 \pi + k^2 \pi = 0 \quad (\text{B.27})$$

where

$$\Pi = \pi e^{i\omega t} . \quad (\text{B.28})$$

Considering that the potential π may be considered a product of three coordinate functions, π may be expressed in spherical coordinates by

$$\pi = R(r)\Theta(\theta)\Phi(\phi) . \quad (\text{B.29})$$

The well-known functional forms of Eq. (B.29) are

$$\frac{d^2 r R(r)}{dr^2} + \left[k^2 - \frac{n(n+1)}{r^2} \right] r R(r) = 0 \quad (\text{B.30})$$

$$\frac{1}{\sin \theta} \frac{d}{d\theta} \left(\sin(\theta) \frac{d\Theta(\theta)}{d\theta} \right) + \left[n(n+1) - \frac{m^2}{\sin^2 \theta} \right] \Theta(\theta) = 0 \quad (\text{B.31})$$

$$\frac{d^2 \Phi(\phi)}{d\phi^2} + m^2 \Phi(\phi) = 0 \quad (\text{B.32})$$

where n is an integer and m ranges from $-n$ to n . The solutions to Eq. (B.30) are the Ricatti-Bessel functions:

$$\psi_n(kr) = (\pi kr/2)^{1/2} J_{n+1/2}(kr) \quad (\text{B.33})$$

$$\chi_n(kr) = -(\pi kr/2)^{1/2} N_{n+1/2}(kr) \quad (\text{B.34})$$

where $J_{n+1/2}(kr)$ and $N_{n+1/2}(kr)$ are the half-integral order Bessel and Neumann functions. Note that a linear combination of these solutions can be written as

$$\zeta_n(kr) = \psi_n(kr) + i\chi_n(kr) = (\pi kr/2)^{1/2} H_{n+1/2}^{(2)}(kr) \quad (\text{B.35})$$

where $H_{n+1/2}^{(2)}(kr)$ is the half-integral order Hankel function. This will be used later. The solutions to Eq. (B.31) are the associated Legendre polynomials given by

$$\Theta(\theta) = P_n^{(m)}(\cos\theta). \quad (\text{B.36})$$

The solutions to Eq. (B.32) are $\sin(m\phi)$ and $\cos(m\phi)$.

The general solution to the scalar wave Eq. (B.17) in spherical coordinates is simply the superposition of all the particular solutions:

$$r\pi = r \sum_{n=0}^{\infty} \sum_{m=-n}^n \pi_n^{(m)} \quad (\text{B.37})$$

$$= r \sum_{n=0}^{\infty} \sum_{m=-n}^n \left\{ [c_n \psi_n(kr) + d_n \chi_n(kr)] P_n^{(m)}(\cos\theta) [a_n \cos(m\phi) + b_n \sin(m\phi)] \right\}.$$

At the bubble/liquid boundary there is an incident field ($\mathbf{E}^{(i)}, \mathbf{H}^{(i)}$), an internal field within the bubble ($\mathbf{E}^{(w)}, \mathbf{H}^{(w)}$) and a scattered field ($\mathbf{E}^{(s)}, \mathbf{H}^{(s)}$). Thus we must have corresponding expressions for π_i : $\pi_1^{(i)}, \pi_2^{(i)}, \pi_1^{(w)}, \pi_2^{(w)}$, and $\pi_1^{(s)}, \pi_2^{(s)}$. The boundary conditions are thus expressed by

$$\frac{\partial}{\partial r} [r(\pi_1^{(i)} + \pi_1^{(s)})] = \frac{\partial}{\partial r} [r\pi_1^{(w)}] \quad (\text{B.38})$$

$$\frac{\partial}{\partial r} [r(\pi_2^{(i)} + \pi_2^{(s)})] = \frac{\partial}{\partial r} [r\pi_2^{(w)}] \quad (\text{B.39})$$

$$im_2^2 k_0 r (\pi_1^{(i)} + \pi_1^{(s)}) = im_1^2 k_0 r \pi_1^{(w)} \quad (\text{B.40})$$

$$im_2^2 k_0 r (\pi_2^{(i)} + \pi_2^{(s)}) = im_2^2 k_0 r \pi_2^{(w)} \quad (\text{B.41})$$

where $k_0 = 2\pi/\lambda_0$ is the propagation constant in free space and m_1 and m_2 are the indices of refraction of the liquid and the air. Equation (B.37) determines the constants in the boundary conditions. The potentials π_i are found to be

$$r\pi_1^{(i)} = \frac{1}{k_2^2} \sum_{n=1}^{\infty} i^{n-1} \frac{2n+1}{n(n+1)} \psi_n(k_2 r) P_n^{(l)}(\cos\theta) \cos\phi \quad (\text{B.42})$$

$$r\pi_2^{(i)} = \frac{1}{k_2^2} \sum_{n=1}^{\infty} i^{n-1} \frac{2n+1}{n(n+1)} \psi_n(k_2 r) P_n^{(1)}(\cos\theta) \sin\phi \quad (\text{B.43})$$

$$r\pi_1^{(s)} = \frac{1}{k_2^2} \sum_{n=1}^{\infty} i^{n-1} \frac{2n+1}{n(n+1)} a_n \zeta_n(k_2 r) P_n^{(1)}(\cos\theta) \cos\phi \quad (\text{B.44})$$

$$r\pi_2^{(s)} = \frac{1}{k_2^2} \sum_{n=1}^{\infty} i^{n-1} \frac{2n+1}{n(n+1)} b_n \zeta_n(k_2 r) P_n^{(1)}(\cos\theta) \sin\phi \quad (\text{B.45})$$

$$r\pi_1^{(w)} = \frac{1}{k_1^2} \sum_{n=1}^{\infty} i^{n-1} \frac{2n+1}{n(n+1)} c_n \psi_n(k_1 r) P_n^{(1)}(\cos\theta) \cos\phi \quad (\text{B.46})$$

$$r\pi_2^{(w)} = \frac{1}{k_1^2} \sum_{n=1}^{\infty} i^{n-1} \frac{2n+1}{n(n+1)} d_n \psi_n(k_1 r) P_n^{(1)}(\cos\theta) \sin\phi. \quad (\text{B.47})$$

Given that the terms in the series are independent of each other, the boundary conditions must hold for each corresponding term. This gives four linear equations for a_n , b_n , c_n , and d_n at the spherical surface, $r=a$:

$$m[(k_2 a) - a_n \zeta_n'(k_2 a)] = c_n(\psi_n'(k_1 a)) \quad (\text{B.48})$$

$$m[\psi_n'(k_2 a) - b_n \zeta_n'(k_2 a)] = d_n(\psi_n'(k_1 a)) \quad (\text{B.49})$$

$$\psi_n(k_2 a) - a_n \zeta_n(k_2 a) = c_n \zeta_n(k_1 a) \quad (\text{B.50})$$

$$m^2[\psi_n(k_2 a) - b_n \zeta_n(k_2 a)] = d_n \psi_n(k_1 a) \quad (\text{B.51})$$

Only a_n and b_n are of importance since we want only the scattering wave coefficients.

$$a_n = \frac{\psi_n(\alpha)\psi_n'(\beta) - m\psi_n(\beta)\psi_n'(\alpha)}{\zeta_n(\alpha)\psi_n'(\beta) - m\psi_n(\beta)\zeta_n'(\alpha)} \quad (\text{B.52})$$

$$b_n = \frac{m\psi_n(\alpha)\psi_n'(\beta) - \psi_n(\beta)\psi_n'(\alpha)}{m\zeta_n(\alpha)\psi_n'(\beta) - \psi_n(\beta)\zeta_n'(\alpha)} \quad (\text{B.53})$$

where $m = k_1/k_2$, $\alpha = k_2a = 2\pi m_2a/\lambda_0$, $\beta = k_1a = m\alpha$ and λ_0 is the wavelength of the incident light in vacuum.

Far-field assumption

If we now make a far-field assumption so that $k_2r \ll n$, then the radial components of the fields fall off as (λ/r^2) and can be neglected so that

$$E_\theta = -\frac{H_\phi}{m_2} = \frac{ie^{-ik_2r}}{k_2r} S_2 \cos\phi \quad (\text{B.54})$$

$$E_\phi = \frac{H_\theta}{m_2} = -\frac{ie^{-ik_2r}}{k_2r} S_1 \sin\phi. \quad (\text{B.55})$$

The amplitude functions S_1 and S_2 are given by

$$S_1 = \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} (a_n \pi_n(\cos\theta) + b_n \tau_n(\cos\theta)) \quad (\text{B.56})$$

$$S_2 = \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} (a_n \tau_n(\cos\theta) + b_n \pi_n(\cos\theta)) \quad (\text{B.57})$$

where π_n and τ_n are given by

$$\pi_n(\cos\theta) = \frac{P_n^{(1)}(\cos\theta)}{\sin\theta} \quad (\text{B.58})$$

$$\tau_n(\cos\theta) = \frac{d}{d\theta} P_n^{(1)}(\cos\theta) . \quad (\text{B.59})$$

These amplitude functions S_1 and S_2 , are used in Chapter VI.

Appendix C

ADDITIONAL FIGURES

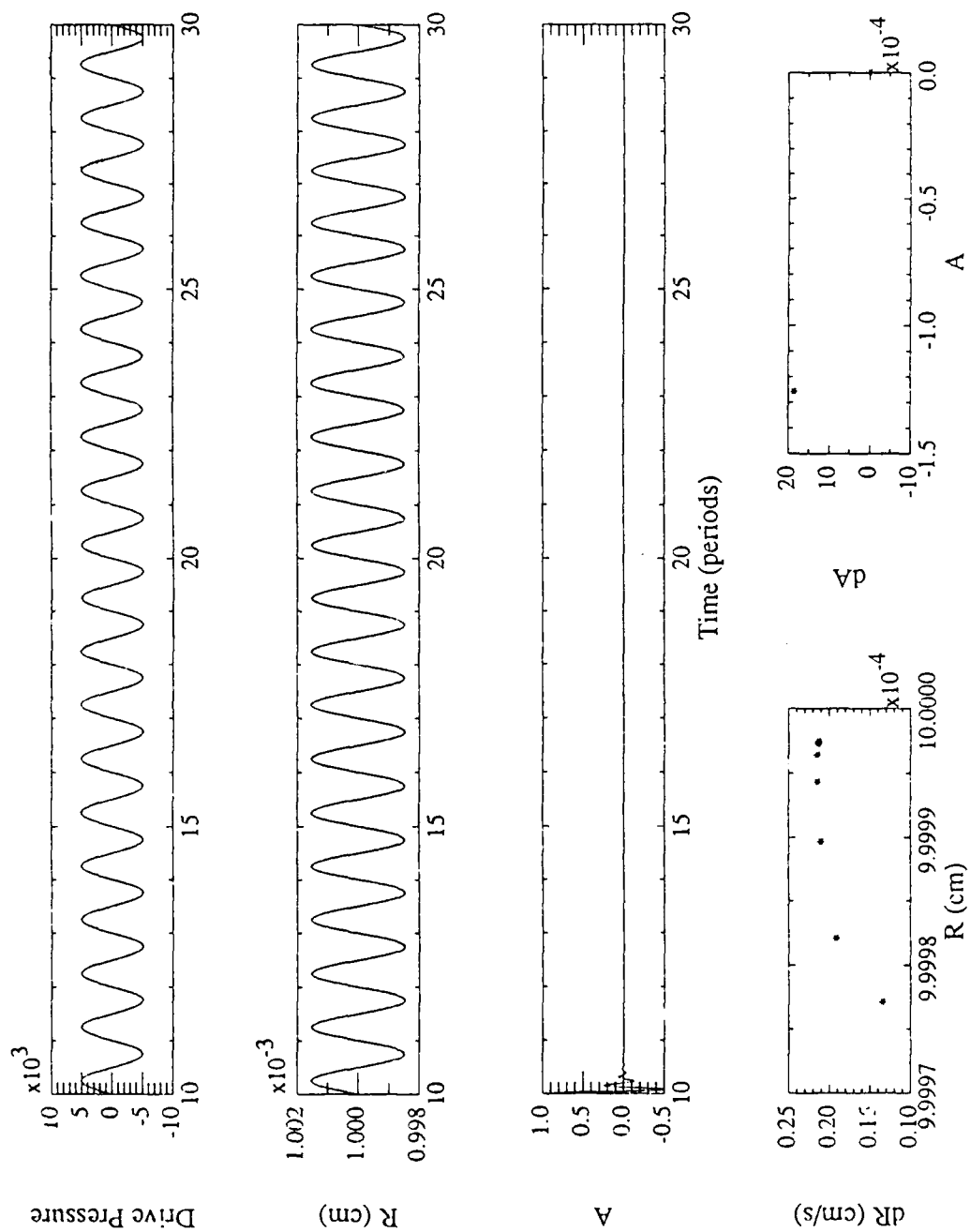


Figure C1: SOE_005_010

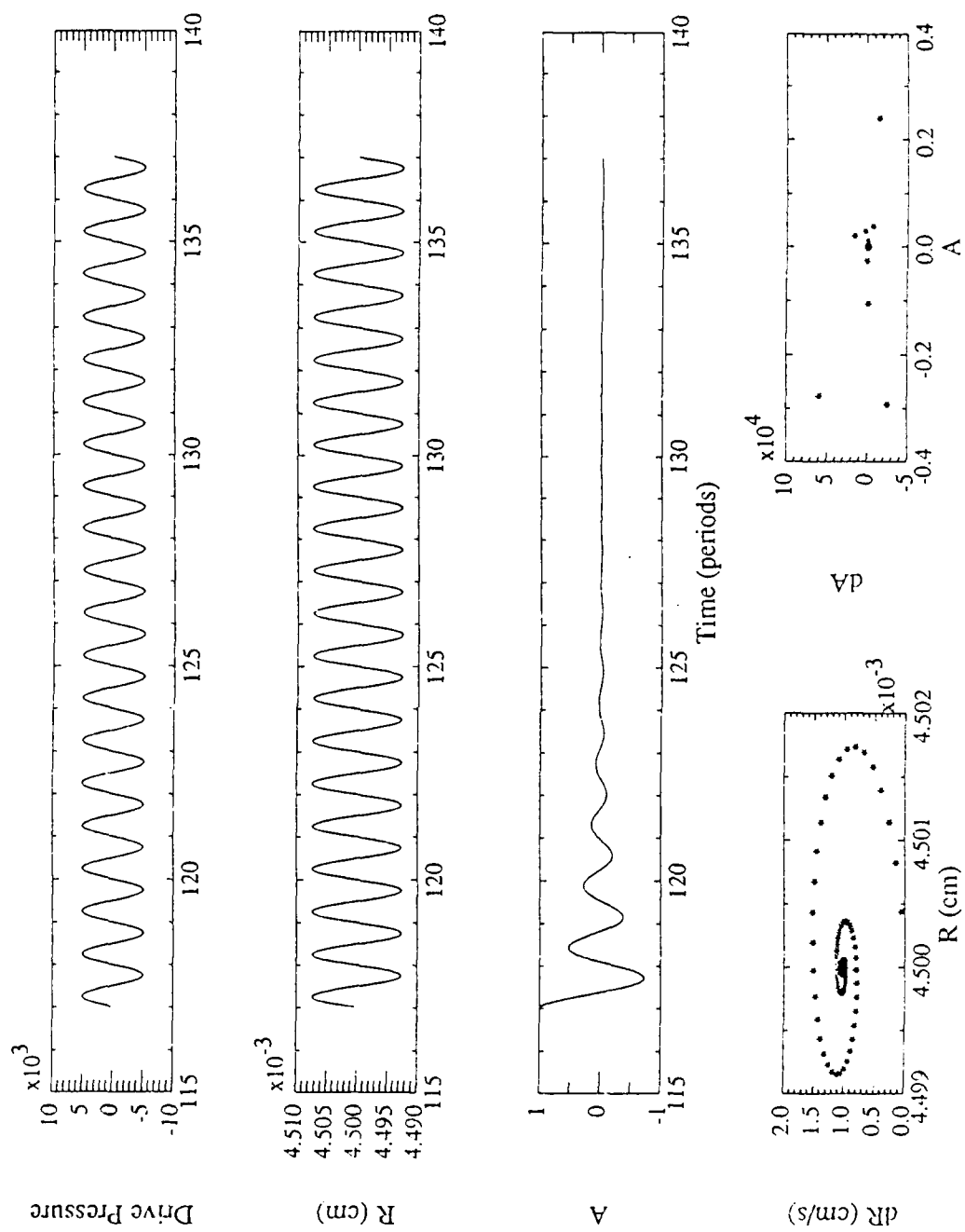


Figure C2: SOE_005_045

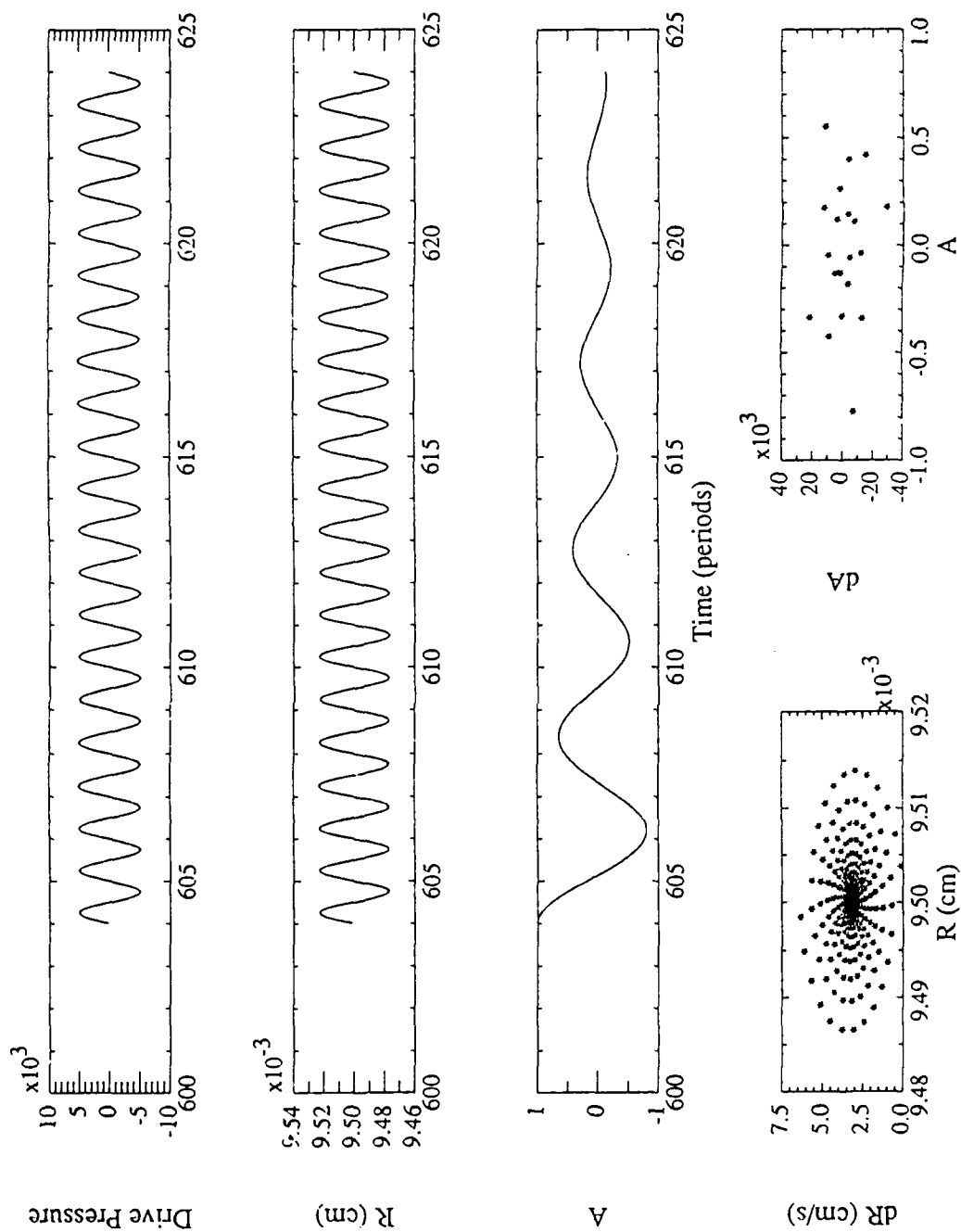


Figure C3: SOE_005_095

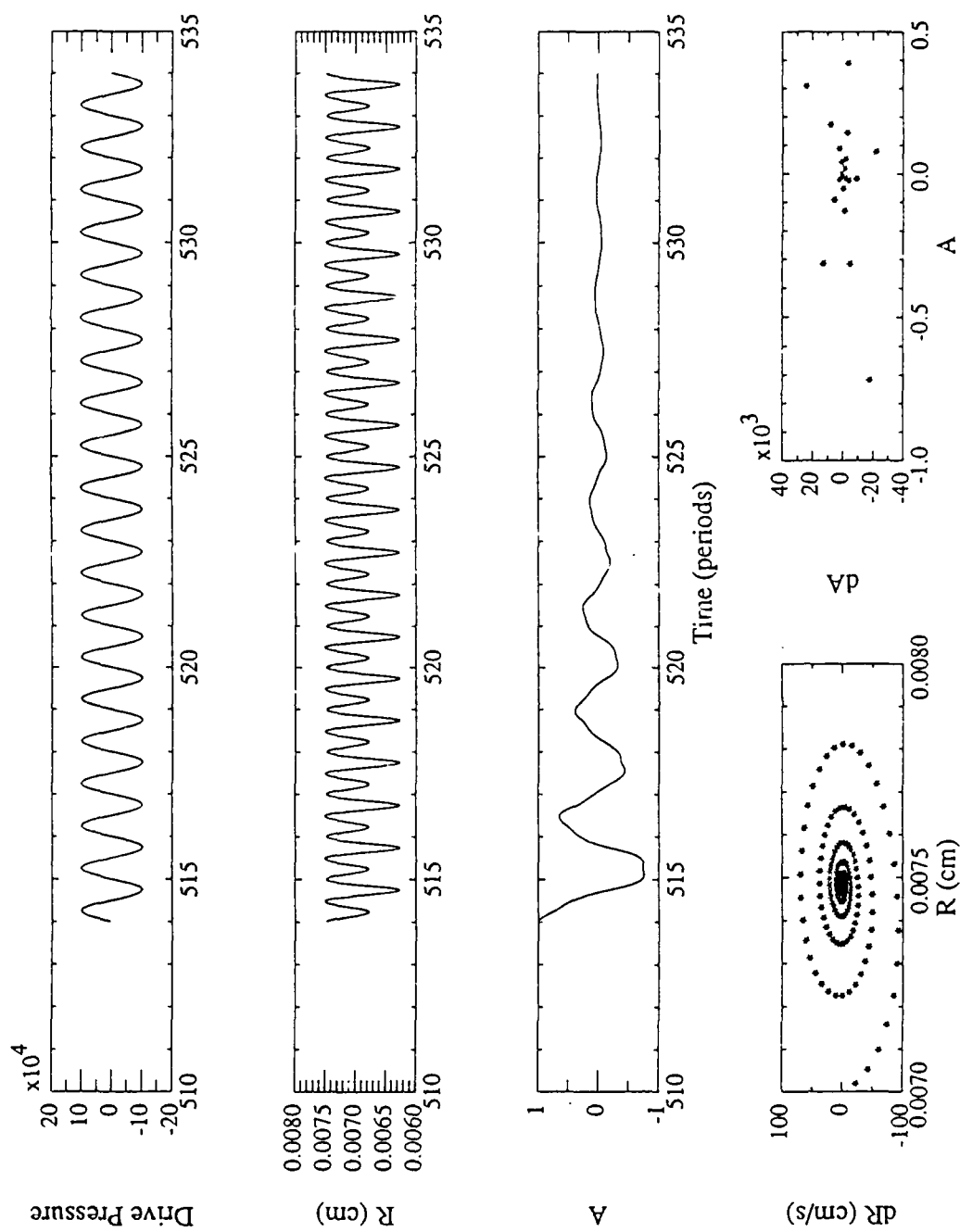


Figure C4: SOE_100_070

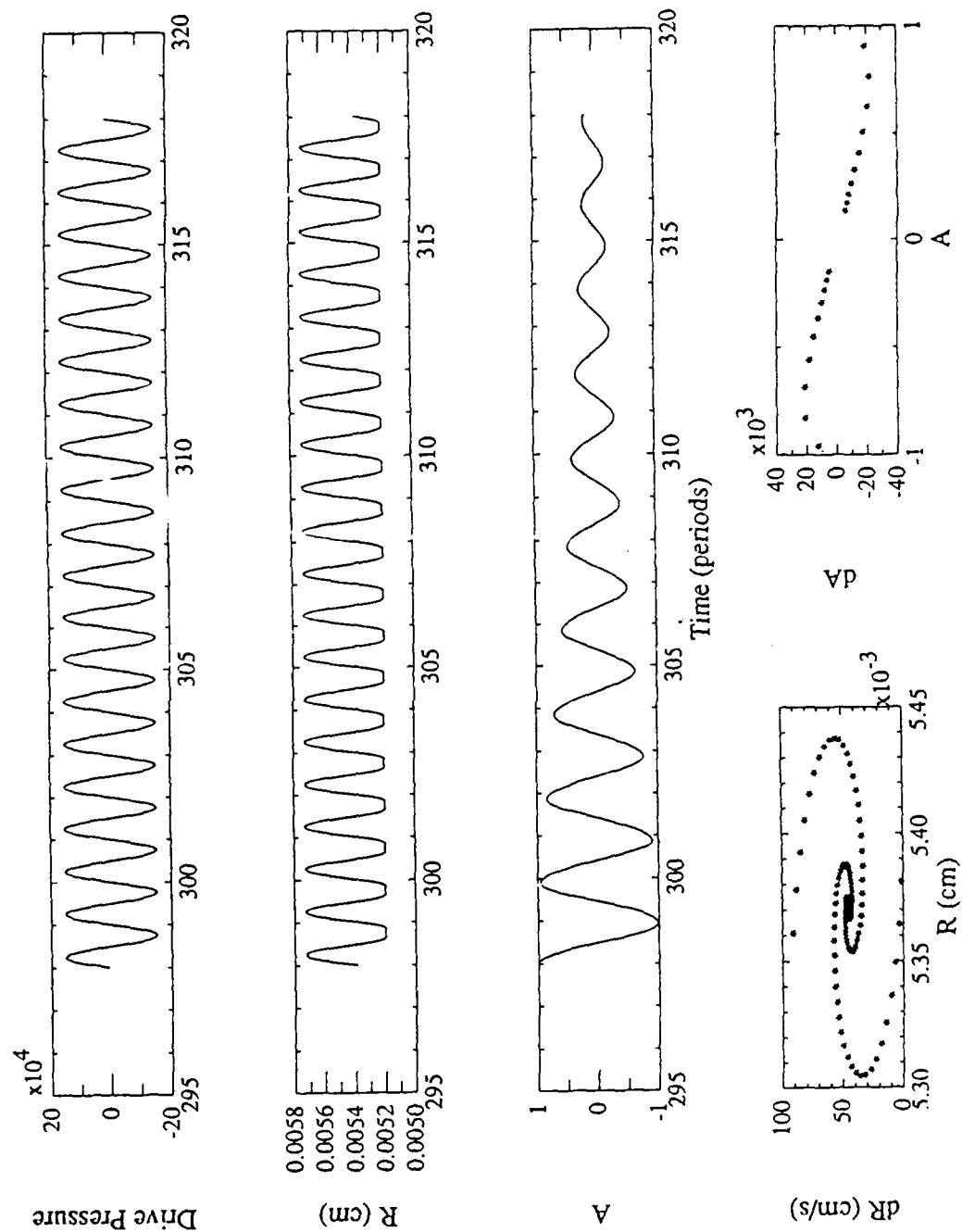


Figure C5: SOE_150_054

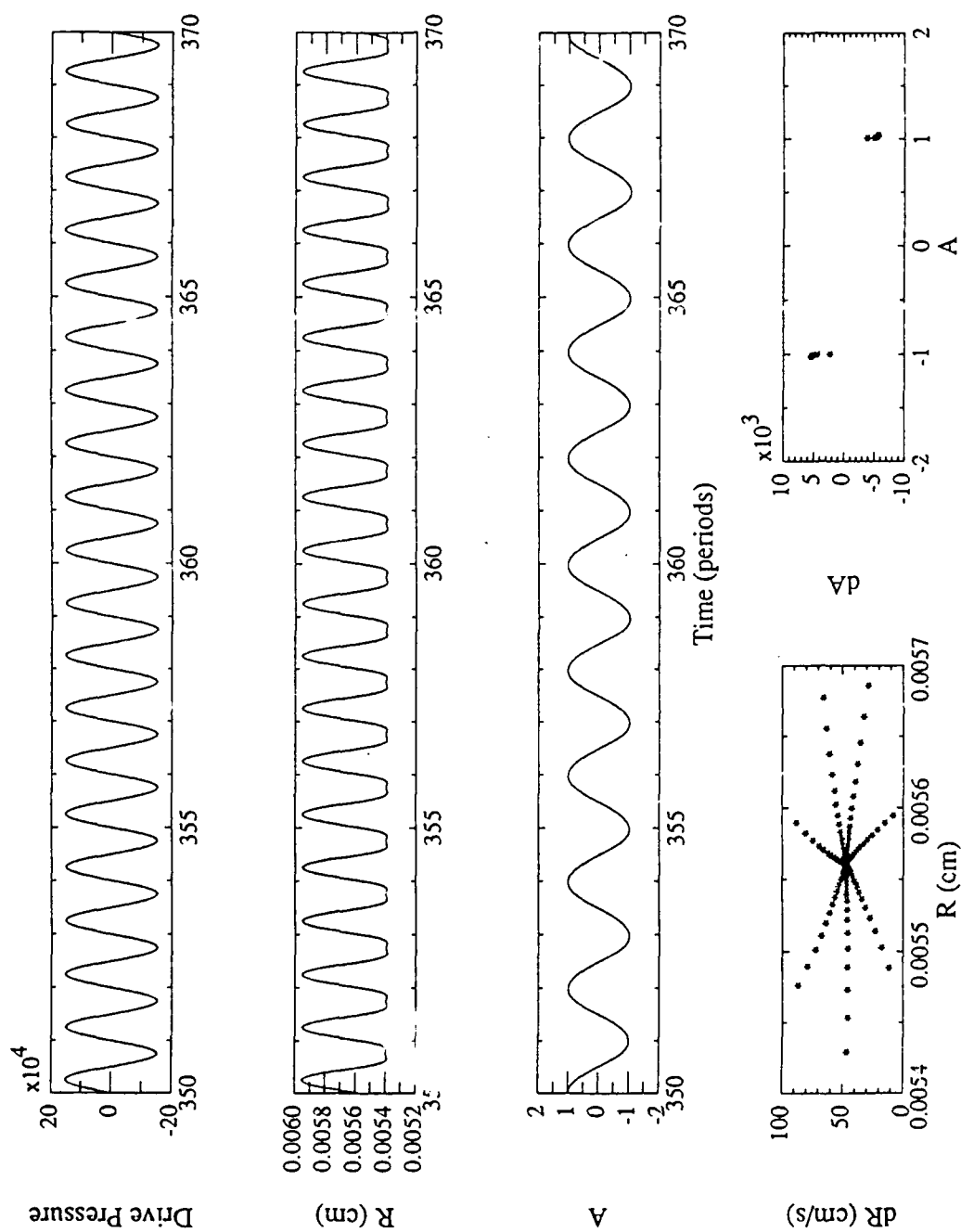


Figure C6: SOE_150_056

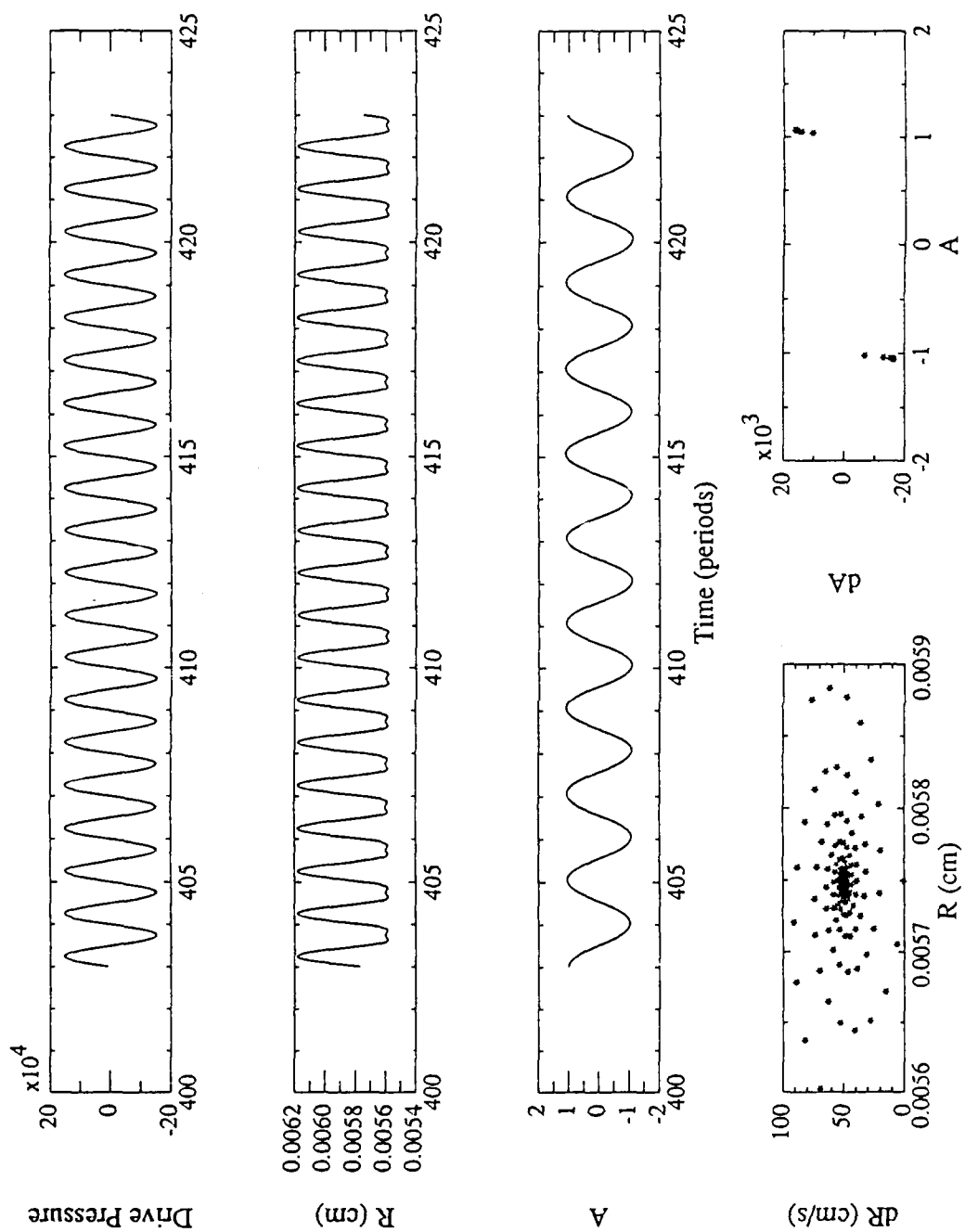


Figure C7: SOE_150_058

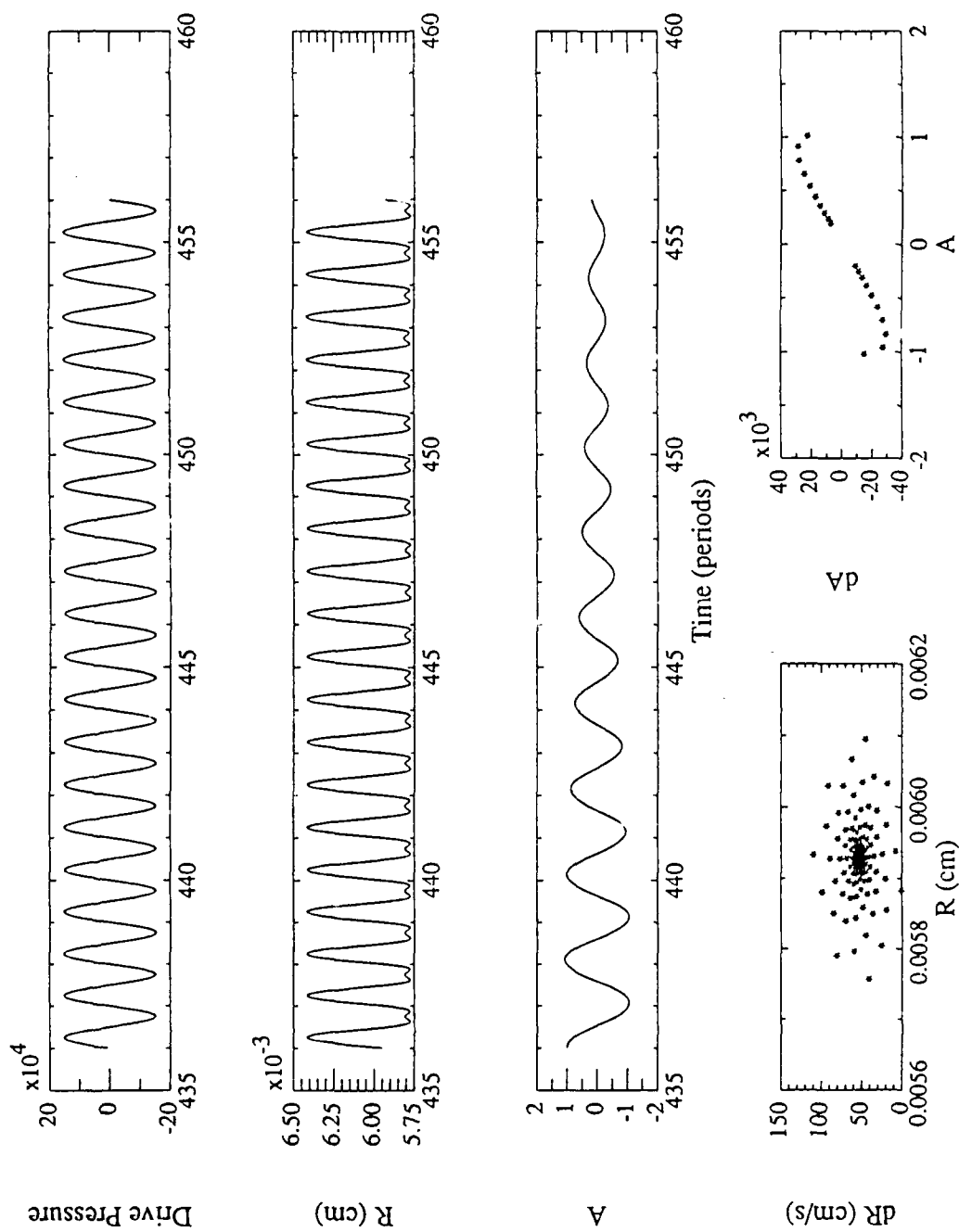


Figure C8: SOE_150_060

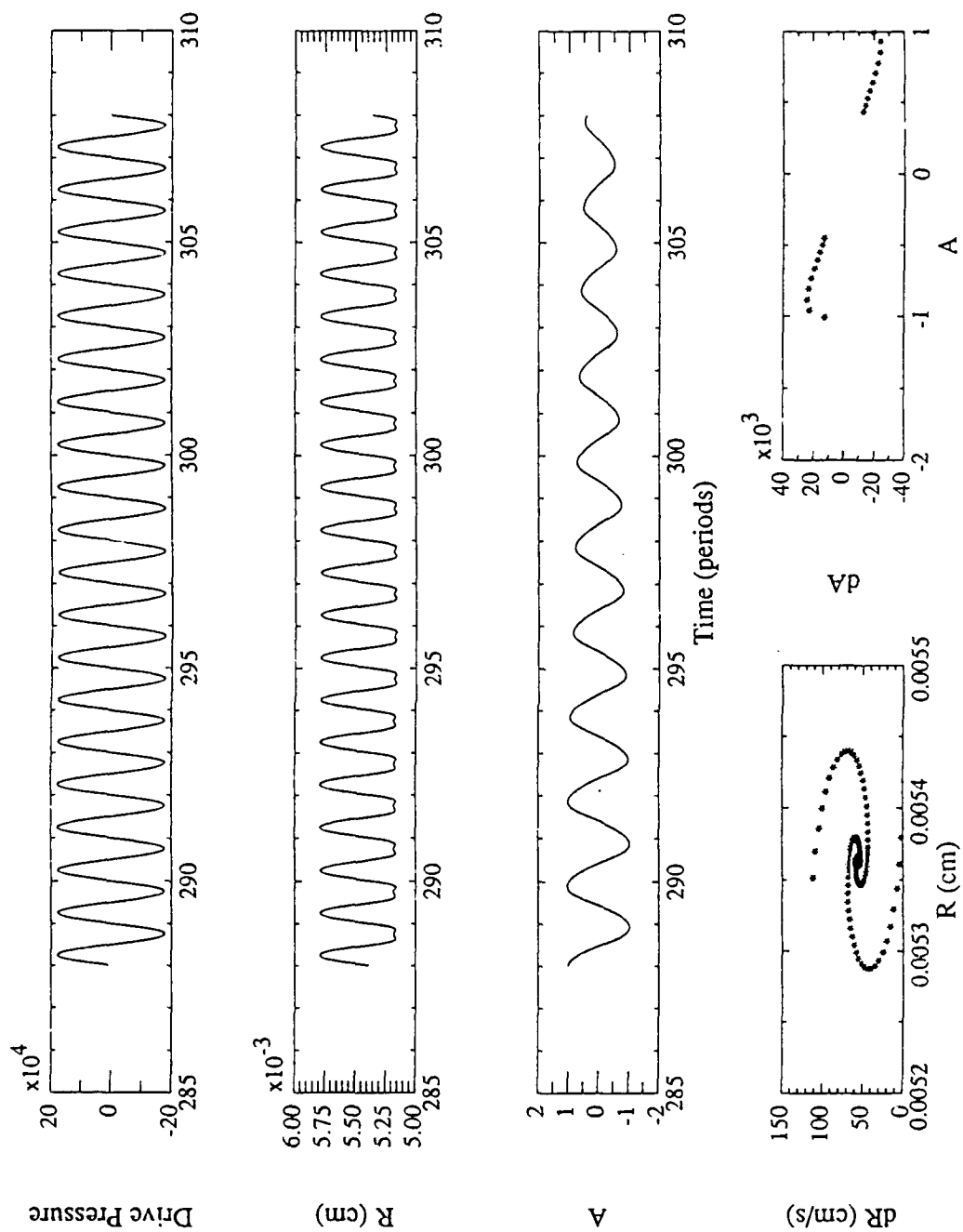


Figure C9: SOE_175_054

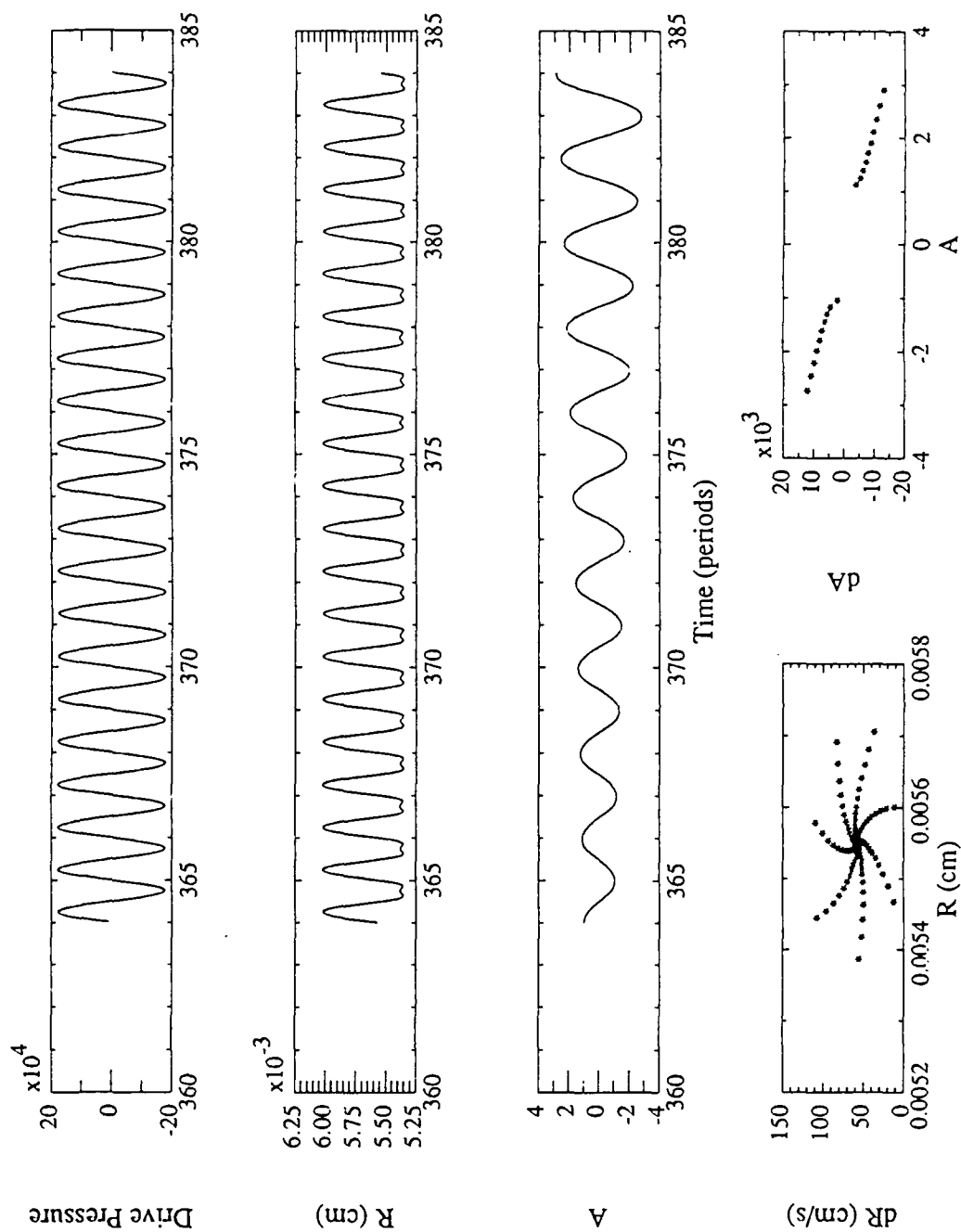


Figure C10: SOE_175_056

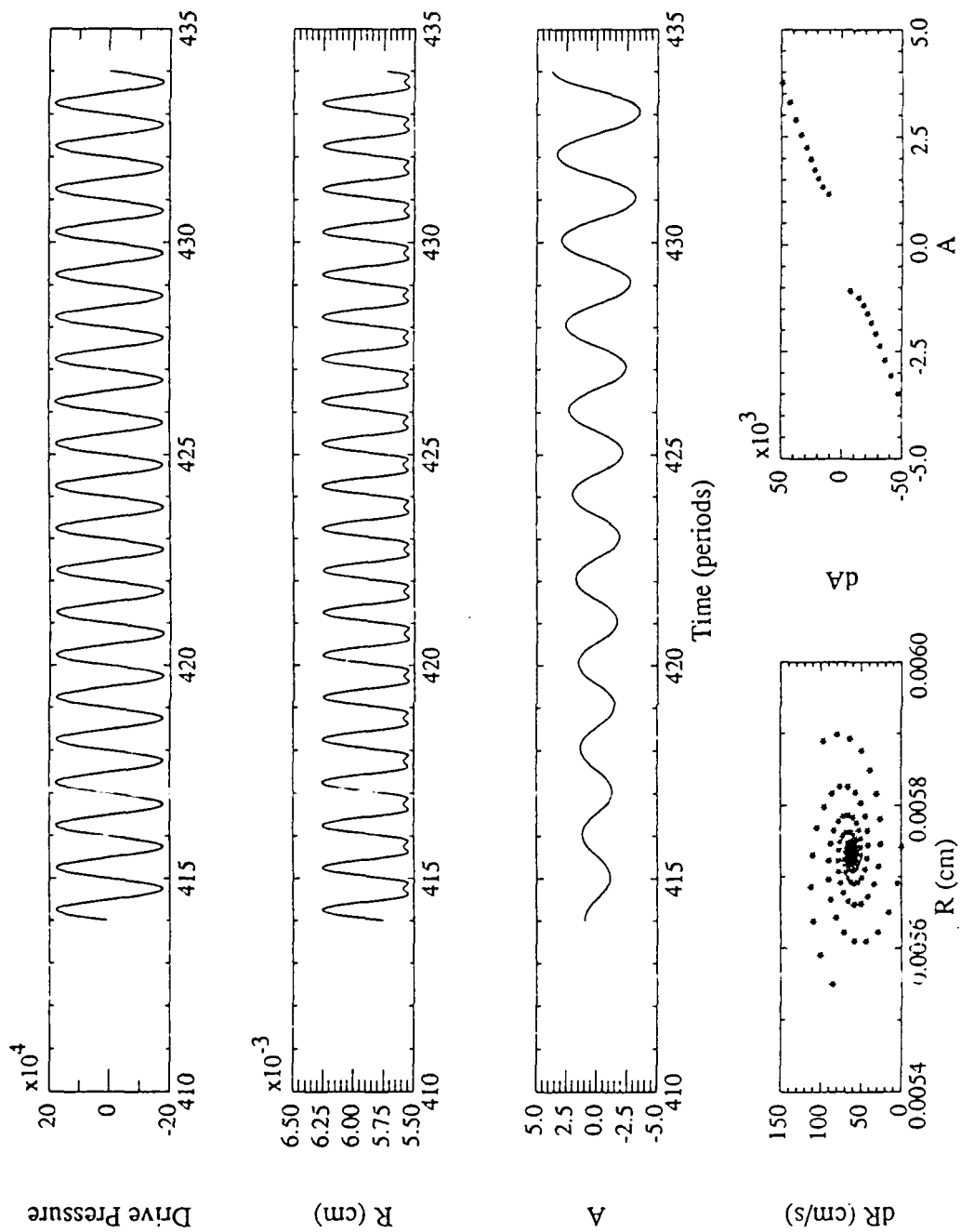


Figure C11: SOE_175_058

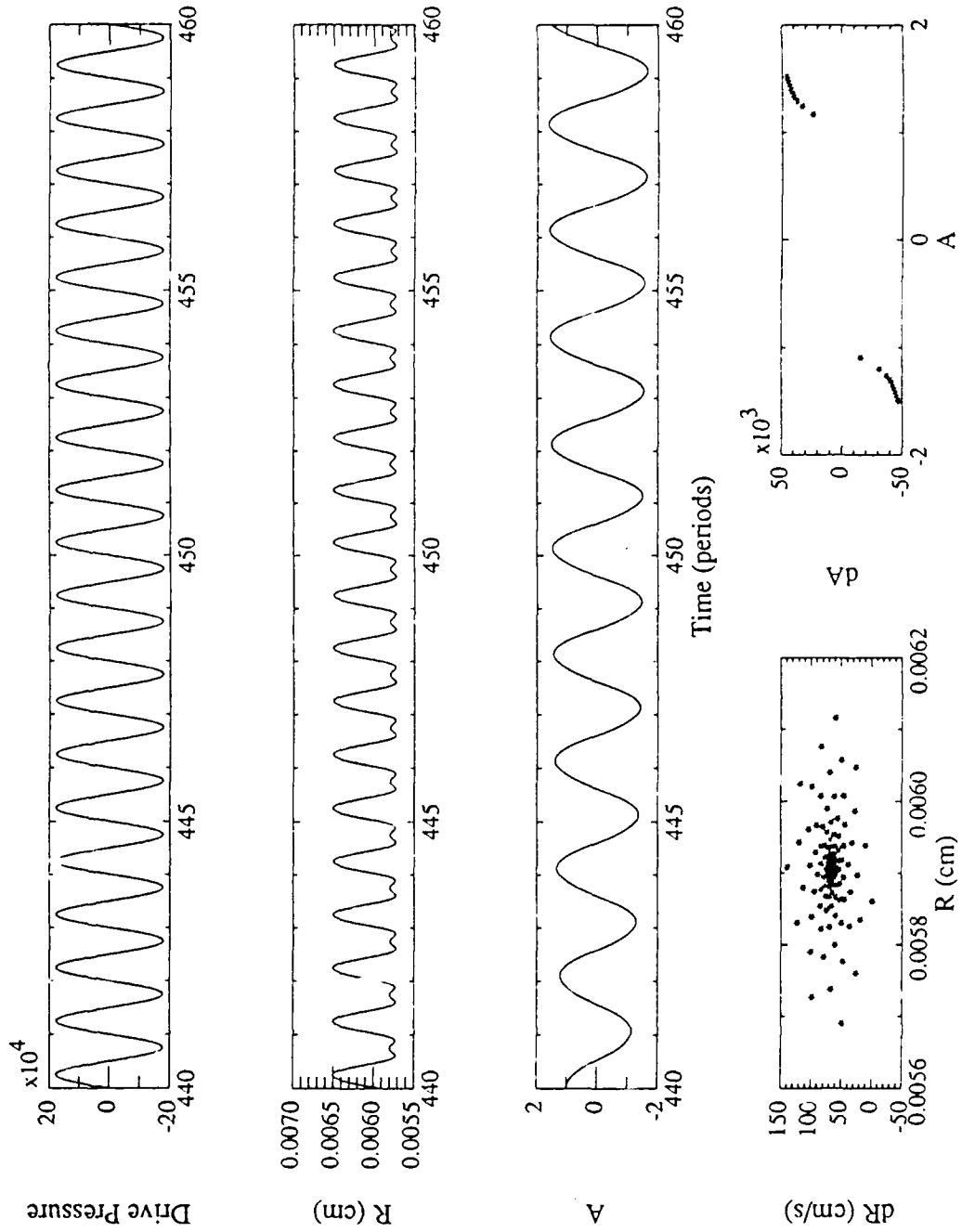


Figure C12: SOE_175_060

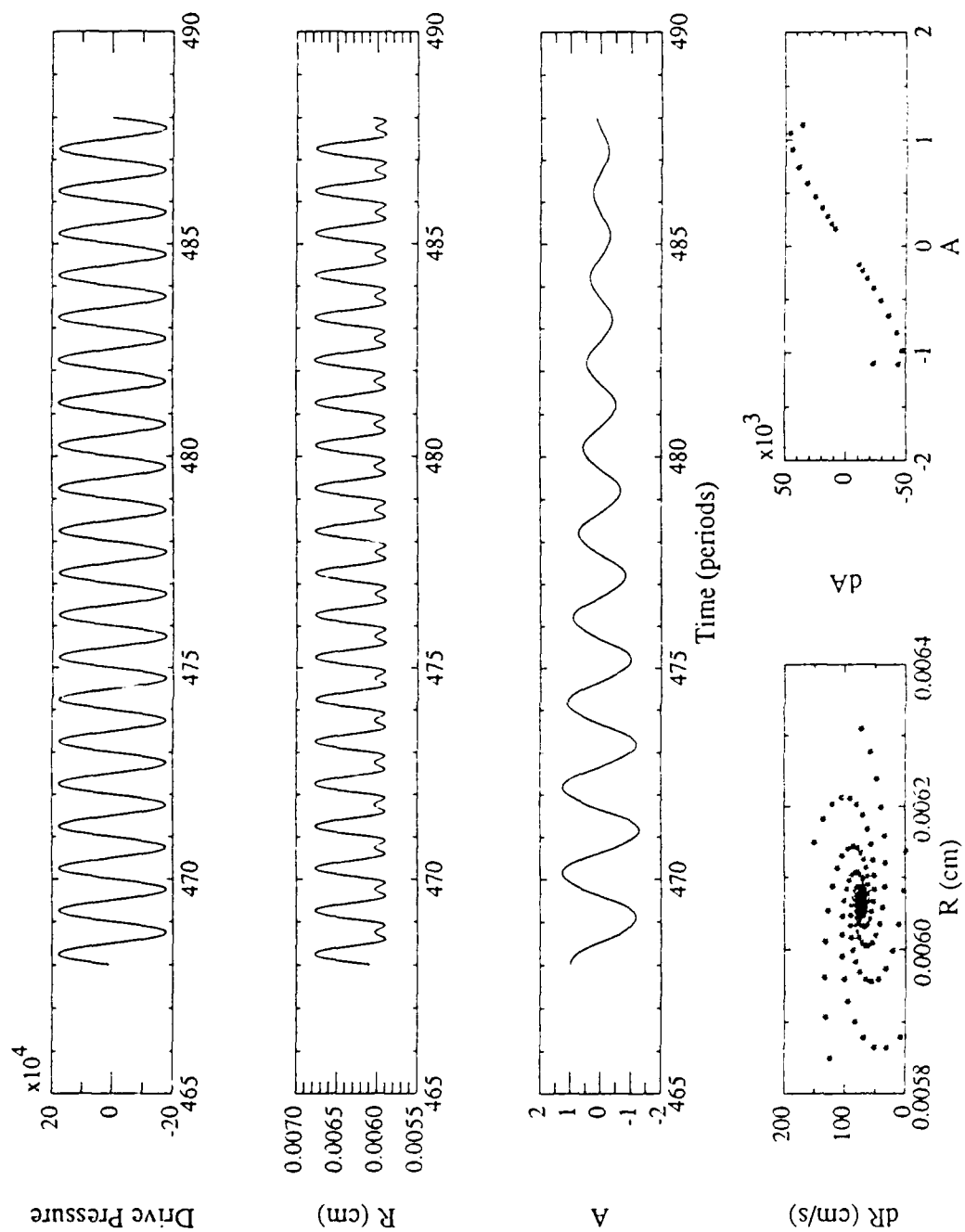


Figure C13: SOE_175_062

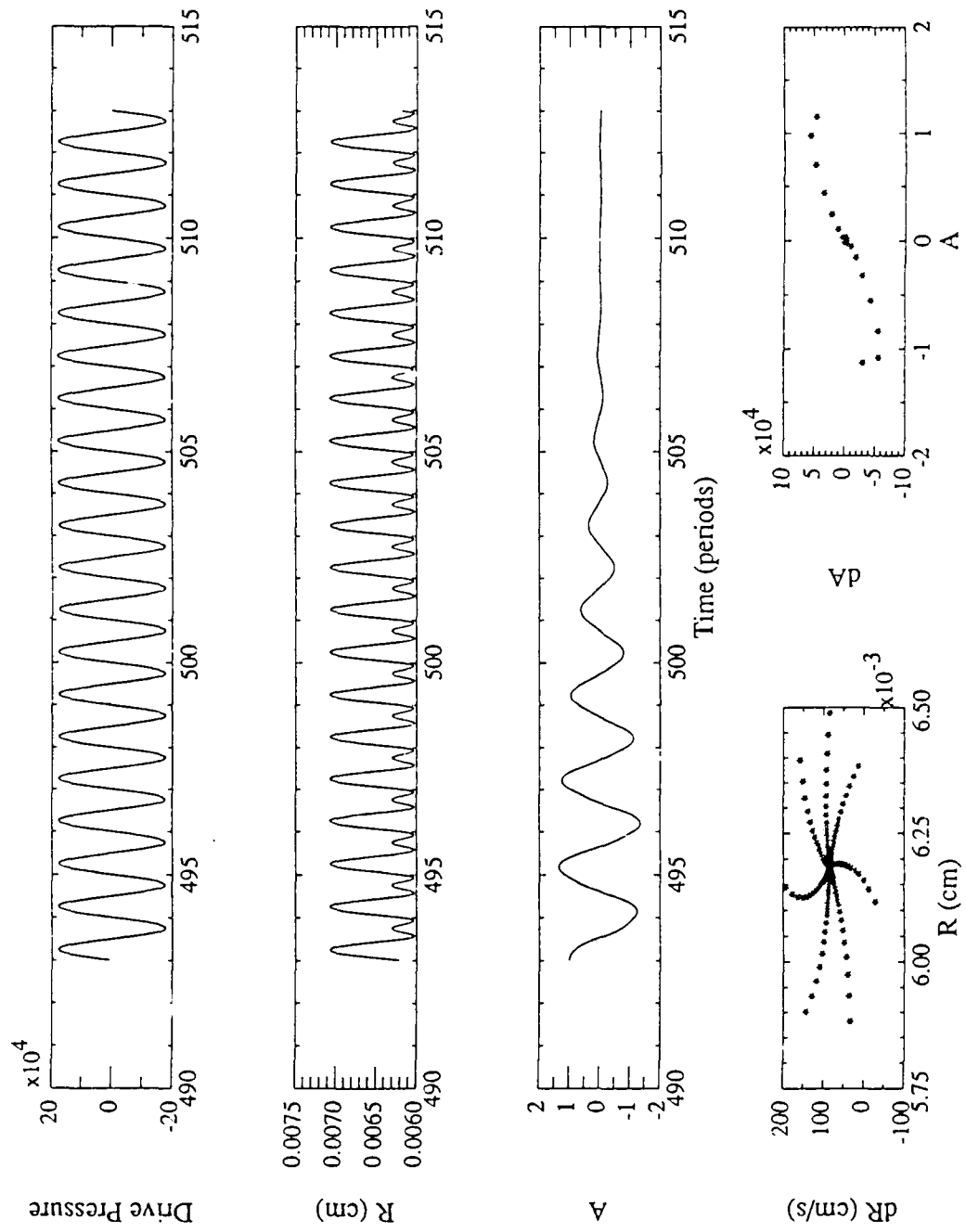


Figure C14: SOE_175_064

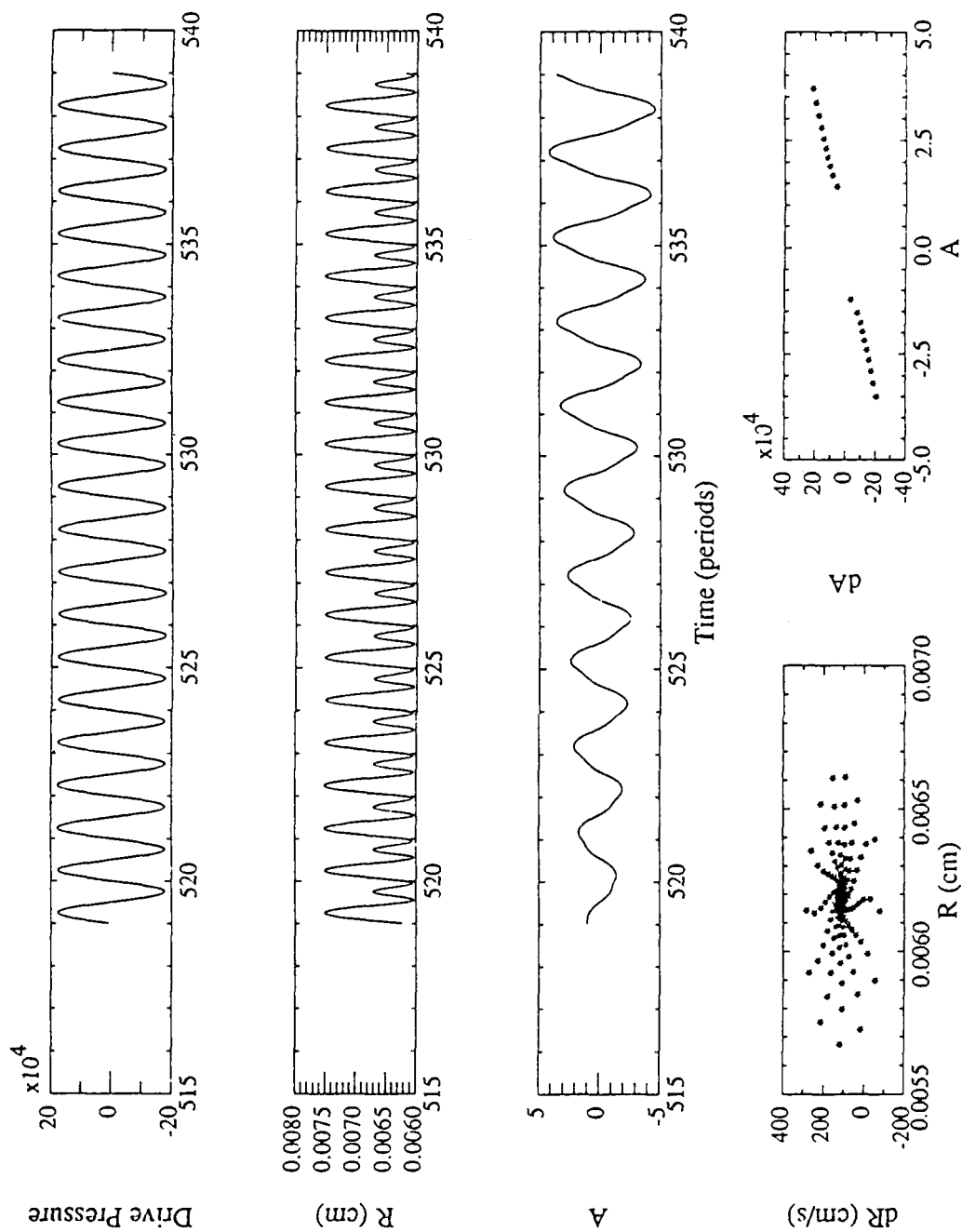


Figure C15: SOE_175_066

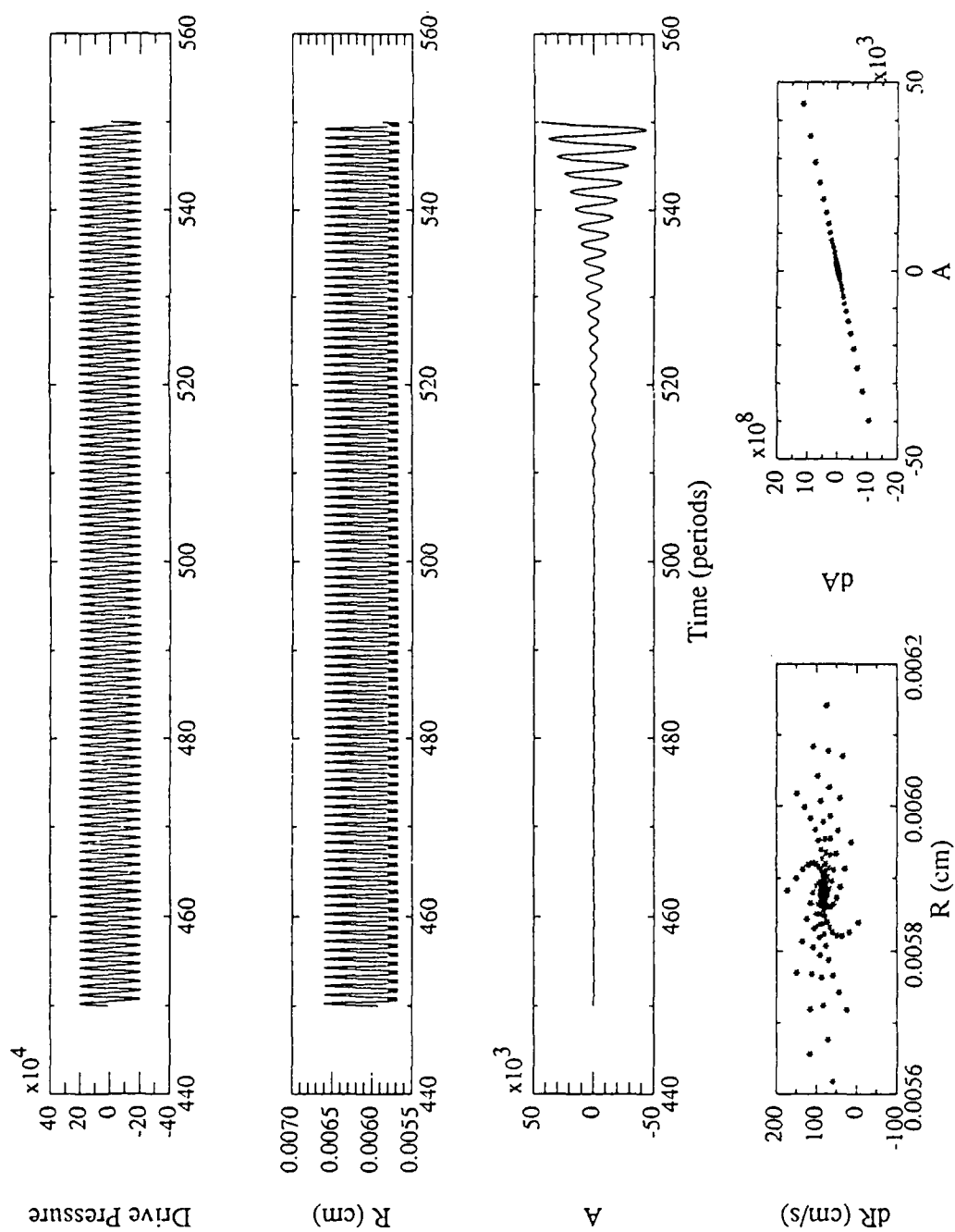


Figure C16: SOE_200_060

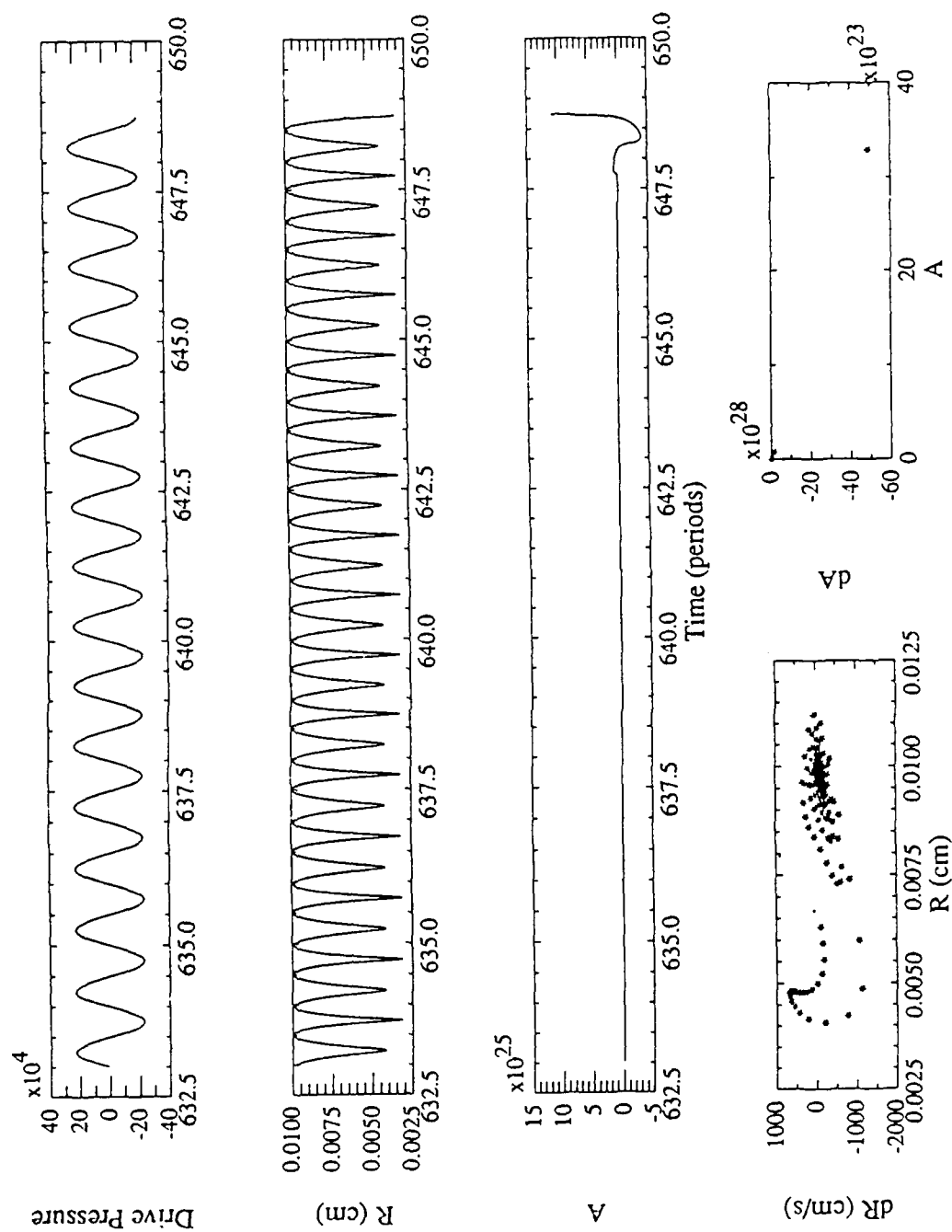


Figure C17: SOE_225_066

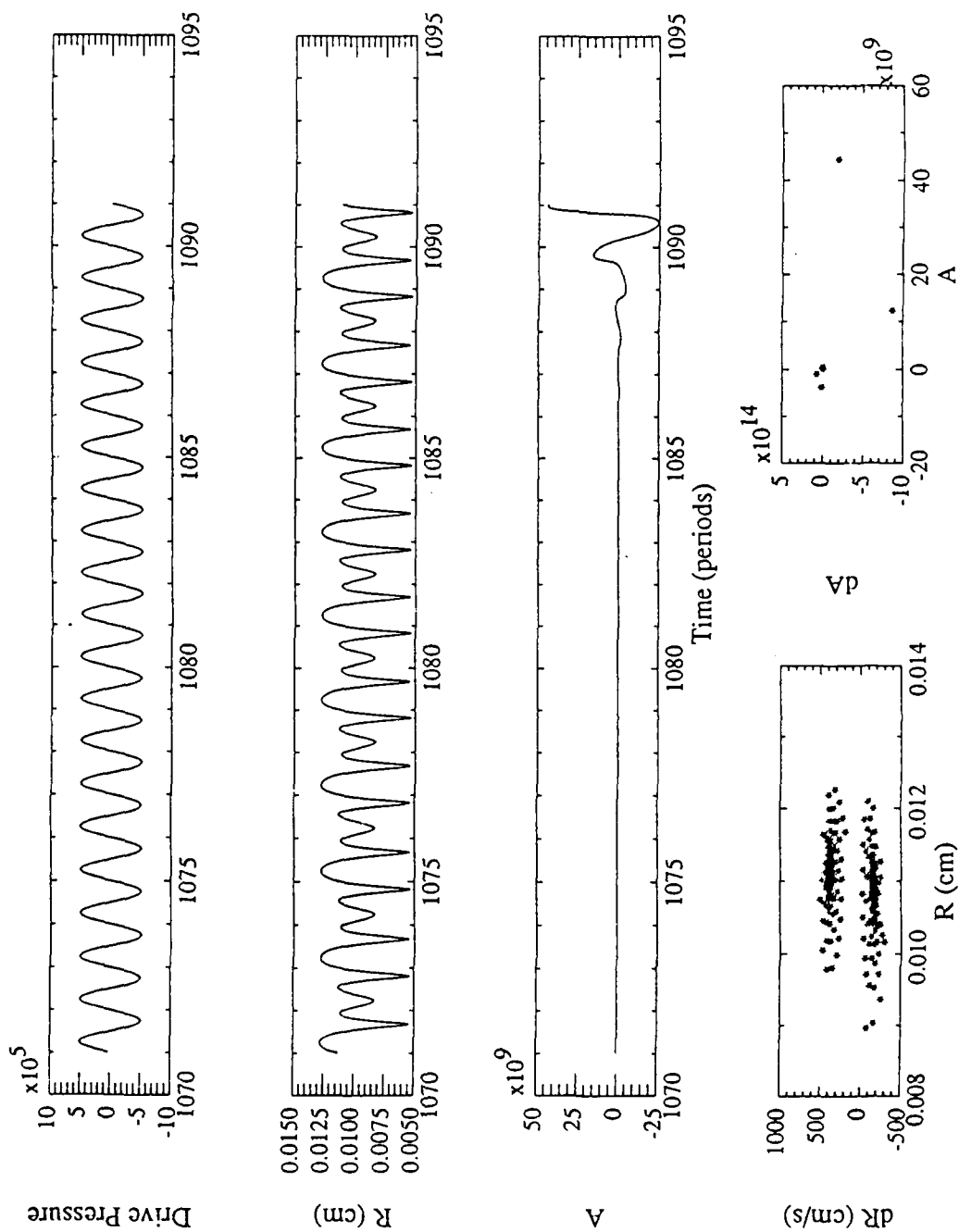


Figure C18: SOE_500_090

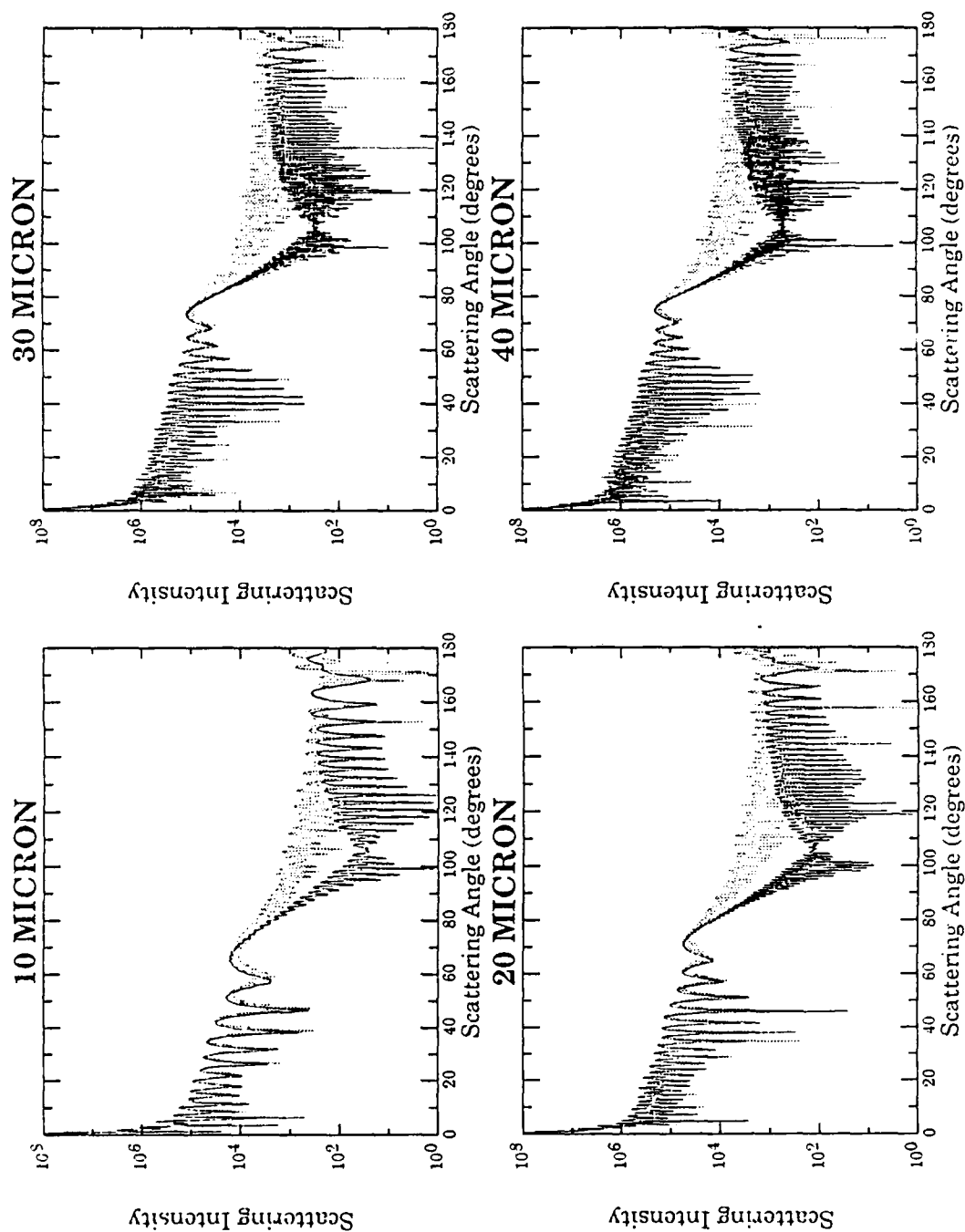


Figure C19: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of scattering angle for 10 μ , 20 μ , 30 μ , and 40 μ bubble radii.

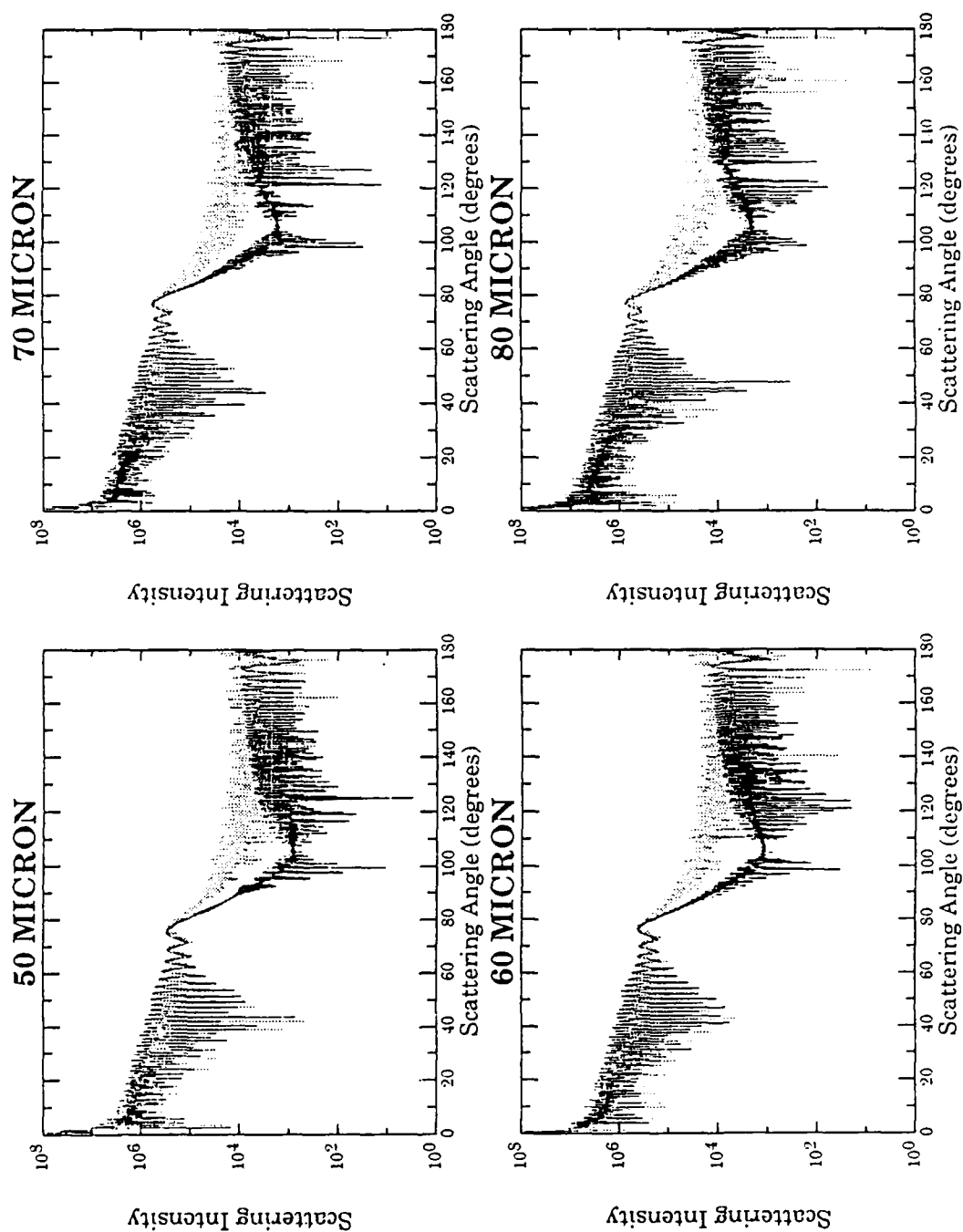


Figure C20: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of scattering angle for 50 μ , 60 μ , 70 μ , and 80 μ bubble radii.

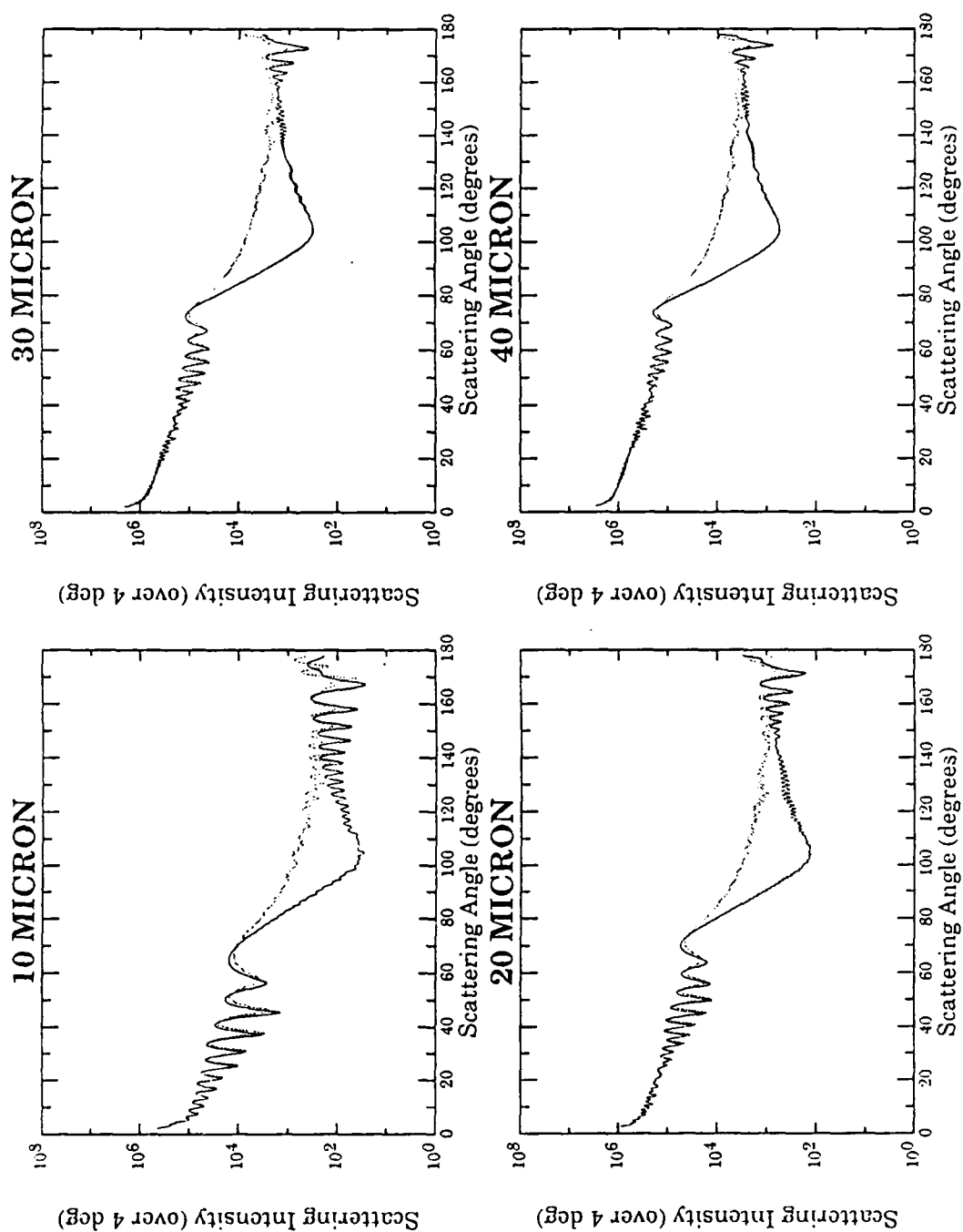


Figure C21: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a solid angle of 4 degrees as a function of scattering angle for 10 μ , 20 μ , 30 μ , and 40 μ bubble radii.

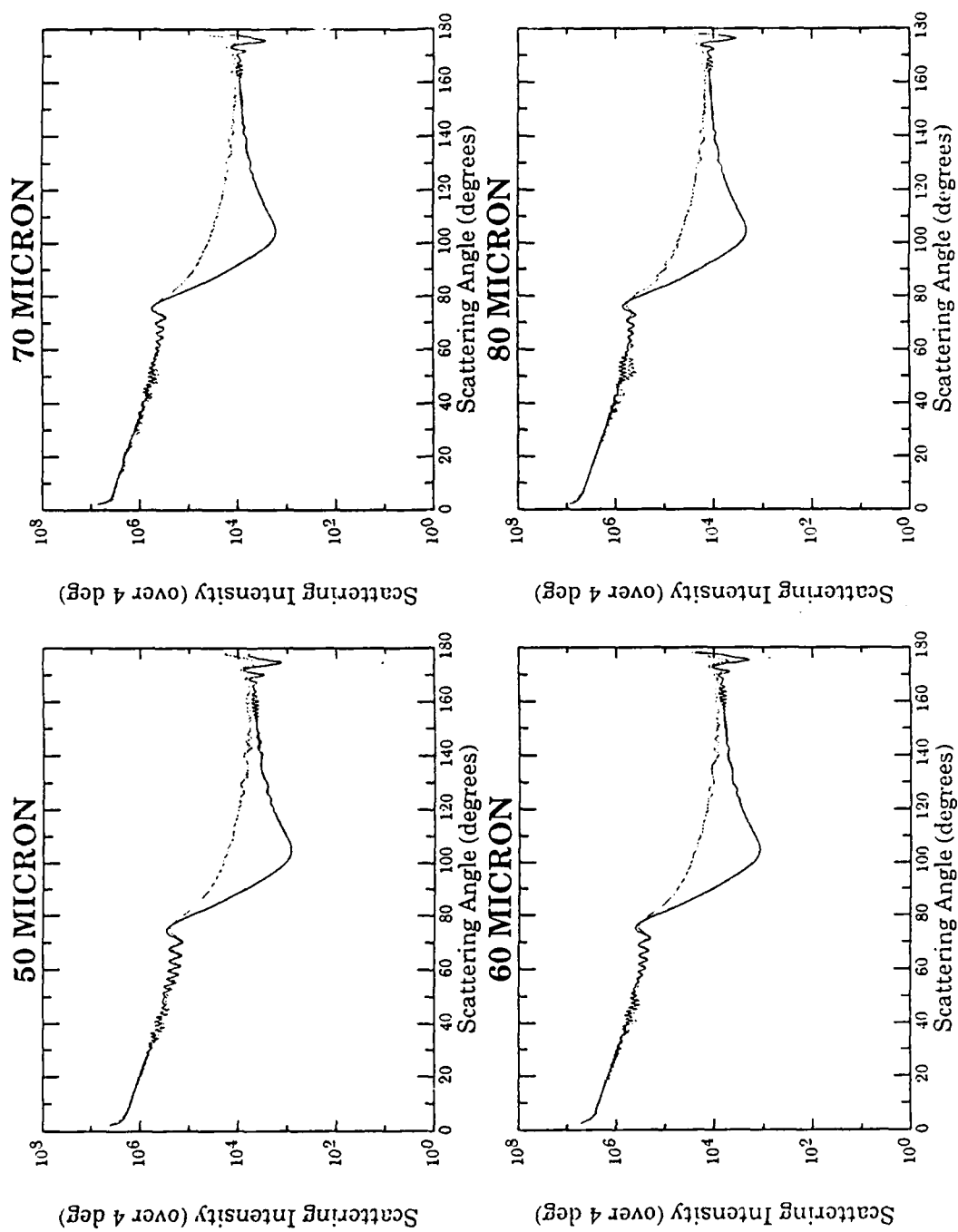


Figure C22: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a solid angle of 4 degrees as a function of scattering angle for 50 μ , 60 μ , 70 μ , and 80 μ bubble radii.

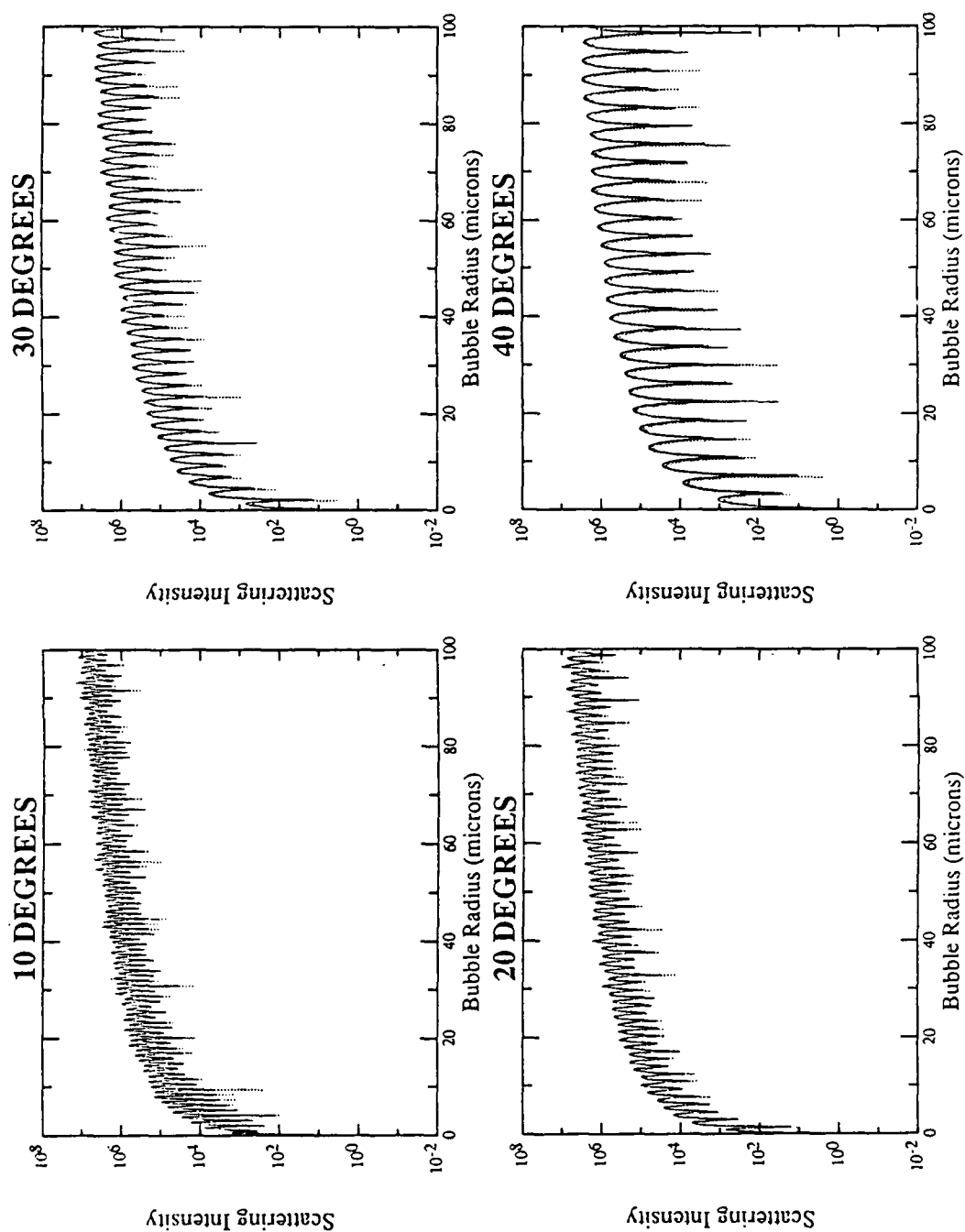


Figure C23: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of bubble radius for 10, 20, 30, and 40 degree scattering angles.

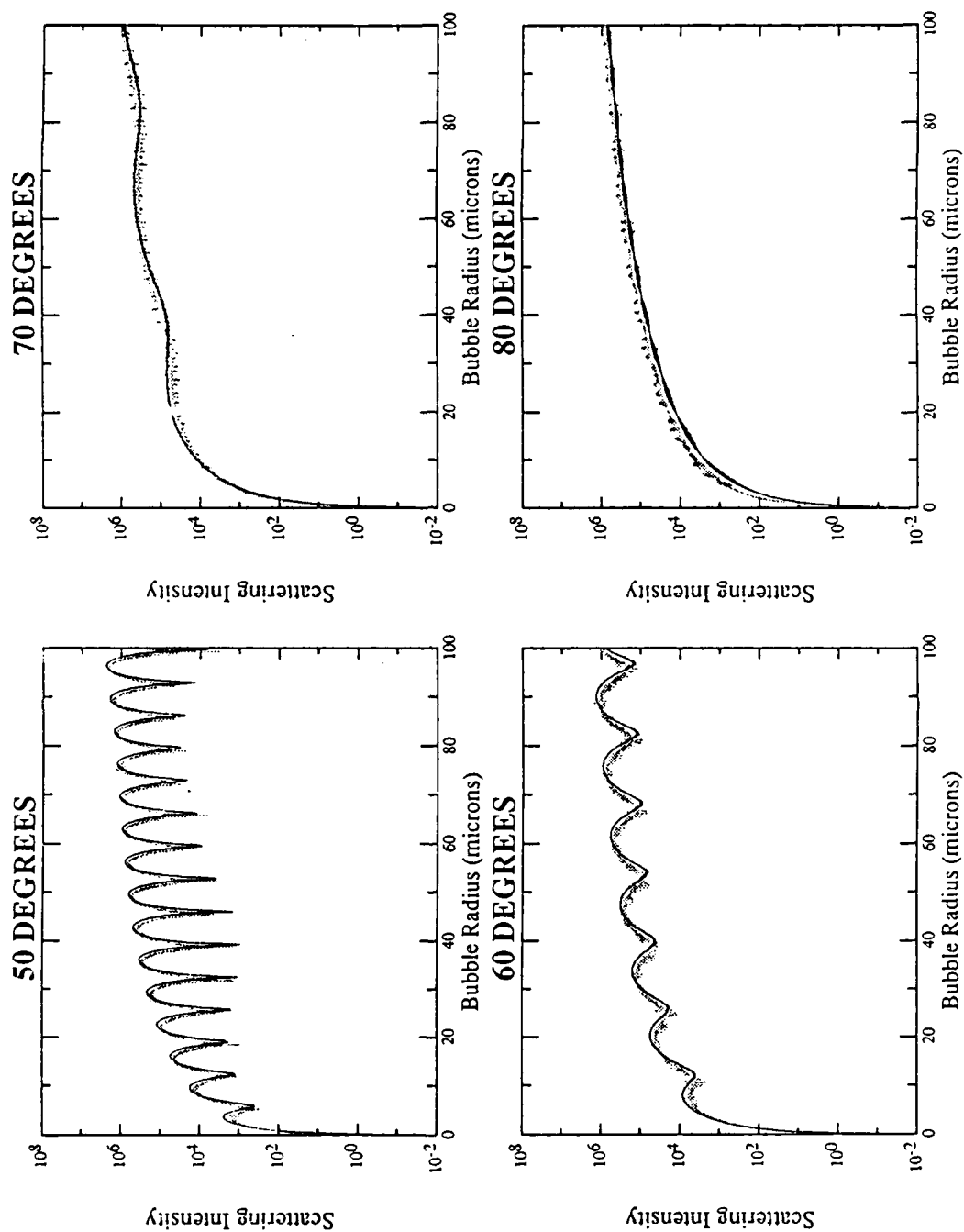


Figure C24: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of bubble radius for 50, 60, 70, and 80 degree scattering angles.

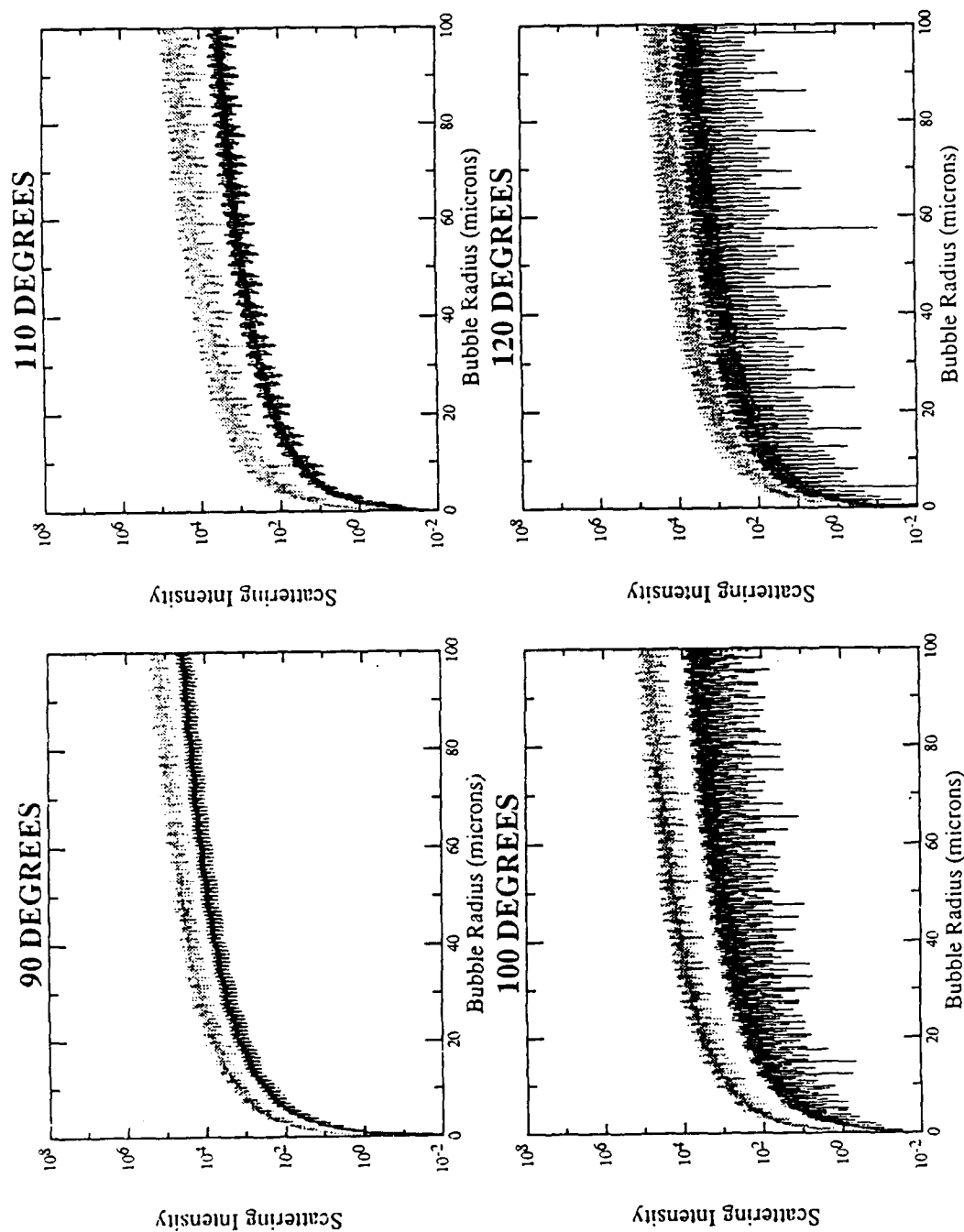


Figure C25: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of bubble radius for 90, 100, 110, and 120 degree scattering angles.

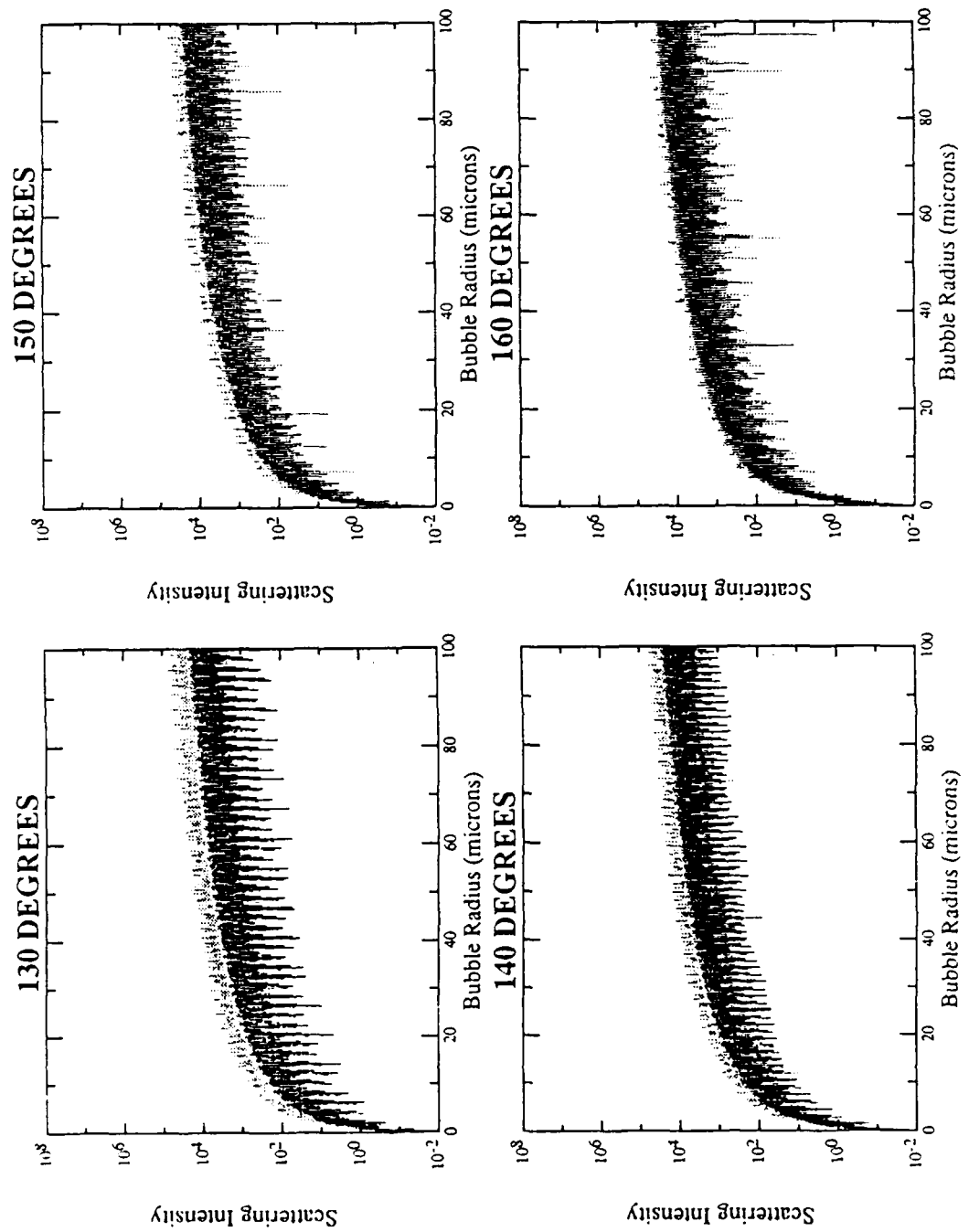


Figure C26: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of bubble radius for 130, 140, 150, and 160 degree scattering angles.

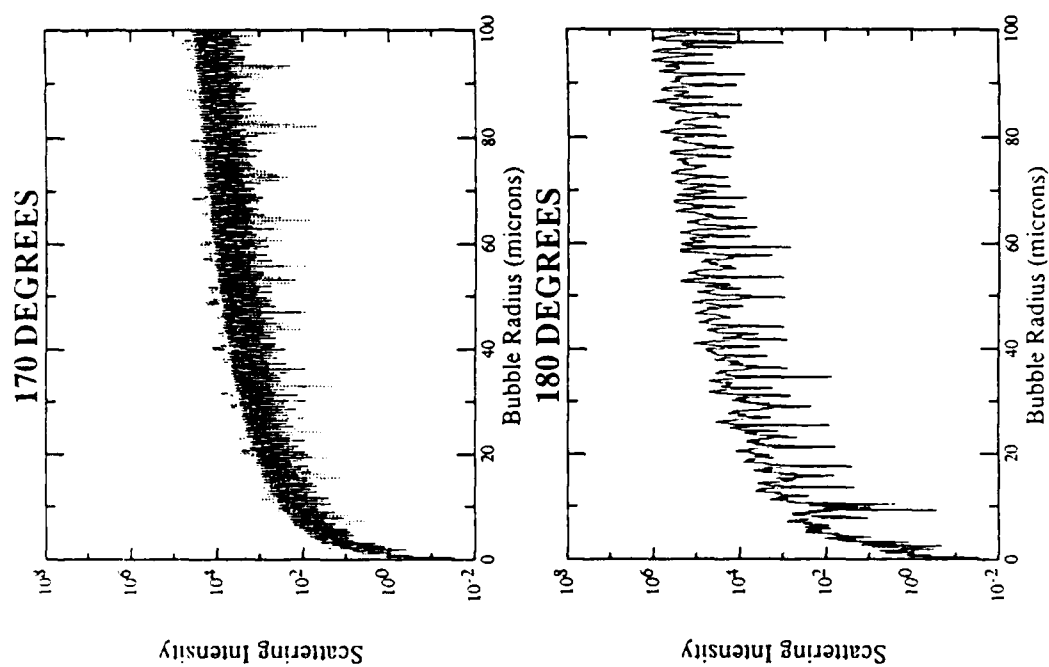


Figure C27: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), as a function of bubble radius for 170 and 180 degree scattering angles.

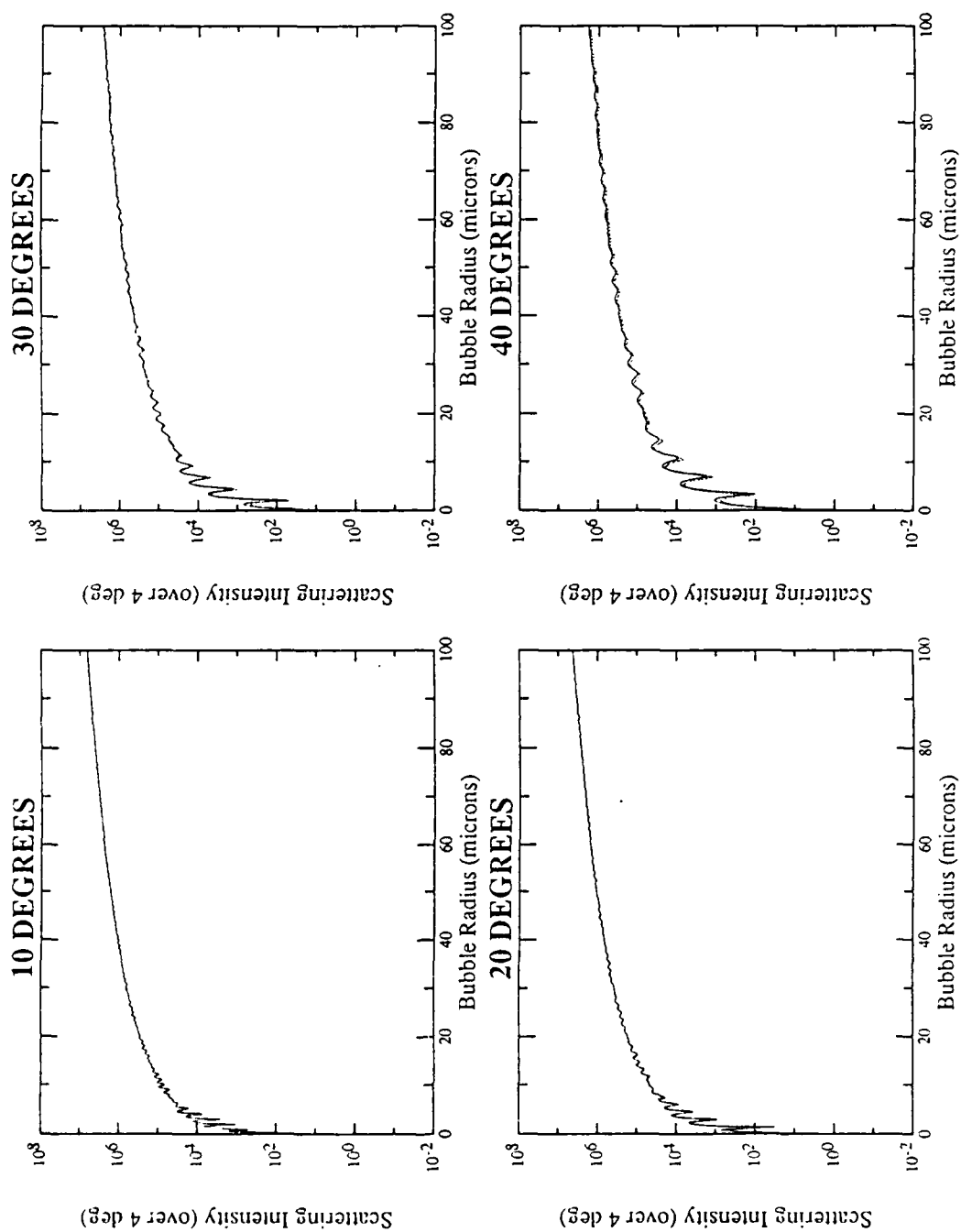


Figure C28: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a 4 degree solid angle, as a function of bubble radius for 10, 20, 30, and 40 degree scattering angles.

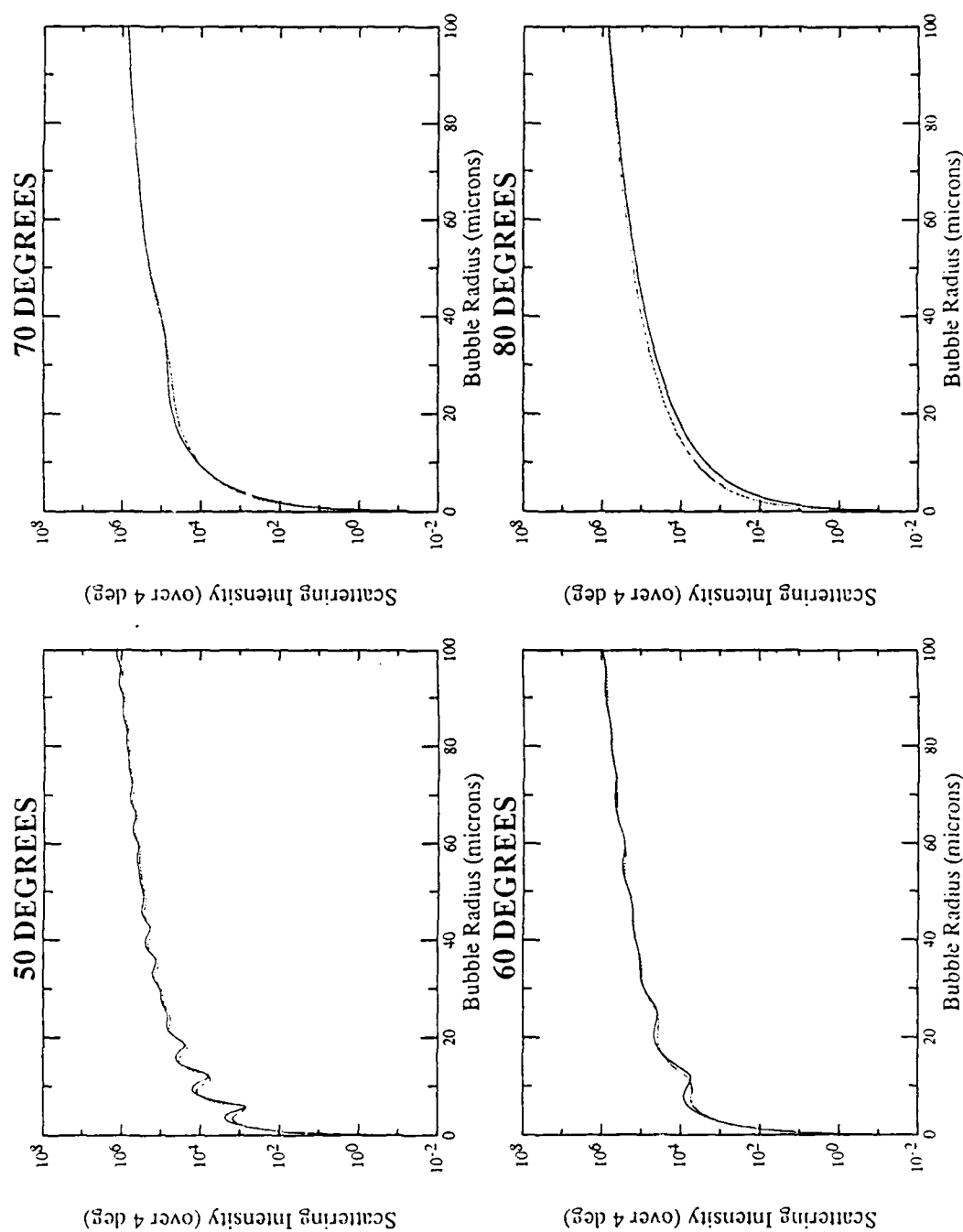


Figure C29: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a 4 degree solid angle, as a function of bubble radius for 50, 60, 70, and 80 degree scattering angles.

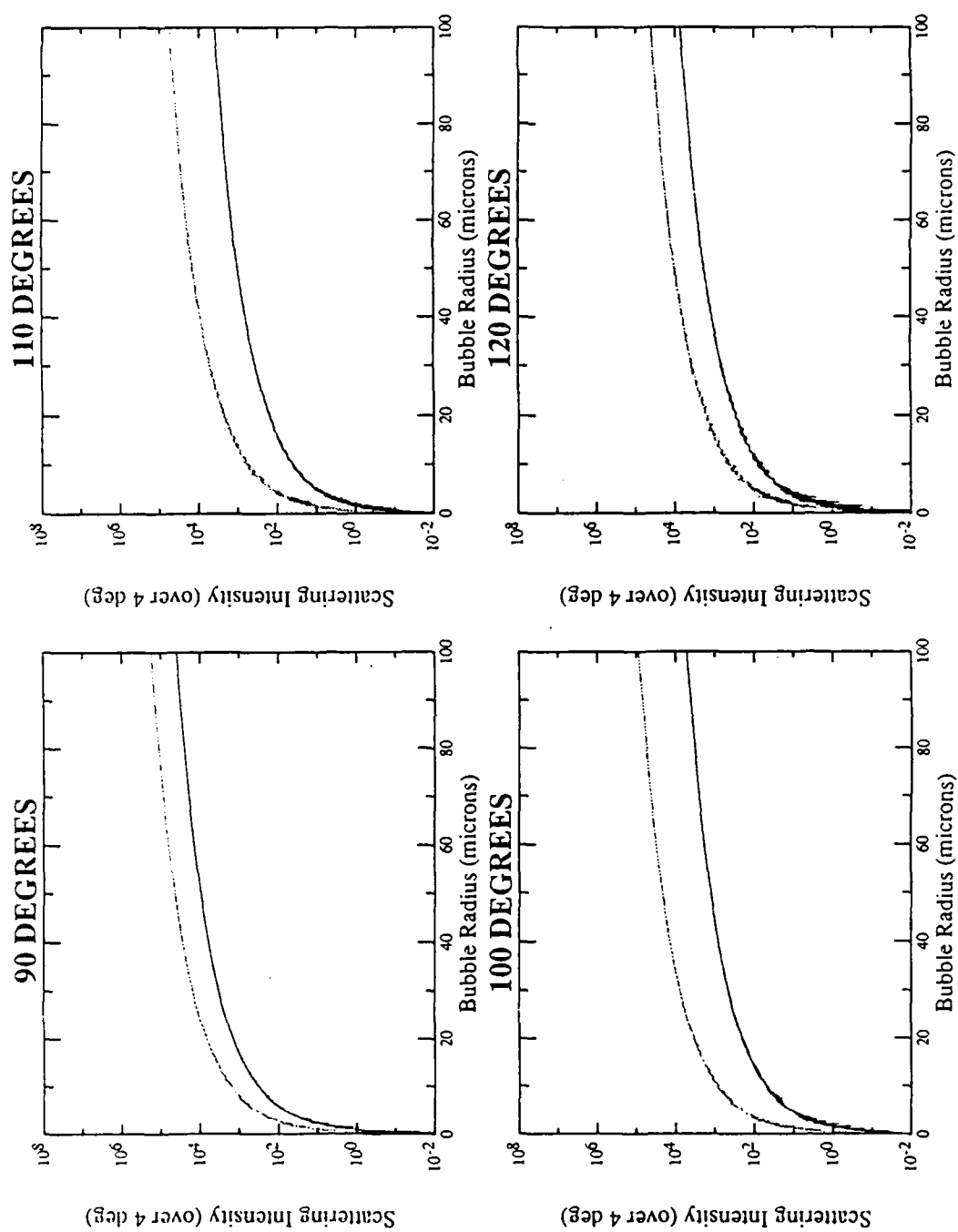


Figure C30: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a 4 degree solid angle, as a function of bubble radius for 90, 100, 110, and 120 degree scattering angles.

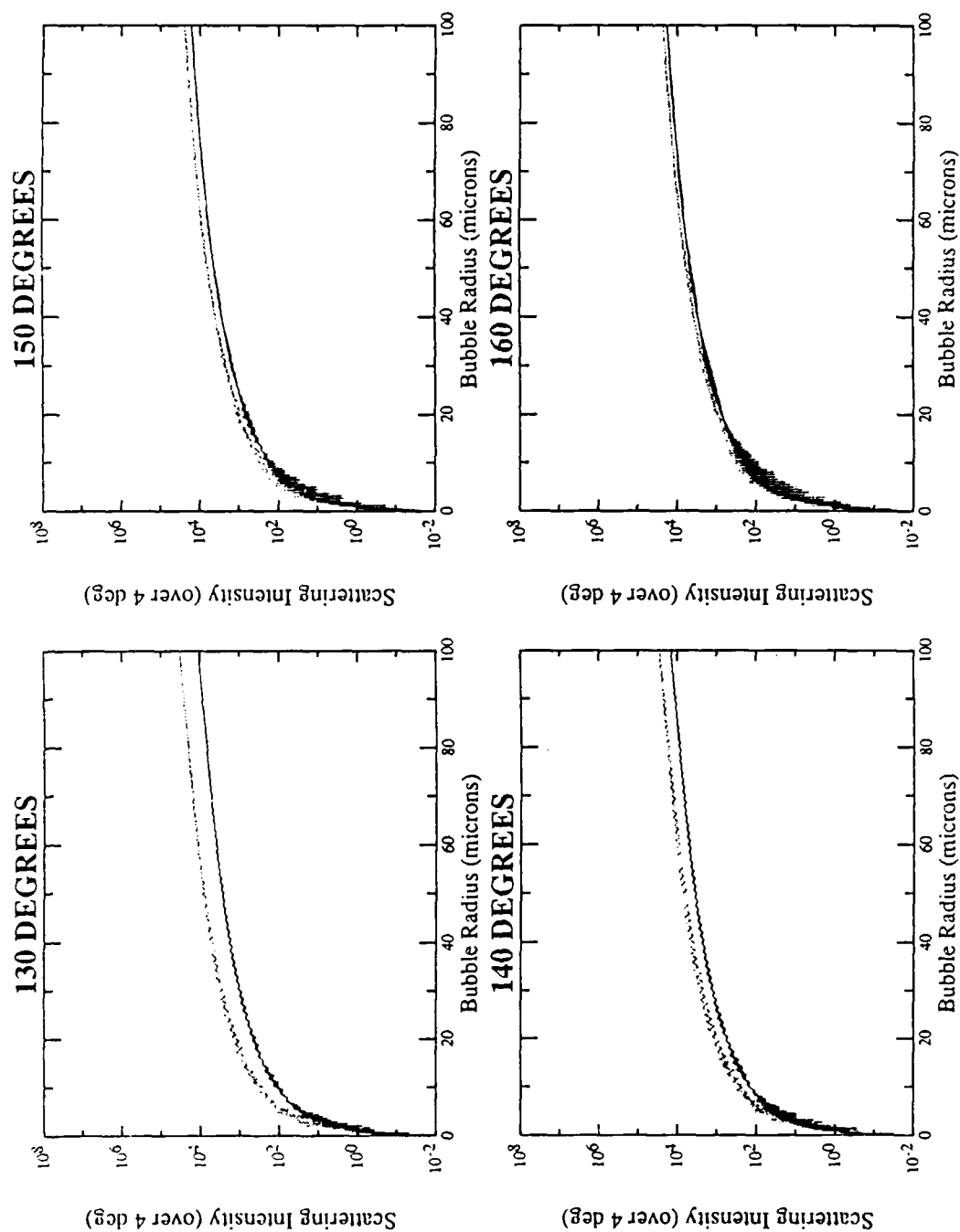


Figure C31: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a 4 degree solid angle, as a function of bubble radius for 130, 140, 150, and 160 degree scattering angles.

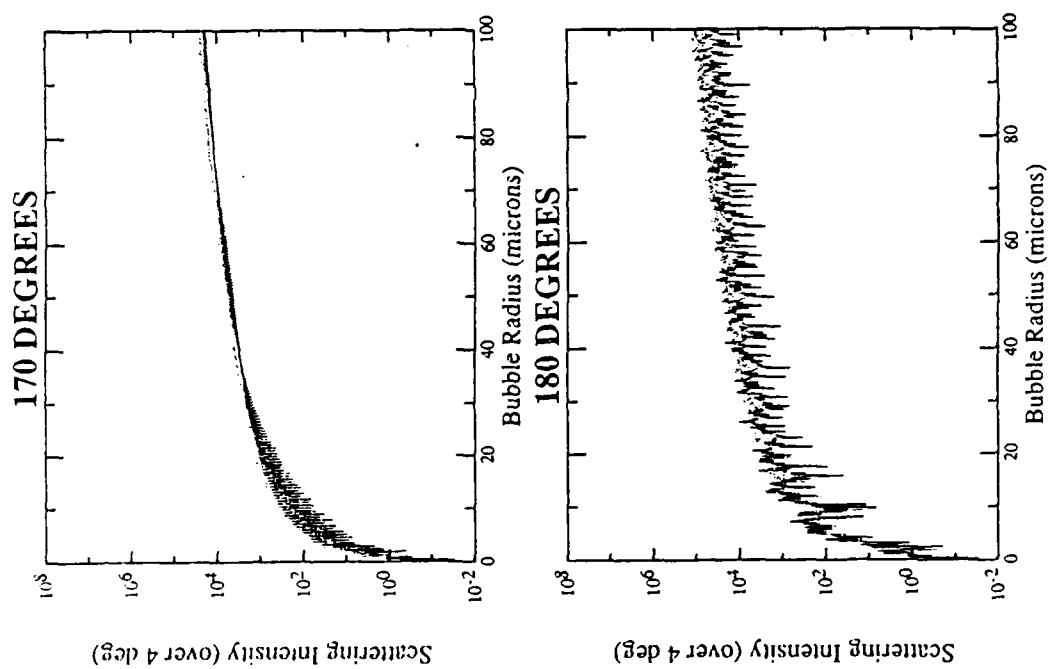


Figure C32: Mie scattering intensities, I_1 (dotted line) and I_2 (solid line), integrated over a 4 degree solid angle, as a function of bubble radius for 170 and 180 degree scattering angles.

Appendix D

This appendix contains the program listings for some of the programs used in this experimental study. Those programs included are:

DTHRESH	Analytic threshold solutions for shape oscillations
SOE	Shape Oscillation Equation solver (Numerical thresholds)
MIENOS	Mie scattering program
SLICE	Program to generate color pressure slices
SHAPE	Experimental process control and data acquisition program

PROGRAM THRESH2

C THIS PROGRAM WILL CALCULATE THE CRUM & ELLER THRESHOLD PROBLEM

```

REAL*4 R4

REAL*8 RMIN,RMAX,RINC,FREQ,NU,RHO,SIGMA,GAMMA,PA_INIT,PA_FINAL,PA_INC
REAL*8 OMEGA,OMEGA0,A,B,C,D,BETA,T0,T2,T4,T6,S1,S2,AA,PA,E
REAL*8 PA_SAVE,MU_PRIME,RO,TWOPI,ROOT,MU_TOL,PA_STEP,KAPPA
REAL*8 TMAX,TINC,T,RAVG

REAL*8 XCOF(5),COF(5),ROOTR(5),ROOTI(5)

INTEGER N,IER,J,I,I4,N_MIN,N_MAX,J_MIN,J_MAX

CHARACTER*40 FILENAME

LOGICAL CAL_GAMMA

EXTERNAL KAPPA

DATA TWOPI/6.2831853071796D0/

CALL SCRINI
CALL HEADER('CRUM & ELLER SURFACE THRESHOLDS')

CALL ASKR('ENTER MINIMUM BUBBLE RADIUS (MICRONS) (REAL) [10.0] -> ',
2      R4,10.)
RMIN=DBLE(R4)*1.D-4
CALL ASKR('ENTER MAXIMUM BUBBLE RADIUS (MICRONS) (REAL) [100.0] -> ',
2      R4,100.)
RMAX=DBLE(R4)*1.D-4
CALL ASKR('ENTER BUBBLE RADIUS INCREMENT (MICRONS) (REAL) [1.0] -> ',
2      R4,1.0)
RINC=DBLE(R4)*1.D-4
CALL ASKR('ENTER DRIVING FREQUENCY (KHz) (REAL) [22.22] -> ',R4,
2      22.22)
FREQ=DBLE(R4)
FREQ=FREQ*1000.D0
OMEGA=TWOPI*FREQ
CALL ASKR('ENTER KINEMATIC VISCOSITY (CM^2/SEC) (REAL) [0.0091] -> ',
2      R4,0.0091)
NU=DBLE(R4)
CALL ASKR('ENTER LIQUID DENSITY (G/CM^3) (REAL) [1.0] -> ',R4,
2      1.0)
RHO=DBLE(R4)
CALL ASKR('ENTER SURFACE TENSION (DYN/CM) (REAL) [72.5] -> ',R4,
2      72.5)
SIGMA=DBLE(R4)
CALL ASKR('ENTER AMBIENT PRESSURE PO (BAR) (REAL) [1.0] -> ',R4,1.0)
PO=DBLE(R4)
CALL ASKR('ENTER INITIAL PRESSURE PA (BAR) (REAL) [0.05] -> ',R4,
2      ,0.05)
PA_INIT=DBLE(R4)
CALL ASKR('ENTER FINAL PRESSURE PA (BAR) (REAL) [0.3] -> ',R4,
2      0.3)
PA_FINAL=DBLE(R4)
CALL ASKR('ENTER PRESSURE INCREMENT PA (BAR) (REAL) [0.01] -> ',R4

```

```

2      ,0.01)
PA_INC=DBLE(R4)
CALL ASKR('ENTER POLYTROPIC EXPONENT (0.0=CALCULATED) (REAL) [1.4] ->',
2      R4,1.4)
GAMMA=DBLE(R4)
CAL_GAMMA=R4.EQ.0.0
CALL ASKR('ENTER ERROR TOLERANCE (REAL) [0.001] ->',R4,0.001)
MU_TOL=DBLE(R4)
CALL ASKI('ENTER MINIMUM N VALUE (INTEGER) [2] -> ',N_MIN,2)
CALL ASKI('ENTER MAXIMUM N VALUE (INTEGER) [5] -> ',N_MAX,5)
CALL ASKI('ENTER MINIMUM J (INTEGER) [1] -> ',J_MIN,1)
CALL ASKI('ENTER MAXIMUM J (INTEGER) [3] -> ',J_MAX,3)
CALL ASK('ENTER BASE OUTPUT FILE NAME [THRESH.DAT] -> ',FILENAME,
2      'THRESH.DAT')

CALL HEADER('CRUM & ELLER SURFACE THRESHOLDS')

CALL SAY('MINIMUM BUBBLE RADIUS (MICRONS)      =',REAL(RMIN*1.D4))
CALL SAY('MAXIMUM BUBBLE RADIUS (MICRONS)      =',REAL(RMAX*1.D4))
CALL SAY('BUBBLE RADIUS INCREMENT (MICRONS)    =',REAL(RINC*1.D4))
CALL SAY('DRIVING FREQUENCY (KHZ)              =',REAL(FREQ/1000.D0))
CALL SAY('KINIMATIC VISCOSITY (CM^2/SEC)       =',REAL(NU))
CALL SAY('LIQUID DENSITY (G/CM^3)              =',REAL(RHO))
CALL SAY('SURFACE TENSION (DYN/CM)             =',REAL(SIGMA))
CALL SAY('AMBIENT PRESSURE PO (BAR)            =',REAL(PO))
CALL SAY('INITIAL PRESSURE (DIMENSIONLESS)      =',REAL(PA_INIT))
CALL SAY('FINAL PRESSURE (DIMENSIONLESS)       =',REAL(PA_FINAL))
CALL SAY('PRESSURE INCREMENT (DIMENSIONLESS)    =',REAL(PA_INC))
CALL SAY('POLYTROPIC EXPONENT (DIMENSIONLESS)  =',REAL(GAMMA))
CALL SAY('ERROR TOLERANCE (DIMENSIONLESS)      =',REAL(MU_TOL))
CALL SAY('MINIMUM N VALUE                      =',FLOAT(N_MIN))
CALL SAY('MAXIMUM N VALUE                      =',FLOAT(N_MAX))
CALL SAY('MINIMUM J VALUE                      =',FLOAT(J_MIN))
CALL SAY('MAXIMUM J VALUE                      =',FLOAT(J_MAX))
TYPE *, 'BASE FILE NAME                      = ',FILENAME

CALL PROMPT('PROCESSING ...',0)

OPEN(UNIT=1,NAME=FILENAME(1:INDEX(FILENAME, '.'))// 'STATS',
2      STATUS='NEW')
CALL OUT(1, 'MINIMUM BUBBLE RADIUS (CM)          =',REAL(RMIN*1.D4))
CALL OUT(1, 'MAXIMUM BUBBLE RADIUS (CM)          =',REAL(RMAX*1.D4))
CALL OUT(1, 'BUBBLE RADIUS INCREMENT (CM)        =',REAL(RINC*1.D4))
CALL OUT(1, 'DRIVING FREQUENCY (KHZ)            =',REAL(FREQ/1000.D0))
CALL OUT(1, 'KINIMATIC VISCOSITY (CM^2/SEC)      =',REAL(NU))
CALL OUT(1, 'LIQUID DENSITY (G/CM^3)            =',REAL(RHO))
CALL OUT(1, 'SURFACE TENSION (DYN/CM)           =',REAL(SIGMA))
CALL OUT(1, 'AMBIENT PRESSURE PO (BAR)           =',REAL(PO))
CALL OUT(1, 'INITIAL PRESSURE (DIMENSIONLESS)    =',REAL(PA_INIT))
CALL OUT(1, 'FINAL PRESSURE (DIMENSIONLESS)     =',REAL(PA_FINAL))
CALL OUT(1, 'PRESSURE INCREMENT (DIMENSIONLESS) =',REAL(PA_INC))
CALL OUT(1, 'POLYTROPIC EXPONENT (DIMENSIONLESS)=',REAL(GAMMA))
CALL OUT(1, 'ERROR TOLERANCE (DIMENSIONLESS)    =',REAL(MU_TOL))
CALL OUT(1, 'MINIMUM N VALUE                    =',FLOAT(N_MIN))
CALL OUT(1, 'MAXIMUM N VALUE                    =',FLOAT(N_MAX))
CALL OUT(1, 'MINIMUM J VALUE                    =',FLOAT(J_MIN))
CALL OUT(1, 'MAXIMUM J VALUE                    =',FLOAT(J_MAX))

```

```

WRITE(1,*) 'BASE FILE NAME' = ' ,FILENAME
CLOSE(UNIT=1)

CALL LIB$INIT_TIMER

PO=PO*1.D6
PA_INIT=PA_INIT*1.D6
PA_FINAL=PA_FINAL*1.D6
PA_INC=PA_INC*1.D6

DO N=N_MIN,N_MAX,1
  DO J=J_MIN,J_MAX,1

    OPEN(UNIT=1,NAME=FILENAME(1:INDEX(FILENAME,'.')-1)//'_N'//
2      CHAR(N+ICHAR('0'))//'_J'//CHAR(J+ICHAR('0'))//
2      FILENAME(INDEX(FILENAME,'.') :),STATUS='NEW')

    DO RO=RMIN,RMAX,RINC
      PA_SAVE=0
      MU_PRIME=2*(N+2)*(2*N+1)*NU/(OMEGA*RO*RO)
      IF(CAL_GAMMA) GAMMA=KAPPA(PO,FREQ,RO)

      PA=PA_INIT
      PA_STEP=PA_INC
      DO WHILE(PA.LE.PA_FINAL)
        OMEGA0=SQRT(3*GAMMA*PO/(RHO*RO*RO))
        BETA=OMEGA/OMEGA0
        AA=PA/(3*GAMMA*PO*(BETA*BETA-1))
        A=4*SIGMA*(N+2)*(N*N-1)/(RHO*OMEGA*OMEGA*RO*RO)
        B=(3*GAMMA+1-BETA*BETA)/4
        C=(3*GAMMA+1+5*BETA*BETA)/(4*(1-4*BETA*BETA))
        D=(12*(3*GAMMA+1+11*BETA*BETA)*C-(3*GAMMA+1)*(3*GAMMA+2)
2        )/(24*(1-9*BETA*BETA))
        T0=A+(3*A*(1-B)-(2*N+2.5D0))*AA*AA
        T2=(2*N+1-1.5D0*A)*AA+(3*A*(2*B+C-1.25D0)-3*C+0.75D0-
2        (N+0.5D0)*(5*C+2*B-1.5D0))*AA*AA*AA
        T4=(1.5D0*A*(1-C)+0.75D0+(N+0.5D0)*(8*C-1))*AA*AA
        T6=(A*(3*C-1.5D0*D-1.25D0)+3*C-0.75D0+(N+0.5D0)*(18*D+0.5D0-
2        5*C))*AA*AA*AA

        IF(J.EQ.1) THEN
          S1=T2/2+T2*T4/8+T4*T6/24
          E=T0/2-0.5D0+T2*T2/16+T4*T4/12
          XCOF(1)=E*E-S1*S1
          XCOF(2)=0
          XCOF(3)=1+(E*T2*T2)/(4*S1*S1)
          XCOF(4)=0
          XCOF(5)=(T2/(2*S1))**4/4
        ELSE IF(J.EQ.2) THEN
          S1=T4/4-T2*T2/16+T2*T6/24
          E=T0/4-1-T2*T2/24+T4*T4/128+T6*T6/40
          XCOF(1)=E*E-S1*S1
          XCOF(2)=0
          XCOF(3)=1+(E*T4*T4)/(32*S1*S1)
          XCOF(4)=0
          XCOF(5)=(T4/(4*S1))**4/16
        ELSE IF(J.EQ.3) THEN

```

```

S1=T6/6-T2*T4/24
E=T0/6-1.5D0-T2*T2/96-T4*T4/60+T6*T6/432
XCOF(1)=E-E-S1*S1
XCOF(2)=0
XCOF(3)=1+(E*T6*T6)/(108*S1*S1)
XCOF(4)=0
XCOF(5)=(T6/(6*S1))**4/36
ENDIF
CALL SOLVE(XCOF,COF,4,ROOTR,ROOTI,IER)
CALL DPOLRT(XCOF,COF,4,ROOTR,ROOTI,IER)
IF (IER.NE.0) THEN
  TYPE *, 'ERROR IN DPOLRT, IER=', IER
  TYPE *, 'R, PA=', RO, PA
  GOTO 10
ENDIF

ROOT=ROOTR(1)          !FIND CLOSEST ROOT TO MU_PRIME
DO I=2,4,1
  IF (ABS(ROOTR(I)-MU_PRIME).LT.ABS(ROOT-MU_PRIME).AND.ROOTR(I)
    .GT.0) ROOT=ROOTR(I)
END DO

IF (ROOT.GE.MU_PRIME) THEN
  IF (ABS(ROOT-MU_PRIME).LE.MU_TOL) THEN
    PA_SAVE=PA
    GOTO 11              !FOUND THRESHOLD!
  ENDIF
  PA=MAX(PA-PA_STEP,PA_INIT)
  PA_STEP=PA_STEP/2      !HALVE STEP SIZE
  PA=MAX(PA-PA_STEP,PA_INIT)
  IF (PA_STEP.LT.PA_INC/100000) THEN
    TYPE *, 'UNABLE TO OBTAIN DESIRED TOLERANCE IN 50000 HALVES'
    TYPE *, 'RO, MU_PRIME=', REAL(RO*1.D4), REAL(MU_PRIME)
    TYPE *, 'ROOT=', REAL(ROOT)
    PA_SAVE=PA           !GIVE VALUE ANYWAY
    GOTO 11
  ENDIF
ENDIF

10    PA=PA+PA_STEP

END DO    !PA
GOTO 12

IF (ROOT.GT.MU_PRIME) THEN      !NO THRESHOLD
  PA_SAVE=PA_INIT
ELSE
  PA_SAVE=PA_FINAL
ENDIF

11    TMAX=1.D0/22220.D0
    TINC=TMAX/200
    RAVG=0
    I=0
    DO T=0.D0,TMAX,TINC
      RAVG=RAVG+RO*((1+AA*AA*B)+AA*COS(OMEGA*T)+AA*AA*C*
        COS(2*OMEGA*T)+AA*AA*AA*D*COS(3*OMEGA*T))
      I=I+1
    END DO
    RAVG=RAVG/I

```

```

        WRITE(1,*) REAL(RO*1.D4), REAL(RAVG*1.D4), REAL(PA_SAVE*1.D-6),
2         REAL(ROOT-MU_PRIME)

12      CONTINUE
        END DO          !RO
        CLOSE(UNIT=1)

        END DO          !J
        END DO          !N

        CALL LIB$SHOW_TIMER
        CALL EXIT
        END

        SUBROUTINE SAY(STR,VAL)
        CHARACTER*(*) STR
        REAL*4 VAL

        IF (ABS(VAL) .LT. 32767) THEN
            IF (IFIX(VAL) .EQ. VAL) THEN
                TYPE *, STR, IFIX(VAL)
                RETURN
            ENDIF
        ENDIF

        TYPE *, STR, VAL

        RETURN
        END

        SUBROUTINE OUT(CHAN,STR,VAL)
        INTEGER CHAN
        CHARACTER*(*) STR
        REAL*4 VAL

        IF (ABS(VAL) .LT. 32767) THEN
            IF (IFIX(VAL) .EQ. VAL) THEN
                WRITE(CHAN,*) STR, IFIX(VAL)
                RETURN
            ENDIF
        ENDIF

        WRITE(CHAN,*) STR, VAL

        RETURN
        END

        SUBROUTINE SOLVE(XCOF,COF,NEQ,ROOTR,ROOTI,IER)

        REAL*8 XCOF(1),COF(1),ROOTR(1),ROOTI(1)
        INTEGER NEQ,IER
        COMPLEX*16 A,B,C,MU

        IER=0

        A=DCMPLX(XCOF(5))
        B=DCMPLX(XCOF(3))
        C=DCMPLX(XCOF(1))

```

```
MU=SQRT ( (-B+SQRT (B*B-4*A*C) ) / (2*A) )  
ROOTR (1) =DREAL (MU)  
ROOTI (1) =DIMAG (MU)  
ROOTR (2) =DREAL (-MU)  
ROOTI (2) =DIMAG (-MU)
```

```
MU=SQRT ( (-B-SQRT (B*B-4*A*C) ) / (2*A) )  
ROOTR (3) =DREAL (MU)  
ROOTI (3) =DIMAG (MU)  
ROOTR (4) =DREAL (-MU)  
ROOTI (4) =DIMAG (-MU)
```

```
RETURN  
END
```

```

PROGRAM SOE_DOUBLE
C
C THIS IS A PROGRAM TO EVALUATE THE SHAPE OSCILLATION EQUATION FOUND IN:
C   BUBBLE PHENOMENA IN SOUND FIELDS: PART TWO BY A. PROSPERETTI
C   IN THE ACOUSTIC CAVITATION SERIES: PART THREE
C
C    $R(\theta, \phi; T) = R(T) + \sum(L, M) \{ A_{LM}(T) Y_{LM}(\theta, \phi) \}$ 
C    $R(\theta, \phi; T) = R(T) + A_N(T) P_N(\cos(\theta))$ 
C
C WHERE  $A_N$  ARE THE COEFFICIENTS TO BE CALCULATED AND  $P_N$  IS A LEGENDRE POLYNOMIAL
C WE ASSUME THAT WE HAVE AXISMETRY (IE. THE BUBBLE SHAPE IS AXISYMMETRICAL)
C
C THE COEFFICIENT EQUATION IS:
C
C    $DDA_N + [ 3DR/R + 2(N+2)(2N+1)(\mu/(\rho R^2)) ] DA_N +$ 
C    $(N-1) * [ -DDR/R + (N+1)(N+2)(\sigma/(\rho R^3)) +$ 
C    $2(N+2)(\mu DR)/(\rho R^3) ] A_N = 0$ 
C
C WHERE "D" MEANS FIRST TIME DERIVATIVE AND "DD" MEANS SECOND TIME DERIVATIVE
C
C   A_N      =      LEGENDRE COEFFICIENTS
C   MU       =      VISCOSITY OF LIQUID
C   N        =      MODE NUMBER      = 1      TRANSLATION
C                                   > 1      SHAPE MODE
C
C   R        =      MEAN BUBBLE RADIUS AS CALCULATED FROM RPE
C   RHO      =      DENSITY OF LIQUID
C   SIGMA    =      SURFACE TENSION BETWEEN LIQUID AND AIR BUBBLE
C
C TO SOLVE LET
C
C   Y3 = A_N
C   Y4 = DA_N
C
C THEN CONVERT TO 2 COUPLED FIRST ORDER EQUATIONS:
C
C   DY3 = Y4
C   DY4 = - [ 3DR/R + 2(N+2)(2N+1)(\mu/(\rho R^2)) ] Y4 -
C           (N-1) [ -DDR/R + (N+1)(N+2)(\sigma/(\rho R^3)) +
C           2(N+2)(\mu DR)/(\rho R^3) ] Y3
C
C*****
C THE RPE IS ALSO SOVLED BY 2 COUPLED FIRST ORDER EQUATIONS:
C
C SOLVE THE RAYLEIGHT-PLESSET EQUATION USING THE POLYTROPIC APPROXIMATION
C
C  $R DDR + 3/2 DR^2 = 1/\rho [ P_i - P(T) - 2\sigma/R - 4\mu/R DR ]$ 
C
C ASSUME POLYTROPIC APPROX:       $P_i = P_o(R_o/R)^{3\gamma}$ 
C                                $P(T) = P_{inf}(1 - \nu \sin(\omega T))$ 
C                                $P_o - P_{inf} = 2\sigma/R_o$  (NO TRANSFER)
C                                $R = R_o X$ 
C
C RESULTING EQUATION BECOMES:
C
C  $DDX = -1.5 DX^2/X + 1/(\rho R_o^2 X) [ P_o X^{-3\gamma} - (P_o - 2\sigma/R_o)(1 -$ 
C    $\nu \sin(\omega T)) - 2\sigma/(R_o X) - 4\mu DX/X ]$ 
C
C WHERE DX MEANS DERIVATIVE OF X WITH RESPECT TO T

```

```

C      DDX MEANS 2ND DERIVATIVE OF X WITH RESPECT TO T
C
C      CONVERT TO 2 COUPLED FIRST ORDER EQUATIONS
C
C      LET Y1 = X
C      Y2 = DX
C
C      DY1 = Y2
C      DY2 = -1.5*Y2^2/Y1 + 1/(RHO*RO^2*Y1) [PO*Y1^-3GAMMA-(PO-2SIGMA/RO) (1-
C      NU*SIN(OMEGA*T))-2SIGMA/(RO*Y1)-4MU*Y2/Y1]
C
C      WHERE
C      PINF          AMBIENT PRESSURE
C      RHO           DENSITY OF THE LIQUID
C      RO            INITIAL BUBBLE RADIUS
C      OMEGA         DRIVING FREQUENCY
C      SIGMA         INTERFACIAL TENSION (AIR BUBBLE IN WATER)
C      MU            VISCOSITY OF LIQUID
C      NU            DRIVING FREQUENCY PRESSURE AMPLITUDE (DIMENSIONLESS
C      AS A FACTOR OF PINF NORMALLY 1)
C      GAMMA         POLYTROPIC EXPONENT
C
C*****
C
C      COMBINING ALL 4 EQUATIONS GIVES:
C
C      DY1 = Y2
C      DY2 = -1.5*Y2^2/Y1 + 1/(RHO*RO^2*Y1) [PO*Y1^-3GAMMA-(PO-2SIGMA/RO) (1-
C      NU*SIN(OMEGA*T))-2SIGMA/(RO*Y1)-4MU*Y2/Y1]
C      DY3 = Y4
C      DY4 = -(3Y2/Y1+2(N+2)(2N+1)MU/(RHO*RO^2*Y1^2))Y4 - (N-1)[-DY2/Y1 +
C      (N+1)(N+2)SIGMA/(RHO*RO^3*Y1^3) + 2(N+2)MU*Y2/(RHO*RO^2*Y1^3)]Y3
C
C*****
C
C      THE INPUTS TO THIS PROGRAM CAN EITHER BE INTERACTIVELY ENTERED OR FROM
C      A BATCH FILE. THE INPUTS FROM A BATCH FILE ARE THE SAME AS THE INTERACTIVE
C      COMMANDS. THE PROGRAM WILL OUTPUT A NUMBER OF FILES DEPENDING ON THE
C      ANSWERS GIVEN IN THE INPUT.
C      (NOTE THAT "BASE" IS THE FIRST PART OF THE FILENAME GIVEN IN THE INPUT.)
C
C      FILE NAME      WHEN OUTPUT      DESCRIPTION
C      BASE.DAT        WHEN RT CURVES ARE SPECIFIED      CONTAINS THE FOLLOWING:
C                                                           T, R, V, A, DA
C                                                           T=TIME(PERIODS), R=RADIUS(CM),
C                                                           V=RADIAL VELOCITY(CM/S),
C                                                           A=SURFACE OSCILLATION AMPLITUDE
C                                                           DA=SURF. OSC. VELOCITY
C
C      BASE.DRIVE      WHEN RT CURVES ARE SPECIFIED      CONTAINS THE TIME(PERIODS), AND
C                                                           THE DRIVE AMPLITUDE IN AMBIENT
C                                                           PRESSURE UNITS
C
C      BASE.STATS      ALWAYS              CONTAINS INFORMATION ABOUT
C                                                           THE INPUTS TO THE PROGRAM
C
C      BASE.RSECT      ALWAYS              CONTAINS THE R AND DR SECTION

```

```

C                                     POINTS IN (CM) AND (CM/S) .
C                                     SECTION POINTS ARE CALCULATED
C                                     AT THE END OF EACH DRIVING
C                                     PERIOD.
C
C BASE.ASECT  WHEN MODE <> 0          CONTAINS THE A AND DA SECTION
C                                     POINTS IN ARBITRARY UNITS.
C                                     SECTION POINTS ARE CALCULATED
C                                     AT THE END OF EACH DRIVING
C                                     PERIOD.
C *****
C      IMPLICIT REAL*8 (A-Z)

      CHARACTER*1 CH
      LOGICAL BATCH, RT
      INTEGER NEQ, METHOD, POINTS, EXTEND

      COMMON /MAIN/ MU, NU, OMEGA, SIGMA, GAMMA, RHO, RO, PO, TOL, METHOD, NEQ, POINTS,
2          EXTEND, RT, PINF, FREQ
      COMMON /SIMPLE/ C1, C2, C3, C4, C5, C6, C7, C8, C9
      COMMON /INITIAL/ Y1_INIT, Y2_INIT, Y3_INIT, Y4_INIT, TMIN, TMAX, TINC, TSKIP

      TYPE 1
1      FORMAT(' INTERACTIVE OR BATCH (I/B)? 'S)
      ACCEPT '(A1)', CH
      BATCH=CH.EQ.'B'.OR.CH.EQ.'B'

      CALL READ_VALUES(BATCH)

      CALL LIB$INIT_TIMER

      CALL SOLVE

      IF(.NOT.BATCH) CALL PROMPT('DONE.', 0)

      CALL LIB$SHOW_TIMER

      IF(RT) THEN
        CLOSE(UNIT=1)
        CLOSE(UNIT=3)
      ENDIF
      CLOSE(UNIT=10)
      IF(NEQ.GT.2) CLOSE(UNIT=11)

      CALL EXIT
      END

      SUBROUTINE SOLVE
      IMPLICIT REAL*8 (A-Z)
      INTEGER NEQ, IDO, I, METHOD, IWK(4), INDEX, IER, METH, MITER, NUM_WRITE, POINTS
      INTEGER OLD_NEQ, EXTEND
      DIMENSION Y(4), PARAM(50), WK(100)
      LOGICAL STEADY, STEADY_STATE, FIRST, RT

      COMMON /MAIN/ MU, NU, OMEGA, SIGMA, GAMMA, RHO, RO, PO, TOL, METHOD, NEQ, POINTS,
2          EXTEND, RT, PINF, FREQ
      COMMON /SIMPLE/ C1, C2, C3, C4, C5, C6, C7, C8, C9
      COMMON /INITIAL/ Y1_INIT, Y2_INIT, Y3_INIT, Y4_INIT, TMIN, TMAX, TINC, TSKIP

```

```

EXTERNAL FCT,FCNJ,STEADY_STATE

DATA R_TOL/1.D-6/,V_TOL/0.1D0/

Y(1)=Y1_INIT + 1          ! INITIAL CONDITIONS (T=0)
Y(2)=Y2_INIT
IF (NEQ.GT.2) THEN
  Y(3)=Y3_INIT + 1
  Y(4)=Y4_INIT
ENDIF

DO I=1,50,1                ! DEFAULT PARAMETERS
  PARAM(I)=0.0
END DO
PARAM(4)=1.E7              ! MAX STEPS
PARAM(10)=1.0              ! ABSOLUTE ERROR CONTROL

IDO = 1
INDEX = 1
X=TMIN
H=TINC/100.
METH=2
MITER=0
NUM_WRITE=0
STEADY=.FALSE.
FIRST=.TRUE.

C FIND STEADY STATE SOLUTION FOR R

OLD_NEQ=NEQ
NEQ=2
DO T=TMIN+POINTS*TINC,TMAX,TINC*POINTS      ! SECTION POINTS ONLY
  XEND=T
  IF (METHOD.EQ.1) THEN
    CALL DIVPRK(IDO, NEQ, FCT, X, XEND, TOL, PARAM, Y)
    IF (IDO.NE.2) THEN
      TYPE *, 'ERROR, IDO=', IDO
      TYPE *, 'HTRIAL = ', PARAM(31)
      TYPE *, 'NSTEP  = ', PARAM(34)
      TYPE *, 'NFCN   = ', PARAM(35)
      DO I=1,NEQ,1
        TYPE *, 'Y = ', Y(I)
      END DO
      STOP
    ENDIF
  ELSE
    CALL DGEAR(NEQ, FCT, FCNJ, X, H, Y, XEND, TOL, METH, MITER,
2      INDEX, IWK, WK, IER)
    IF (INDEX.NE.0) THEN
      TYPE *, 'ERROR IN DGEAR, INDEX <> 0'
      TYPE *, 'XEND=', XEND
      TYPE *, 'H=', H
      STOP
    ENDIF
    IF (IER.GT.128) THEN
      TYPE *, 'ERROR IN DGEAR, IER > 128'
      TYPE *, 'IER=', IER
      TYPE *, 'XEND=', XEND
    ENDIF
  ENDIF

```

```

        TYPE *, 'H=', H
        STOP
    ENDIF
ENDIF
IF (X.GT.TSKIP) THEN
    IF (FIRST) THEN
        R_SECT_1=Y(1)
        V_SECT_1=Y(2)
        R_SECT_0=1.E30
        V_SECT_0=1.E30
        WRITE(10,*) REAL(RO*R_SECT_1), REAL(RO*V_SECT_1)
        FIRST=.FALSE.
    ELSE
        R_SECT_2=Y(1)
        V_SECT_2=Y(2)
        IF (ABS(R_SECT_1-R_SECT_2).LE.R_TOL.AND.
2      ABS(V_SECT_1-V_SECT_2).LE.V_TOL) STEADY=.TRUE.      !PERIOD 1
        IF (ABS(R_SECT_0-R_SECT_2).LE.R_TOL.AND.
2      ABS(V_SECT_0-V_SECT_2).LE.V_TOL) STEADY=.TRUE.      !PERIOD 1/2
        R_SECT_0=R_SECT_1
        V_SECT_0=V_SECT_1
        R_SECT_1=R_SECT_2
        V_SECT_1=V_SECT_2
        WRITE(10,*) REAL(RO*R_SECT_2), REAL(RO*V_SECT_2)
    ENDIF
    IF (STEADY) GOTO 99
ENDIF
END DO
IF (.NOT.STEADY) THEN
    TYPE *, 'STEADY STATE NOT REACHED'
    RETURN
ENDIF

99      CONTINUE

C STEADY STATE REACHED, NOW PERTURB A'S (IF NEEDED)

NEQ=OLD_NEQ
IDO = 1
INDEX = 1
TMAX=EXTEND*TINC*POINTS
IF (.NOT.RT) TINC=TINC*POINTS      !SECTION OUTPUT ONLY

DO T=X+TINC,X+TMAX,TINC
    XEND=T
    IF (METHOD.EQ.1) THEN
        CALL DIVPRK(IDO, NEQ, FCT, X, XEND, TOL, PARAM, Y)
        IF (IDO.NE.2) THEN
            TYPE *, 'ERROR, IDO=', IDO
            TYPE *, 'HTRIAL = ', PARAM(31)
            TYPE *, 'NSTEP  = ', PARAM(34)
            TYPE *, 'NFCN   = ', PARAM(35)
            DO I=1,NEQ,1
                TYPE *, 'Y = ', Y(I)
            END DO
            STOP
        ENDIF
    ELSE
        CALL DGEAR(NEQ, FCT, FCNJ, X, H, Y, XEND, TOL, METH, MITER,
2      INDEX, IWK, WK, IER)
    ENDIF
END DO

```

```

      IF (INDEX.NE.0) THEN
        TYPE *, 'ERROR IN DGEAR, INDEX <> 0'
        TYPE *, 'XEND=', XEND
        TYPE *, 'H=', H
        STOP
      ENDIF
      IF (IER.GT.128) THEN
        TYPE *, 'ERROR IN DGEAR, IER > 128'
        TYPE *, 'IER=', IER
        TYPE *, 'XEND=', XEND
        TYPE *, 'H=', H
        STOP
      ENDIF
      ENDIF
      IF (RT) THEN
        IF (NEQ.EQ.2) WRITE(1, *) REAL(X*FREQ), REAL(RO*Y(1)), REAL(RO*Y(2))
        IF (NEQ.EQ.4) WRITE(1, *) REAL(X*FREQ), REAL(RO*Y(1)), REAL(RO*Y(2)),
2      REAL(Y(3)), REAL(Y(4))
        WRITE(3, *) REAL(X*FREQ), REAL(PINF*NU*SIN(OMEGA*T))
        NUM_WRITE=NUM_WRITE+1
        IF (NUM_WRITE.EQ.POINTS) THEN
          NUM_WRITE=0
          WRITE(10, *) REAL(RO*Y(1)), REAL(RO*Y(2))
          IF (NEQ.GT.2) WRITE(11, *) REAL(Y(3)), REAL(Y(4))
        ENDIF
      ELSE
        WRITE(10, *) REAL(RO*Y(1)), REAL(RO*Y(2))           !SECTION R DR
        IF (NEQ.GT.2) WRITE(11, *) REAL(Y(3)), REAL(Y(4))   !SECTION A DA
      ENDIF
    END DO

    IDO = 3           !DONE
    IF (METHOD.EQ.1) CALL IVPK (IDO, NEQ, FCT, X, XEND, TOL, PARAM, Y)

    RETURN
  END

  SUBROUTINE FCT(NEQ,X,Y,DERY)
    IMPLICIT REAL*8 (A-Z)
    INTEGER NEQ,METHOD
    DIMENSION Y(NEQ),DERY(NEQ)
    COMMON /MAIN/ MU,NU,OMEGA,SIGMA,GAMMA,RHO,RO,PO,TOL,METHOD
    COMMON /SIMPLE/ C1,C2,C3,C4,C5,C6,C7,C8,C9

    C1=(2.0*SIGMA)/RO
    C2=PO-C1
    C3=4.0*MU
    C4=-3.0*GAMMA
    C5=RHO*RO*RO
    C6=2*(N+2)*(2*N+1)*MU/C5
    C7=( (N+1)*(N+2)*SIGMA)/(C5*RO)
    C8=2*(N+2)*MU/C5
    C9=N-1

    DERY(1)=Y(2)
    DERY(2)=(-1.5*Y(2)*Y(2))/Y(1)+(1.0/(C5*Y(1)))*(PO*Y(1)**C4-
2      C2*(1.0-NU*SIN(OMEGA*X))-(C1+C3*Y(2))/Y(1))
    IF (NEQ.GT.2) THEN
      DERY(3)=Y(4)

```

```

      DERY(4) = -(3.0*Y(2)/Y(1)+C6/(Y(1)*Y(1)))*Y(4)-C9*(-DERY(2)/Y(1)+
2      (C7+C8*Y(2))/Y(1)**3)*Y(3)
      ENDIF

```

```

      RETURN
      END

```

```

      SUBROUTINE FCNJ(N,X,Y,PD)
      INTEGER N
      REAL*8 Y(N),PD(N,N),X
      RETURN
      END

```

```

      SUBROUTINE READ_VALUES(BATCH)
      IMPLICIT REAL*8 (A-Z)
      REAL*4 R4
      INTEGER PERIODS,POINTS,SKIP,MODE,NEQ,METHOD,EXTEND
      CHARACTER*60 FILENAME
      CHARACTER*1 CH
      LOGICAL BATCH,RT

```

```

      EXTERNAL KAPPA

```

```

      COMMON /MAIN/ MU,NU,OMEGA,SIGMA,GAMMA,RHO,RO,PO,TOL,METHOD,NEQ,POINTS,
2      EXTEND,RT,PINF,FREQ
      COMMON /SIMPLE/ C1,C2,C3,C4,C5,C6,C7,C8,C9
      COMMON /INITIAL/ Y1_INIT,Y2_INIT,Y3_INIT,Y4_INIT,TMIN,TMAX,TINC,TSKIP

```

```

      DATA TWOPI/6.2831853/

```

```

      IF (BATCH) THEN
        TYPE 1
1      FORMAT(' ENTER INPUT DATA FILE NAME -> ',$)
        ACCEPT '(A60)',FILENAME
        OPEN(UNIT=3,NAME=FILENAME,STATUS='OLD',READONLY)
2      FORMAT(I)
3      FORMAT(G)
4      FORMAT(A1)
        READ(3,4) CH
        METHOD=0
        IF (CH.EQ.'R'.OR.CH.EQ.'r') METHOD=1
        IF (CH.EQ.'G'.OR.CH.EQ.'g') METHOD=2
        IF (METHOD.EQ.0) STOP 'INVALID METHOD SPECIFIED'
        READ(3,3) R4
        RO=DBLE(R4)
        READ(3,3) R4
        PINF=DBLE(R4)
        READ(3,3) R4
        FREQ=DBLE(R4)
        READ(3,3) R4
        RHO=DBLE(R4)
        READ(3,3) R4
        SIGMA=DBLE(R4)
        READ(3,3) R4
        MU=DBLE(R4)
        READ(3,3) R4
        NU=DBLE(R4)
        READ(3,2) PERIODS
        READ(3,2) POINTS

```

```

READ(3,2) SKIP
READ(3,3) R4
Y1_INIT=DBLE(R4)
READ(3,3) R4
Y2_INIT=DBLE(R4)
READ(3,2) MODE
READ(3,3) R4
TOL=DBLE(R4)
READ(3,2) EXTEND
READ(3,4) CH
RT=CH.EQ.'R'.OR.CH.EQ.'R'
READ(3,'(A60)') FILENAME
CLOSE(UNIT=3)
ELSE
CALL SCRINI
CALL HEADER('SHAPE OSCILLATION EQUATION SOLVER')
CALL ASK('ENTER INTEGRATION METHOD (R=RUNGE-KUTTA, G=GEAR) [R] ->',
2   CH,'R')
METHOD=1
IF(CH.EQ.'G'.OR.CH.EQ.'G') METHOD=2
CALL ASKR('ENTER INITIAL BUBBLE RADIUS (CM) [0.005] ->',R4,0.005)
RO=DBLE(R4)
CALL ASKR('ENTER AMBIENT PRESSURE (BAR) [1.01325] ->',R4,1.01325)
PINF=DBLE(R4)
CALL ASKR('ENTER DRIVING FREQUENCY (KHz) [22.22] ->',R4,
2   22.22)
FREQ=DBLE(R4)
CALL ASKR('ENTER DENSITY OF LIQUID (GRAMS/LITER) [1.0] ->',R4,1.)
RHO=DBLE(R4)
CALL ASKR('ENTER INTERFACIAL TENSION (DYNES/CM) [72.5] ->',
2   R4,72.5)
SIGMA=DBLE(R4)
CALL ASKR('ENTER VISCOSITY OF THE LIQUID (G/CM-SEC) [0.01] ->',
2   R4,0.01)
MU=DBLE(R4)
CALL ASKR('ENTER PRESSURE AMPLITUDE FACTOR (DIMENSIONLESS) '//
2   ' [1.0] ->',R4,1.)
NU=DBLE(R4)
CALL ASKI('ENTER MAXIMUM NUMBER OF DRIVING FREQUENCY PERIODS [10] ->',
2   PERIODS,10)
CALL ASKI('ENTER NUMBER OF INTEGRATION POINTS/PERIOD [100] ->',
2   POINTS,100)
CALL ASKI('ENTER NUMBER OF PERIODS TO SKIP [0] ->',SKIP,0)
CALL ASKR('ENTER INITIAL DISPLACEMENT (DIMENSIONLESS) [0.0] ->',
2   R4,0.0)
Y1_INIT=DBLE(R4)
CALL ASKR('ENTER INITIAL VELOCITY (DIMENSIONLESS) [0.0] ->',
2   R4,0.0)
Y2_INIT=DBLE(R4)
CALL ASKI('ENTER MODE NUMBER (N > 2) [2] ->',MODE,2)
IF(MODE.LE.0) THEN
CALL PROMPT('RADIUS VALUES ONLY.',0)
CALL WAIT
ENDIF
CALL ASKR('ENTER ERROR TOLERANCE (REAL) [0.0001] ->',R4,0.0001)
TOL=DBLE(R4)
CALL ASKI('ENTER NUMBER OF CYCLES TO OUTPUT AFTER STEADY
2 STATE [20] ->',EXTEND,20)

```

```

CALL ASK('RT CURVES OR SECTION CURVES (R/S) [S] ->',CH,'S')
RT=CH.EQ.'R'.OR.CH.EQ.'R'
CALL ASK('ENTER OUTPUT FILE NAME [SHAPE.DAT] ->',FILENAME,'SHAPE.DAT')
ENDIF

NEQ=4
IF (MODE.LE.0) NEQ=2
Y3_INIT=0
Y4_INIT=0

PINF=PINF*1.0D6          !CONVERT TO CGS
FREQ=FREQ*1000          !CONVERT TO HERTZ
OMEGA=TWOPI*FREQ !CONVERT TO ANGULAR RADIAN
PO=PINF+2*SIGMA/RO
TMIN=0
TMAX=(1/FREQ)*FLOAT(PERIODS)
TINC=(1/FREQ)/FLOAT(POINTS)
TSKIP=(1/FREQ)*FLOAT(SKIP)
GAMMA=KAPPA()
FREE_RES=1/(TWOPI*RO)*SQRT((3*GAMMA*PO)/RHO)

OPEN(UNIT=1,NAME=FILENAME(1:INDEX(FILENAME,'.'))// 'STATS',
2      STATUS='NEW')

LUN=1
10 IF (METHOD.EQ.1) THEN
    WRITE(LUN,*) 'USING RUNGE-KUTTA METHOD'
ELSE
    WRITE(LUN,*) 'USING GEAR METHOD'
ENDIF
WRITE(LUN,*) 'INITIAL PRESSURE (BAR) = ',REAL(PO/1.0D6)
WRITE(LUN,*) 'DENSITY OF LIQUID (GRAMS/LITER) = ',REAL(RHO)
WRITE(LUN,*) 'INITIAL BUBBLE RADIUS (CM) = ',REAL(RO)
WRITE(LUN,*) 'DRIVING FREQUENCY (KHz) = ',REAL(FREQ/1000)
WRITE(LUN,*) 'INTERFACIAL TENSION (DYNES/CM) = ',REAL(SIGMA)
WRITE(LUN,*) 'VISCOSITY OF THE LIQUID (G/CM-SEC) = ',REAL(MU)
WRITE(LUN,*) 'PRESSURE AMPLITUDE FACTOR = ',REAL(NU)
WRITE(LUN,*) 'POLYTROPIC EXPONENT = ',REAL(GAMMA)
WRITE(LUN,*) 'MAX NUMBER OF DRIVING FREQ PERIODS = ',PERIODS
WRITE(LUN,*) 'NUMBER OF INTEGRATION POINTS/PERIOD = ',POINTS
WRITE(LUN,*) 'FREE RESONANCE FREQUENCY (KHz) = ',REAL(FREE_RES/1000)
WRITE(LUN,*) 'DRIVING FREQ / RESONANCE FREQ = ',REAL(FREQ/FREE_RES)
WRITE(LUN,*) 'INITIAL DISPLACEMENT FROM EQUIL = ',REAL(Y1_INIT)
WRITE(LUN,*) 'INITIAL VELOCITY = ',REAL(Y2_INIT)
WRITE(LUN,*) 'SHAPE MODE NUMBER = ',MODE
WRITE(LUN,*) 'ERROR TOLERANCE = ',REAL(TOL)
WRITE(LUN,*) 'DATA FILE NAME = '//FILENAME(1:20)

IF (LUN.EQ.1) THEN
    WRITE(LUN,*) 'EXTENDED CYCLES = ',EXTEND
    IF (RT) WRITE(LUN,*) 'RT CURVES SPECIFIED'
    WRITE(LUN,*) '*****'
    WRITE(LUN,*) 'TMIN, TMAX, TINC ',REAL(TMIN),REAL(TMAX),REAL(TINC)
    CLOSE(UNIT=1)
    IF (.NOT.BATCH) THEN
        LUN=5
        CALL HEADER('SHAPE OSCILLATION EQUATION SOLVER')
        GOTO 10
    
```

```

ENDIF
ENDIF

IF (.NOT. BATCH) CALL PROMPT('PROCESSING ...', 0)

C1=(2*SIGMA)/RO
C2=PO-C1
C3=4*MU
C4=-3*GAMMA
C5=RHO*RO*RO
C6=2*(MODE+2)*(2*MODE+1)*MU/C5
C7=((MODE+1)*(MODE+2)*SIGMA)/(C5*RO)
C8=2*(MODE+2)*MU/C5
C9=MODE-1

IF (RT) OPEN (UNIT=1, NAME=FILENAME, STATUS='NEW')
OPEN (UNIT=2, NAME=FILENAME(1:INDEX(FILENAME, '.'))// 'DGEAR',
2      STATUS='NEW')
IF (RT) OPEN (UNIT=3, NAME=FILENAME(1:INDEX(FILENAME, '.'))// 'DRIVE',
2      STATUS='NEW')
OPEN (UNIT=10, NAME=FILENAME(1:INDEX(FILENAME, '.'))// 'RSECT',
2      STATUS='NEW')
IF (NEQ.GT.2) OPEN (UNIT=11, NAME=FILENAME(1:INDEX(FILENAME, '.'))//
2      'ASECT', STATUS='NEW')

RETURN
END

SUBROUTINE ASK(PROMP, RET_VAL, DEF_VAL)
CHARACTER*(*) PROMP, RET_VAL, DEF_VAL
CHARACTER*80 STR
EXTERNAL INPUT

CALL PROMPT(PROMP, 0)
NCHAR=INPUT(STR, 80)

IF (NCHAR.EQ.0) THEN
    RET_VAL=DEF_VAL
ELSE
    RET_VAL=STR(1:NCHAR)
ENDIF

RETURN
END

SUBROUTINE ASKI(PROMP, RET_VAL, DEF_VAL)
CHARACTER*(*) PROMP
INTEGER RET_VAL, DEF_VAL
CHARACTER*80 STR

CALL PROMPT(PROMP, 0)
NCHAR=INPUT(STR, 80)

DO WHILE (INDEX(STR, '.').NE.0)
    CALL ERRMSG('INVALID VALUE, ENTER AN INTEGER')
    CALL PROMPT(PROMP, 0)
    NCHAR=INPUT(STR, 80)
END DO

```

```

      IF (NCHAR.EQ.0) THEN
        RET_VAL=DEF_VAL
      ELSE
        DECODE (NCHAR,1,STR) RET_VAL
1      FORMAT (I)
      ENDIF

      RETURN
      END

      SUBROUTINE ASKR (PROMP, RET_VAL, DEF_VAL)
      CHARACTER* (40) PROMP
      REAL RET_VAL, DEF_VAL
      CHARACTER*80 STR

      CALL PROMPT (PROMP, 0)
      NCHAR=INPUT (STR, 80)

      DO WHILE (INDEX (STR, '.') .EQ.0 .AND. NCHAR.NE.0)
        CALL ERRMSG ('INVALID VALUE, INCLUDE DECIMAL POINT')
        CALL PROMPT (PROMP, 0)
        NCHAR=INPUT (STR, 80)
      END DO

      IF (NCHAR.EQ.0) THEN
        RET_VAL=DEF_VAL
      ELSE
        DECODE (NCHAR,1,STR) RET_VAL
1      FORMAT (G)
      ENDIF

      RETURN
      END

      SUBROUTINE PROMPT (STR, NUM)
C
C CALL:          CALL PROMPT (STR, NUM)
C      STR      ASCII MESSAGE STRING
C      NUM      NUMBER OF DASHES TO PUT AFTER THE PROMPT
C THIS ROUTINE WILL OUTPUT A MESSAGE ON LINE 22 IN INVERSE SCREEN
C FORMAT. THE CURSOR WILL REMAIN AT THE END OF THE MESSAGE UPON RETURN.
C
      CHARACTER* (40) STR
      BYTE ESC
      INTEGER NUM
      DATA ESC/27/

      TYPE 100, ESC, ESC, ESC
100    FORMAT ('+A1'[22H'A1'[J'A1'[7M'S)
      DO I=1, LEN (STR), 1
        TYPE 110, STR (I:I)
      END DO
      TYPE 110, ESC, '[', 'M'
110    FORMAT ('+A1$)

      IF (%LOC (NUM) .EQ.0) GOTO 200
      IF (NUM.LE.0) GOTO 200

```

```

DO 120 I=1,NUM,1
120  TYPE 110,'_'
    CALL POS(22,LEN(STR))

```

```

200  RETURN
    END

```

```

SUBROUTINE ERRMSG(STR)

```

```

C CALL:          CALL ERRMSG(STR)
C   STR      ASCIZ STRING MESSAGE
C THIS ROUTINE WILL OUTPUT AN ERROR MESSAGE ON LINE 23 AND WAIT 3
C SECONDS BEFORE ERASING THE ERROR MESSAGE AND RETURNING. THE MESSAGE
C WILL BE OUTPUT IN INVERSE SCREEN FORM.
C

```

```

    CHARACTER*(*) STR
    BYTE ESC
    DATA ESC/27/

```

```

    TYPE 100,ESC,ESC,ESC
100  FORMAT('+'A1'[23H'A1'[J'A1'[7M'S)
    DO I=1,LEN(STR),1
        TYPE 110,STR(I:I)
    END DO
    TYPE 110,ESC,['','M'
110  FORMAT('+'A1$)
    CALL WAIT

```

```

    RETURN
    END

```

```

SUBROUTINE HEADER(STR)

```

```

C
C CALL:          CALL HEADER(STR)
C   STR      ASCIZ MESSAGE STRING
C THIS ROUTINE WILL CLEAR THE SCREEN AND OUTPUT A TITLE IN DOUBLE
C SIZE LETTERS ON THE FIRST TWO LINES OF THE SCREEN. THE CURSOR WILL
C BE LEFT ON THE 3RD ROW COLUMN 1.
C

```

```

    CHARACTER*(*) STR
    BYTE ESC
    DATA ESC/27/

```

```

    TYPE 100,ESC,ESC,ESC
100  FORMAT('+'A1'[H'A1'[J'A1'#3'S)
    DO I=1,LEN(STR),1
        TYPE 105,STR(I:I)
105  FORMAT('+'A1$)
    END DO
    TYPE *
    TYPE 110,ESC
110  FORMAT('+'A1'#4'S)
    DO I=1,LEN(STR),1
        TYPE 105,STR(I:I)
    END DO
    TYPE *

```

```

RETURN
END

```

```

INTEGER FUNCTION INPUT (BUF, MAXCNT)
C
C CALL      NCHAR=INPUT (BUF, MAXCNT)
C
C NCHAR  NUMBER OF CHARACTERS ACTUALLY INPUT.
C        IF NCHAR=0 THEN NUMBER OF CHARACTERS INPUT EXCEEDS
C        MAXCNT.
C BUF    STORAGE AREA FOR RETURNED CHARACTERS, WILL BE PADDED
C        WITH TRAILING BLANKS UP TO MAXCNT.
C MAXCNT MAXIMUM NUMBER OF CHARACTERS TO ACCEPT.
C
C CHARACTER* (1) BUF
C CHARACTER*512 INPUT_LINE
C INTEGER MAXCNT, 1, LINE_SIZE, STATUS, NCHAR

INPUT_LINE=' '
STATUS=LIB$GET_INPUT (INPUT_LINE, LINE_SIZE)
IF (.NOT. STATUS) CALL LIB$SIGNAL (%VAL (STATUS))

BUF=INPUT_LINE (1:MAXCNT)

NCHAR=LINE_SIZE
IF (BUF (1:1) .EQ. CHAR (3)) NCHAR=-3      ! ^C
IF (BUF (1:1) .EQ. CHAR (27)) NCHAR=-27    ! <ESC>
IF (LINE_SIZE.GT.MAXCNT) NCHAR=0          ! TOO MANY CHARACTERS

INPUT=NCHAR

RETURN
END

REAL*8 FUNCTION KAPPA

IMPLICIT REAL*16 (A-Z)

REAL*8 MU8, NU8, OMEGA8, SIGMA8, GAMMA8, RHO8, RO8
COMMON /MAIN/ MU8, NU8, OMEGA8, SIGMA8, GAMMA8, RHO8, RO8

R=QEXT (RO8)          ! CM          ! BUBBLE RADIUS
U=QEXT (MU8)          ! G/CM S      ! VISCOSITY OF WATER
RHO=QEXT (RHO8)        ! G/CM^3      ! DENSITY OF WATER
OMEGA=QEXT (OMEGA8)    ! HERTZ      ! ANGULAR DRIVING FREQ

C CONSTANTS

RHO_G=0.001204Q0      ! G/CM^3      ! DENSITY OF AIR
C=1.5Q5               ! CM/S        ! SOUND SPEED IN WATER
RG=8.314Q7            ! DYN/MOL K   ! GAS CONSTANT
M=28.964Q0            ! G/MOL       ! MOLECULAR WEIGHT AIR
ML=18.02Q0            ! G/MOL       ! MOLECULAR WEIGHT WATER
TINF=293Q0            ! K           ! AMBIENT TEMPERATURE
CV_G=20.8Q7           ! DYN/MOL K   ! SPECIFIC HEAT AIR (V)

```

```

CP_G=2.91Q8                !DYN/MOL K      !SPECIFIC HEAT AIR (P)
CV_L=76.0Q7                !DYN/MOL K      !SPECIFIC HEAT WATER (V)
CP_L=CV_L                  !DYN/MOL K      !SPECIFIC HEAT WATER (P)
GAMMA=CP_G/CV_G            !UNITLESS
K_L=5.92Q4                 !DYN/(S CM K)  !THERMAL CONDUCTIVITY
K_G=2.5Q3                  !DYN/(S CM K)  ! " AIR

K=K_L/K_G                  !UNITLESS
CG=SQRT((GAMMA*RG/M)*TINF) !CM/S          !SOUND SPEED IN AIR
DG=K_G/(RHO_G*CP_G)*M      !CM^2/S        !THERM. DIFFUS. AIR
DT=K_L/(RHO*CP_L)*ML       !CM^2/S        !THERM. DIFFUS. WATER

X=R*SQRT(2*OMEGA/DG)
A1= SINH(X)-SIN(X)
A2= SINH(X)+SIN(X)
A3= COSH(X)-COS(X)

DTH=3*(GAMMA-1)*((X*A2-2*A3)/(X*X*A3+3*(GAMMA-1)*X*A1))
KAPPA=DBLE(GAMMA/(1+DTH*DTH)/(1+3*(GAMMA-1)/X*A1/A3))

RETURN
END

LOGICAL FUNCTION STEADY_STATE(X,Y,NUM_POINTS_CYCLE)
IMPLICIT REAL*8 (A-Z)
DIMENSION Y(1)
INTEGER NUM_POINTS_CYCLE, NUM
COMMON /MAIN/ MU, NU, OMEGA, SIGMA, GAMMA, RHO, RO
DATA NUM/0/, R_TOL/1.D-6/, V_TOL/0.1D0/

STEADY_STATE=.FALSE.

IF (NUM.EQ.NUM_POINTS_CYCLE*2) THEN
  STEADY_STATE=ABS(R_SECT_1-R_SECT_2).LE.R_TOL.AND.
  2 ABS(V_SECT_1-V_SECT_2).LE.V_TOL
  NUM=NUM_POINTS_CYCLE
  R_SECT_1=R_SECT_2
  V_SECT_1=V_SECT_2
ENDIF

NUM=NUM+1
IF (NUM.EQ.NUM_POINTS_CYCLE) THEN
  R_SECT_1=Y(1)
  V_SECT_1=Y(2)
  WRITE(10,*) REAL(RO*R_SECT_1), REAL(RO*V_SECT_1)
ELSE IF (NUM.EQ.2*NUM_POINTS_CYCLE) THEN
  R_SECT_2=Y(1)
  V_SECT_2=Y(2)
  WRITE(10,*) REAL(RO*R_SECT_2), REAL(RO*V_SECT_2)
ENDIF

RETURN
END

```

```

PROGRAM MIENOS(INPUT,OUTPUT,S1,S2,TAPE5=INPUT,TAPE6=OUTPUT,
$ TAPE10=S1,TAPE11=S2)
PARAMETER ( MAXANG = 2000, MOMDIM = 1 )
LOGICAL SELECT( 8 ), ANYANG, PERFCT, PRNT( 2 )
INTEGER IPOLZN, NUMMOM
REAL MIMCUT, PMOM( 0:MOMDIM, 4 ), XMU( MAXANG )
COMPLEX CREFIN, SFORW, SBACK, S1( MAXANG ), S2( MAXANG ),
$ TFORW( 2 ), TBACK( 2 )
-----
C
C
C DATA SELECT / 8*.TRUE. /, PRNT / .TRUE.,.FALSE. /
C
C
100 PRINT *, 'MIE SCATTERING'
PRINT *
PRINT *, '1. I VS ANGLE'
PRINT *, '2. I VS RADIUS'
PRINT *, '3. EXIT'
PRINT *, 'ENTER OPTION'
READ *, IOPT
IF( IOPT.EQ.3 ) STOP
IF( IOPT.EQ.1 ) THEN
PRINT *, 'ENTER RADIUS IN MICRONS (REAL)'
READ *, RADIUS
RADIUS=RADIUS*1.E-6
PRINT *, 'ENTER NUMBER OF ANGLES (INTEGER, NUMANG)'
ACCEPT *, NUMANG
ELSE IF( IOPT.EQ.2 ) THEN
PRINT *, 'ENTER ANGLE (REAL, DEGREES)'
READ *, ANGLE
NUMANG=1
ELSE
PRINT *, 'ENTER 1,2 OR 3'
GOTO 100
ENDIF
C
PI = 2. * ASIN(1.0)
PERFCT = .FALSE.
MIMCUT = 1.E-6
NUMMOM = 1
IPOLZN = - 4
ANYANG = .TRUE.
WAVELN=488.0E-9
CREFIN=(0.75187,0.0)
C
IF ( NUMANG.GT.MAXANG ) STOP'NUMANG IS TOO LARGE'
C
IF(NUMANG.NE.1)THEN
DO 1 I = 1, NUMANG
XMU( I ) = COS( (I-1) * PI / ( NUMANG-1 ) )
CONTINUE
1 XX=2.0*PI*1.33*RADIUS/WAVELN
PRNT(1)=.TRUE.
CALL MIEVO ( XX, CREFIN, PERFCT, MIMCUT, SELECT, ANYANG,
$ NUMANG, XMU, NUMMOM, IPOLZN, MOMDIM, PRNT,
$ QEXT, QSCA, QOSC, PMOM, SFORW, SBACK, S1,
$ S2, TFORW, TBACK )

```

```

ELSE
  XMU(1)=COS(ANGLE*(2.0*PI/360.))
  PRNT(1)=.FALSE.
  DO 20 I = 1, 1000, 1
    RADIUS=I*1.E-7
    XX=2.0*PI*1.33*RADIUS/WAVELN
    CALL MIEVO ( XX, CREFIN, PERFCT, MIMCUT, SELECT, ANYANG,
$              NUMANG, XMU, NUMMOM, IPOLZN, MOMDIM, PRNT,
$              QEXT, QSCA, GQSC, PMOM, SFORW, SBACK, S1,
$              S2, TFORW, TBACK )
    WRITE(10,19)RADIUS,REAL(S1(1))**2+AIMAG(S1(1))**2
    WRITE(11,19)RADIUS,REAL(S2(1))**2+AIMAG(S2(1))**2
19  FORMAT(2G16.8)
20  CONTINUE
ENDIF
C
  TYPE 888,CHAR(7)
888  FORMAT(' 'A1)
  STOP
  END
  SUBROUTINE MIEVO ( XX, CREFIN, PERFCT, MIMCUT, SELECT, ANYANG,
$                  NUMANG, XMU, NMOM, IPOLZN, MOMDIM, PRNT,
$                  QEXT, QSCA, GQSC, PMOM, SFORW, SBACK, S1,
$                  S2, TFORW, TBACK )
C
C  COMPUTES MIE SCATTERING AND EXTINCTION EFFICIENCIES; ASYMMETRY
C  FACTOR; FORWARD- AND BACKSCATTER AMPLITUDE; SCATTERING
C  AMPLITUDES FOR INCIDENT POLARIZATION BOTH PARALLEL AND PERPENDICULAR
C  TO THE PLANE OF SCATTERING, AS FUNCTIONS OF SCATTERING ANGLE;
C  COEFFICIENTS IN THE LEGENDRE POLYNOMIAL EXPANSIONS OF EITHER THE
C  UNPOLARIZED PHASE FUNCTION OR THE POLARIZED PHASE MATRIX;
C  AND SOME QUANTITIES NEEDED IN POLARIZED RADIATIVE TRANSFER.
C
C  CALLS :  CONFRA, SMALL1, SMALL2, TESTMI, MIPRNT,
C           LPCOEF, ERRMSG, WRTBAD
C
C  I N T E R N A L   V A R I A B L E S
C  -----
C  AN,BN          MIE COEFFICIENTS  LITTLE-A-SUB-N, LITTLE-B-SUB-N
C                  ( REF. 1, EQ. 16 )
C  ANM1,BNM1      MIE COEFFICIENTS  LITTLE-A-SUB-(N-1),
C                  LITTLE-B-SUB-(N-1); USED IN -GQSC- SUM
C  ANP            COEFFS. IN S+ EXPANSION ( REF. 2, P. 1507 )
C  BNP            COEFFS. IN S- EXPANSION ( REF. 2, P. 1507 )
C  ANPM           COEFFS. IN S+ EXPANSION ( REF. 2, P. 1507 )
C                  WHEN MU IS REPLACED BY - MU
C  BNPM           COEFFS. IN S- EXPANSION ( REF. 2, P. 1507 )
C                  WHEN MU IS REPLACED BY - MU
C  CBIGA(N)       BESSEL FUNCTION RATIO CAPITAL-A-SUB-N (REF. 2, EQ. 2)
C                  ( COMPLEX VERSION )
C  CIOR            COMPLEX INDEX OF REFRACTION WITH NEGATIVE
C                  IMAGINARY PART (VAN DE HULST CONVENTION)
C  CIORIV          1 / CIOR
C  COEFF           ( 2N + 1 ) / ( N ( N + 1 ) )
C  CONFRA          VALUE OF LENTZ CONTINUED FRACTION FOR -CBIGA(NTRM)-,
C                  USED TO INITIALIZE DOWNWARD RECURRENCE.

```

```

C CTMP      (COMPLEX) TEMPORARY VARIABLE
C F1,F2,F3  ARITHMETIC STATEMENT FUNCTIONS USED IN DETERMINING
C           WHETHER TO USE UP- OR DOWN-RECURRENCE FOR 'BIGA'
C           ( REF. 2, EQS. 6-8 )
C FN        FLOATING POINT VERSION OF INDEX IN LOOP PERFORMING
C           MIE SERIES SUMMATION
C LITA,LITB(N) MIE COEFFICIENTS -AN-, -BN-, SAVED IN ARRAYS FOR
C           USE IN CALCULATING LEGENDRE MOMENTS *PMOM*
C MAXTRM     MAX. POSSIBLE NO. OF TERMS IN MIE SERIES
C MM         + 1 AND - 1, ALTERNATELY.
C MIM        MAGNITUDE OF IMAGINARY REFRACTIVE INDEX
C MRE        REAL PART OF REFRACTIVE INDEX
C MAXANG     MAX. POSSIBLE VALUE OF INPUT VARIABLE -NUMANG-
C NANGD2     (NUMANG+1)/2 ( NO. OF ANGLES IN 0-90 DEG; ANYANG=F )
C NOABS      TRUE, NO ABSORPTION. CAN OCCUR EITHER BECAUSE
C           PERFCT=TRUE OR BECAUSE IMAG. REFRAC. INDEX=0.
C NP1DN      ( N + 1 ) / N
C NTRM       NO. OF TERMS IN MIE SERIES
C PASS1      TRUE ON FIRST ENTRY, FALSE THEREAFTER
C PIN(J)     ANGULAR FUNCTION LITTLE-PI-SUB-N ( REF. 2, EQ. 3 )
C           AT J-TH ANGLE
C PINM1(J)   LITTLE-PI-SUB-(N-1) ( SEE -PIN- ) AT J-TH ANGLE
C PSINM1     RICATTI-BESSEL FUNCTION PSI-SUB-(N-1) OF ARGUMENT -XX-
C PSIN       RICATTI-BESSEL FUNCTION PSI-SUB-N OF ARGUMENT -XX-
C           ( REF. 1, P. 11 FF. )
C RBIGA(N)   BESSEL FUNCTION RATIO CAPITAL-A-SUB-N (REF. 2, EQ. 2)
C           ( REAL VERSION, FOR WHEN IMAG REFRAC INDEX = 0 )
C REZINV     1 / ( MRE * XX )
C RIORIV     1 / MRE
C RN         1 / N
C RTMP       (REAL) TEMPORARY VARIABLE
C SAVINP     INTERNAL FILE USED FOR STORING USER INPUT VARIABLES
C           ON FIRST ENTRY
C SP(J)      S+ FOR J-TH ANGLE ( REF. 2, P. 1507 )
C SM(J)      S- FOR J-TH ANGLE ( REF. 2, P. 1507 )
C SPS(J)     S+ FOR (NUMANG+1-J)-TH ANGLE ( ANYANG=FALSE )
C SMS(J)     S- FOR (NUMANG+1-J)-TH ANGLE ( ANYANG=FALSE )
C TAUN       ANGULAR FUNCTION LITTLE-TAU-SUB-N ( REF. 2, EQ. 4 )
C           AT J-TH ANGLE
C TCOEF      N ( N+1 ) ( 2N+1 ) (FOR SUMMING TFORW,TBACK SERIES)
C TWONP1     2N + 1
C YESANG     TRUE IF NUMANG .GT. 0
C ZETNM1     RICATTI-BESSEL FUNCTION ZETA-SUB-(N-1) OF ARGUMENT
C           -XX- ( REF. 2, EQ. 17 )
C ZETN       RICATTI-BESSEL FUNCTION ZETA-SUB-N OF ARGUMENT -XX-
C ZINV       1 / ( CIOR * XX )

```

```

C ----- I / O SPECIFICATIONS FOR SUBROUTINES MIEVO, MIEVI -----

```

```

C LOGICAL ANYANG, PERFCT, PRNT( * ), SELECT( * )
C INTEGER IPOLZN, MOMDIM, NUMANG, NMOM
C REAL    GQSC, MIMCUT, PMOM( 0:MOMDIM, * ), QEXT, QSCA,
C S       XMU( * ), XX
C COMPLEX CREFIN, SFORW, SBACK, S1( * ), S2( * ), TFORW( * ),
C S       TBACK( * )
C -----

```

```

C      PARAMETER ( MAXANG = 1501, MXANG2 = MAXANG/2 + 1 )
C      ** NOTE -- MAXTRM = 10100 IS NECES-
C      ** SARY TO DO SOME OF THE TEST PROBS;

C      PARAMETER ( MAXTRM = 3000 )
C      PARAMETER ( ONETHR = 1./3. )

C      CHARACTER*120 SAVINP
C      LOGICAL   DOWN, INPERR, NOABS, OK, PASS1, YESANG
C      REAL      MIM, MRE, MM, NP1DN
C      REAL      RBIGA( MAXTRM ), PIN( MAXANG ), PINM1( MAXANG )
C      COMPLEX   CONFRA
C      COMPLEX   AN, BN, ANM1, BNM1, ANP, BNP, ANPM, BNPM,
$      CIOR, CIORIV, CTMP, ZET, ZETNM1, ZETN, ZINV
C      COMPLEX   CBIGA( MAXTRM ), LITA( MAXTRM ), LITB( MAXTRM ),
$      SP( MAXANG ), SM( MAXANG ), SPS( MXANG2 ), SMS( MXANG2 )
C      EQUIVALENCE ( CBIGA, RBIGA )
C      SAVE PASS1

C      SQ( CTMP ) = REAL( CTMP )**2 + AIMAG( CTMP )**2
C      F1( MRE ) = - 8.0 + MRE**2 * ( 26.22 + MRE * ( - 0.4474
$      + MRE**3 * ( 0.00204 - 0.000175 * MRE ) ) )
C      F2( MRE ) = 3.9 + MRE * ( - 10.8 + 13.78 * MRE )
C      F3( MRE ) = - 15.04 + MRE * ( 8.42 + 16.35 * MRE )

C      DATA PASS1 / .TRUE. /

C
C      ** TEST FOR BAD INPUT
C
C      IF ( NUMANG.GT.MAXANG ) THEN
C          CALL ERRMSG( 'MIEVO--PARAMETER MAXANG TOO SMALL', .TRUE. )
C          INPERR = .TRUE.
C      END IF
C      IF ( NUMANG.LT.0 ) CALL WRTBAD( 'NUMANG', INPERR )
C      IF ( XX.LT.0 .OR. XX.GT.1.E+5 ) CALL WRTBAD( 'XX', INPERR )
C      IF ( .NOT.PERFCT .AND. REAL(CREFIN).LE.0. )
$      CALL WRTBAD( 'CREFIN', INPERR )
C      IF ( MOMDIM.LT.1 ) CALL WRTBAD( 'MOMDIM', INPERR )
C      IF ( SELECT(7) ) THEN
C          IF ( NMOM.LT.0 .OR. NMOM.GT.MOMDIM ) CALL WRTBAD( 'NMOM', INPERR )
C          IF ( ABS(IPOLZN).GT.4 ) CALL WRTBAD( 'IPOLZN', INPERR )
C      END IF

C      IF ( NUMANG .GT. 0 ) THEN
C          IF ( ANYANG ) THEN
C              DO 1 I = 1, NUMANG
C                  IF ( XMU(I).LT.-1.00001 .OR. XMU(I).GT.1.00001 )
$                  CALL WRTBAD( 'XMU', INPERR )
C              CONTINUE
C          ELSE
C              DO 2 I = 1, ( NUMANG + 1 ) / 2
C                  IF ( XMU(I).LT.-0.00001 .OR. XMU(I).GT.1.00001 )
$                  CALL WRTBAD( 'XMU', INPERR )
C              CONTINUE
C          END IF
C      END IF

```

```

      IF ( INPERR )
      $    CALL ERRMSG( 'MIEVO--INPUT ERROR(S).  ABORTING...', .TRUE. )
C
C
      IF ( PASS1 ) THEN
C                                     ** SAVE USER INPUT VALUES
C
      WRITE ( SAVINP, 2001, ERR=990 ) XX, CREFIN, MIMCUT, PERFCT,
      $    SELECT(7), ANYANG, NMOM, IPOLZN, NUMANG, XMU(1)
C
C                                     ** RESET INPUT VALUES FOR TEST CASE
C
      XX      = 10.0
      CREFIN  = ( 1.5, - 0.1 )
      PERFCT  = .FALSE.
      MIMCUT  = 0.0
      SELECT( 7 ) = .TRUE.
      ANYANG  = .TRUE.
      NUMANG  = 1
      XMU( 1 ) = - 0.7660444
      NMOM    = 1
      IPOLZN  = - 1
C
      END IF
C
C
10 IF ( PERFCT ) THEN
C
      NOABS = .TRUE.
C
      IF ( XX .LE. 0.1 ) THEN
C                                     ** USE TOTALLY-REFLECTING
C                                     ** SMALL-PARTICLE LIMIT
C
      CALL SMALL1 ( XX, NUMANG, XMU, QEXT, QSCA, QQSC, SFORM,
      $    SBACK, S1, S2, TFORM, TBACK, LITA, LITB )
      NTRM = 2
      GO TO 200
C
      END IF
C
      ELSE
C
      CIOR = CREFIN
      IF ( AIMAG( CIOR ) .GT. 0.0 ) CIOR = CONJG( CIOR )
      MRE = REAL( CIOR )
      MIM = - AIMAG( CIOR )
      NOABS = MIM .LE. MIMCUT
      CIORIV = 1.0 / CIOR
      RIORIV = 1.0 / MRE
C
      IF ( XX * AMAX1( 1.0, CABS(CIOR) ) .LE. 0.1 ) THEN
C
C                                     ** USE GENERAL-REFRACTIVE-INDEX
C                                     ** SMALL-PARTICLE LIMIT
C                                     ** ( REF. 2, P. 1508 )
C
      CALL SMALL2 ( XX, CIOR, .NOT.NOABS, NUMANG, XMU, QEXT,

```

```

$          QSCA, QQSC, SFORW, SBACK, S1, S2, TFORW,
$          TBACK, LITA, LITB )
      NTRM = 2
      GO TO 200
C
      END IF
C
      END IF
C
      NANGD2 = ( NUMANG + 1 ) / 2
      YESANG = NUMANG .GT. 0
C          ** ESTIMATE NUMBER OF TERMS IN MIE SERIES
C          ** ( REF. 2, P. 1508 )
      IF ( XX.LE.8.0 ) THEN
        NTRM = XX + 4. * XX**ONETHR + 1.
      ELSE IF ( XX.LT.4200. ) THEN
        NTRM = XX + 4.05 * XX**ONETHR + 2.
      ELSE
        NTRM = XX + 4. * XX**ONETHR + 2.
      END IF
      IF ( NTRM+1 .GT. MAXTRM )
$      CALL ERRMSG( 'MIEV0--PARAMETER MAXTRM TOO SMALL', .TRUE. )
C
C
      IF ( PERFCT ) GO TO 55
C
C ----- BEGIN COMPUTATION OF 'BIGA' -----
C
C          ** DECIDE WHETHER 'BIGA' CAN BE
C          ** CALCULATED BY UP-RECURRENCE
      IF ( MRE.LT.1.0 ) THEN
        DOWN = .TRUE.
      ELSE IF ( YESANG ) THEN
        DOWN = .TRUE.
      IF ( MIM*XX .LT. F2( MRE ) ) DOWN = .FALSE.
      ELSE
        DOWN = .TRUE.
      IF ( MIM*XX .LT. F1( MRE ) ) DOWN = .FALSE.
      END IF
C
      ZINV = 1.0 / ( CIOR * XX )
      REZINV = 1.0 / ( MRE * XX )
      IF ( DOWN ) THEN
C          ** COMPUTE INITIAL HIGH-ORDER 'BIGA' USING
C          ** LENTZ METHOD ( REF. 1, PP. 17-20 )
        CTMP = CONFRA( NTRM, ZINV, XX )
C
C          *** DOWNWARD RECURRENCE FOR 'BIGA'
C          *** ( REF. 1, EQ. 22 )
      IF ( NOABS ) THEN
C          ** NO-ABSORPTION CASE
        RBIGA( NTRM ) = REAL( CTMP )
        DO 25 N = NTRM, 2, -1
          RBIGA( N-1 ) = (N*REZINV
$              - 1.0 / ( (N*REZINV) + RBIGA( N ) )
25      CONTINUE

```

```

C
C      ELSE
C
C          ** ABSORPTIVE CASE
C          CBIGA( NTRM ) = CTMP
C          DO 30 N = NTRM, 2, - 1
C              CBIGA( N-1 ) = ( N*ZINV ) - 1.0 / ( ( N*ZINV ) + CBIGA( N ) )
30      CONTINUE
C
C      END IF
C
C      ELSE
C
C          *** UPWARD RECURRENCE FOR 'BIGA'
C          *** ( REF. 1, EQS. 20-21 )
C
C          IF ( NOABS ) THEN
C
C              ** NO-ABSORPTION CASE
C              RTMP = SIN( MRE*XX )
C              RBIGA( 1 ) = - REZINV
C              $          + RTMP / ( RTMP*REZINV - COS( MRE*XX ) )
C              DO 40 N = 2, NTRM
C                  RBIGA( N ) = - ( N*REZINV )
C                  $          + 1.0 / ( ( N*REZINV ) - RBIGA( N-1 ) )
40      CONTINUE
C
C      ELSE
C
C          ** ABSORPTIVE CASE
C
C          CTMP = CEXP( - (0.,2.) * CIOR * XX )
C          CBIGA( 1 ) = - ZINV + (1.-CTMP) /
C          $          ( ZINV * (1.-CTMP) - (0.,1.)*(1.+CTMP) )
C          DO 50 N = 2, NTRM
C              CBIGA( N ) = - ( N*ZINV ) + 1.0 / ( ( N*ZINV ) - CBIGA( N-1 ) )
50      CONTINUE
C          END IF
C      END IF
C
C      ----- END OF 'BIGA' COMPUTATION -----
C
C
C          ** INITIALIZE RICATTI-BESSEL FUNCTIONS
C          ** (PSI,CHI,ZETA)-SUB-(0,1) FOR UPWARD
C          ** RECURRENCE ( REF. 1, EQ. 19 )
C
C      55 XINV = 1.0 / XX
C      PSINM1 = SIN( XX )
C      CHINM1 = COS( XX )
C      PSIN = PSINM1 * XINV - CHINM1
C      CHIN = CHINM1 * XINV + PSINM1
C      ZETNM1 = CMPLX( PSINM1, CHINM1 )
C      ZETN = CMPLX( PSIN, CHIN )
C
C          ** INITIALIZE PREVIOUS COEFFI-
C          ** CIENTS FOR -GQSC- SERIES
C
C      ANM1 = ( 0.0, 0.0 )
C      BNM1 = ( 0.0, 0.0 )
C
C          ** INITIALIZE ANGULAR FUNCTION LITTLE-PI
C          ** AND SUMS FOR S+, S- ( REF. 2, P. 1507 )
C
C      IF ( ANYANG ) THEN
C          DO 60 J = 1, NUMANG
C              PINM1( J ) = 0.0

```

```

        PIN( J ) = 1.0
        SP ( J ) = ( 0.0, 0.0 )
        SM ( J ) = ( 0.0, 0.0 )
60    CONTINUE
    ELSE
        DO 70 J = 1, NANGD2
            PINM1( J ) = 0.0
            PIN( J ) = 1.0
            SP ( J ) = ( 0.0, 0.0 )
            SM ( J ) = ( 0.0, 0.0 )
            SPS( J ) = ( 0.0, 0.0 )
            SMS( J ) = ( 0.0, 0.0 )
70    CONTINUE
    END IF
C          ** INITIALIZE MIE SUMS FOR EFFICIENCIES, ETC.
QSCA = 0.0
QQSC = 0.0
SFORW = ( 0., 0. )
SBACK = ( 0., 0. )
TFORW( 1 ) = ( 0., 0. )
TBACK( 1 ) = ( 0., 0. )
C
C
C ----- LOOP TO SUM MIE SERIES -----
C
MM = + 1.0
DO 100 N = 1, NTRM
C          ** COMPUTE VARIOUS NUMERICAL COEFFICIENTS
    FN = N
    RN = 1.0 / FN
    NP1DN = 1.0 + RN
    TWONP1 = N + ( N + 1 )
    COEFF = TWONP1 / ( FN * ( N + 1 ) )
    TCOEF = TWONP1 * ( FN * ( N + 1 ) )
C
C          ** CALCULATE MIE SERIES COEFFICIENTS
    IF ( PERFCT ) THEN
C          ** TOTALLY-REFLECTING CASE
        AN = ( ( FN*XINV ) * PSIN - PSINM1 ) /
$          ( ( FN*XINV ) * ZETN - ZETNM1 )
        BN = PSIN / ZETN
C
    ELSE IF ( NOABS ) THEN
C          ** NO-ABSORPTION CASE
        AN = ( ( RIORIV*RBIGA(N) + ( FN*XINV ) ) * PSIN - PSINM1 )
$          / ( ( RIORIV*RBIGA(N) + ( FN*XINV ) ) * ZETN - ZETNM1 )
        BN = ( ( MRE * RBIGA(N) + ( FN*XINV ) ) * PSIN - PSINM1 )
$          / ( ( MRE * RBIGA(N) + ( FN*XINV ) ) * ZETN - ZETNM1 )
    ELSE
C          ** ABSORPTIVE CASE
        AN = ( ( CIORIV * CBIGA(N) + ( FN*XINV ) ) * PSIN - PSINM1 )
$          / ( ( CIORIV * CBIGA(N) + ( FN*XINV ) ) * ZETN - ZETNM1 )
        BN = ( ( CIOR * CBIGA(N) + ( FN*XINV ) ) * PSIN - PSINM1 )
$          / ( ( CIOR * CBIGA(N) + ( FN*XINV ) ) * ZETN - ZETNM1 )

```

```

      QSCA = QSCA + TWONP1 * ( SQ( AN ) + SQ( BN ) )
C
END IF
C      ** SAVE MIE COEFFICIENTS FOR *PMOM* CALCULN.
LITA( N ) = AN
LITB( N ) = BN
C      ** INCREMENT MIE SUMS FOR NON-ANGLE-
C      ** DEPENDENT QUANTITIES
C
SFORM      = SFORM      + TWONP1 * ( AN + BN )
TFORW( 1 ) = TFORW( 1 ) + TCOEF * ( AN - BN )
SBACK      = SBACK      + ( MM * TWONP1 ) * ( AN - BN )
TBACK( 1 ) = TBACK( 1 ) + ( MM * TCOEF ) * ( AN + BN )
GQSC = GQSC + ( FN - RN ) * REAL( ANM1 * CONJG( AN )
$      + BNM1 * CONJG( BN ) )
$      + COEFF * REAL( AN * CONJG( BN ) )
C
IF ( YESANG ) THEN
C      ** PUT MIE COEFFICIENTS IN FORM
C      ** NEEDED FOR COMPUTING S+, S-
C      ** ( REF. 2, P. 1507 )
ANP = COEFF * ( AN + BN )
BNP = COEFF * ( AN - BN )
C      ** INCREMENT MIE SUMS FOR S+, S-
C      ** WHILE UPWARD RECURSING
C      ** ANGULAR FUNCTIONS LITTLE PI
C      ** AND LITTLE TAU
IF ( ANYANG ) THEN
C      ** ARBITRARY ANGLES
C
C      ** VECTORIZABLE LOOP
DO 80 J = 1, NUMANG
  RTMP = ( XMU( J ) * PIN( J ) ) - PINM1( J )
  TAUN = FN * RTMP - PINM1( J )
  SP( J ) = SP( J ) + ANP * ( PIN( J ) + TAUN )
  SM( J ) = SM( J ) + BNP * ( PIN( J ) - TAUN )
  PINM1( J ) = PIN( J )
  PIN( J ) = ( XMU( J ) * PIN( J ) ) + NP1DN * RTMP
80 CONTINUE
C
ELSE
C      ** ANGLES SYMMETRIC ABOUT 90 DEGREES
C
ANPM = MM * ANP
BNPM = MM * BNP
C      ** VECTORIZABLE LOOP
DO 90 J = 1, NANGD2
  RTMP = ( XMU( J ) * PIN( J ) ) - PINM1( J )
  TAUN = FN * RTMP - PINM1( J )
  SP( J ) = SP( J ) + ANP * ( PIN( J ) + TAUN )
  SMS( J ) = SMS( J ) + BNPM * ( PIN( J ) + TAUN )
  SM( J ) = SM( J ) + BNP * ( PIN( J ) - TAUN )
  SPS( J ) = SPS( J ) + ANPM * ( PIN( J ) - TAUN )
  PINM1( J ) = PIN( J )
  PIN( J ) = ( XMU( J ) * PIN( J ) ) + NP1DN * RTMP
90 CONTINUE
C
END IF

```

```

      END IF
C      ** UPDATE RELEVANT QUANTITIES FOR NEXT
C      ** PASS THROUGH LOOP
      MM = - MM
      ANM1 = AN
      BNM1 = BN
C      ** UPWARD RECURRENCE FOR RICATTI-BESSEL
C      ** FUNCTIONS ( REF. 1, EQ. 17 )
C
      ZET = ( TWONP1 * XINV ) * ZETN - ZETNM1
      ZETNM1 = ZETN
      ZETN = ZET
      PSINM1 = PSIN
      PSIN = REAL( ZETN )
100 CONTINUE
C
C ----- END LOOP TO SUM MIE SERIES -----
C
C
      QEXT = 2. / XX**2 * REAL( SFORW )
      IF ( PERFCT .OR. NOABS ) THEN
        QSCA = QEXT
      ELSE
        QSCA = 2. / XX**2 * QSCA
      END IF
C
      GQSC = 4. / XX**2 * GQSC
      SFORW = 0.5 * SFORW
      SBACK = 0.5 * SBACK
      TFORW( 2 ) = 0.5 * ( SFORW + 0.25 * TFORW( 1 ) )
      TFORW( 1 ) = 0.5 * ( SFORW - 0.25 * TFORW( 1 ) )
      TBACK( 2 ) = 0.5 * ( SBACK + 0.25 * TBACK( 1 ) )
      TBACK( 1 ) = 0.5 * ( - SBACK + 0.25 * TBACK( 1 ) )
C
      IF ( YESANG ) THEN
C      ** RECOVER SCATTERING AMPLITUDES
C      ** FROM S+, S- ( REF. 1, EQ. 11 )
        IF ( ANYANG ) THEN
C      ** VECTORIZABLE LOOP
          DO 110 J = 1, NUMANG
            S1( J ) = 0.5 * ( SP( J ) + SM( J ) )
            S2( J ) = 0.5 * ( SP( J ) - SM( J ) )
110          CONTINUE
C
          ELSE
C      ** VECTORIZABLE LOOP
            DO 120 J = 1, NANGD2
              S1( J ) = 0.5 * ( SP( J ) + SM( J ) )
              S2( J ) = 0.5 * ( SP( J ) - SM( J ) )
120            CONTINUE
C
            ** VECTORIZABLE LOOP
            DO 130 J = 1, NANGD2
              S1( NUMANG+1 - J ) = 0.5 * ( SPS( J ) + SMS( J ) )
              S2( NUMANG+1 - J ) = 0.5 * ( SPS( J ) - SMS( J ) )
130            CONTINUE
          END IF
C

```

```

      END IF
C
C                                     ** CALCULATE LEGENDRE MOMENTS
200 IF ( SELECT(7) )
  $   CALL LPCOEF ( NTRM, NMOM, IPOLZN, MOMDIM, LITA, LITB, PMOM )
C
C
C   IF ( PASS1 ) THEN
C                                     ** COMPARE TEST CASE RESULTS WITH
C                                     ** CORRECT ANSWERS AND ABORT IF BAD
C
      CALL TESTMI ( QEXT, QSCA, QQSC, SFORW, SBACK, S1, S2,
  $   TFORW, TBACK, PMOM, MOMDIM, OK )
      IF ( .NOT. OK ) THEN
        PRNT(1) = .TRUE.
        PRNT(2) = .TRUE.
        CALL MIPRNT( PRNT, XX, PERFCT, CREFIN, NUMANG, XMU, QEXT,
  $   QSCA, QQSC, SELECT(7), NMOM, IPOLZN, MOMDIM,
  $   PMOM, SFORW, SBACK, TFORW, TBACK, S1, S2 )
        CALL ERRMSG( 'MIEVO -- SELF-TEST FAILED', .TRUE. )
      END IF
C
C                                     ** RESTORE USER INPUT VALUES
C
      READ ( SAVINP, 2001, ERR=990 ) XX, CREFIN, MIMCUT, PERFCT,
  $   SELECT(7), ANYANG, NMOM, IPOLZN, NUMANG, XMU(1)
      PASS1 = .FALSE.
      GO TO 10
    END IF
C
      IF ( PRNT(1) .OR. PRNT(2) )
  $   CALL MIPRNT( PRNT, XX, PERFCT, CREFIN, NUMANG, XMU, QEXT,
  $   QSCA, QQSC, SELECT(7), NMOM, IPOLZN, MOMDIM,
  $   PMOM, SFORW, SBACK, TFORW, TBACK, S1, S2 )
C
      RETURN
C
990 CALL ERRMSG( 'MIEVO--I/O ERROR IN SELF-TEST', .TRUE. )
C
2001 FORMAT( 1P, 4E16.8, 3L2, 3I5, E16.8 )
      END
      SUBROUTINE LPCOEF ( NTRM, NMOM, IPOLZN, MOMDIM, A, B, PMOM )
C
C   CALCULATE LEGENDRE POLYNOMIAL EXPANSION COEFFICIENTS (ALSO
C   CALLED MOMENTS) FOR PHASE QUANTITIES ( REF. 5 FORMULATION )
C
C   *** NOTE *** EQS. 2-5 ARE IN ERROR IN DAVE, APPL. OPT. 9,
C   1888 (1970). EQ. 2 REFERS TO M1, NOT M2; EQ. 3 REFERS TO
C   M2, NOT M1. IN EQS. 4 AND 5, THE SUBSCRIPTS ON THE SECOND
C   TERM IN SQUARE BRACKETS SHOULD BE INTERCHANGED.
C
C   *** NOTE *** THIS SUBROUTINE WORKS CORRECTLY IN THE SPECIAL
C   CASE NTRM = 2 (NO. TERMS IN MIE SERIES = 2) BUT CALLS
C   SUBROUTINE *LPCO2T* TO DO THAT CASE BOTH FOR SPEED AND
C   BECAUSE NTRM = 2, BEING THE SMALL-PARTICLE LIMIT, MAY BE
C   CALLED FREQUENTLY.
C
C   *** NOTE *** SOME IMPROVEMENT IN SPEED IS OBTAINABLE IF
C   MOMENTS FOR ALL FOUR PHASE QUANTITIES ARE DESIRED (NELEM=4),

```

C BECAUSE THE SERIES BEING SUMMED FOR THE THIRD PHASE QUANTITY
 C IS THE REAL PART OF A COMPLEX SERIES, AND THE FOURTH PHASE
 C QUANTITY IS THE IMAGINARY PART OF THAT SERIES. HOWEVER, THIS
 C WOULD INVOLVE SUMMING THAT COMPLEX SERIES. THE EXTRA ARITH-
 C METIC FOR SO DOING WILL BE WASTED IN MOST CASES, SINCE MOST
 C USERS ARE NOT INTERESTED IN THE FOURTH PHASE QUANTITY, WHICH
 C IS RELATED TO CIRCULAR POLARIZATION.

C
 C INTEGER IPOLZN, MOMDIM, NMOM, NTRM
 C REAL PMOM(0:MOMDIM, *)
 C COMPLEX A(*), B(*)

C ** SPECIFICATION OF LOCAL VARIABLES

C AM(M) NUMERICAL COEFFICIENTS A-SUB-M-SUPER-L
 C IN DAVE, EQS. 1-15, AS SIMPLIFIED IN REF. 5.
 C
 C BI(I) NUMERICAL COEFFICIENTS B-SUB-I-SUPER-L
 C IN DAVE, EQS. 1-15, AS SIMPLIFIED IN REF. 5.
 C
 C BIDL(I) 1/2 BI(I) TIMES FACTOR CAPITAL-DEL IN DAVE
 C
 C CM,DM() ARRAYS C AND D IN DAVE, EQS. 16-17 (MUELLER FORM),
 C CALCULATED USING RECURRENCE DERIVED IN REF. 5
 C
 C CS,DS() ARRAYS C AND D IN REF. 4, EQS. A5-A6 (SEKERA FORM),
 C CALCULATED USING RECURRENCE DERIVED IN REF. 5
 C
 C C,D() EITHER -CM,DM- OR -CS,DS-, DEPENDING ON -IPOLZN-
 C
 C EVENL TRUE FOR EVEN-NUMBERED MOMENTS; FALSE OTHERWISE
 C
 C IDEL 1 + LITTLE-DEL IN DAVE
 C
 C MAXTRM MAX. NO. OF TERMS IN MIE SERIES
 C
 C MAXMOM MAX. NO. OF NON-ZERO MOMENTS
 C
 C NELEM NO. OF DISTINCT PHASE MATRIX ELEMENTS
 C FOR WHICH TO GET MOMENTS
 C
 C NUMMOM NUMBER OF NON-ZERO MOMENTS
 C
 C RECIP(K) 1 / K
 C
 C PARAMETER (MAXTRM =3002, MAXMOM = 2*MAXTRM, MXMOM2 = MAXMOM/2,
 C \$ MAXRCP = 4*MAXTRM + 2)
 C REAL AM(0:MAXTRM), BI(0:MXMOM2), BIDL(0:MXMOM2),
 C \$ RECIP(MAXRCP)
 C COMPLEX CM(MAXTRM), DM(MAXTRM), CS(MAXTRM), DS(MAXTRM),
 C \$ C(MAXTRM), D(MAXTRM)
 C EQUIVALENCE (C, CM), (D, DM)
 C LOGICAL PASS1, EVENL
 C SAVE PASS1, RECIP
 C DATA PASS1 / .TRUE. /

```

      IF ( PASS1 ) THEN
C
      DO 1 K = 1, MAXRCP
        RECIP( K ) = 1.0 / K
1      CONTINUE
        PASS1 = .FALSE.
C
      END IF
C
      IF ( NTRM.EQ.1 ) THEN
        CALL LPCO1T ( NMOM, IPOLZN, MOMDIM, A, B, PMOM )
        RETURN
      ELSE IF ( NTRM.EQ.2 ) THEN
        CALL LPCO2T ( NMOM, IPOLZN, MOMDIM, A, B, PMOM )
        RETURN
      END IF
C
      IF ( NTRM+2 .GT. MAXTRM )
$      CALL ERRMSG( 'LPCOEF--PARAMETER MAXTRM TOO SMALL', .TRUE. )
C
C      ** CALCULATE MUELLER C, D ARRAYS
      CM( NTRM+2 ) = ( 0., 0. )
      DM( NTRM+2 ) = ( 0., 0. )
      CM( NTRM+1 ) = ( 1. - RECIP( NTRM+1 ) ) * B( NTRM )
      DM( NTRM+1 ) = ( 1. - RECIP( NTRM+1 ) ) * A( NTRM )
      CM( NTRM ) = ( RECIP(NTRM) + RECIP(NTRM+1) ) * A( NTRM )
$      + ( 1. - RECIP(NTRM) ) * B( NTRM-1 )
      DM( NTRM ) = ( RECIP(NTRM) + RECIP(NTRM+1) ) * B( NTRM )
$      + ( 1. - RECIP(NTRM) ) * A( NTRM-1 )
C
      DO 10 K = NTRM-1, 2, -1
        CM( K ) = CM( K+2 ) - ( 1. + RECIP(K+1) ) * B( K+1 )
$      + ( RECIP(K) + RECIP(K+1) ) * A( K )
$      + ( 1. - RECIP(K) ) * B( K-1 )
        DM( K ) = DM( K+2 ) - ( 1. + RECIP(K+1) ) * A( K+1 )
$      + ( RECIP(K) + RECIP(K+1) ) * B( K )
$      + ( 1. - RECIP(K) ) * A( K-1 )
10 CONTINUE
        CM( 1 ) = CM( 3 ) + 1.5 * ( A( 1 ) - B( 2 ) )
        DM( 1 ) = DM( 3 ) + 1.5 * ( B( 1 ) - A( 2 ) )
C
      IF ( IPOLZN.GE.0 ) THEN
C
      DO 20 K = 1, NTRM + 2
        C( K ) = ( 2*K - 1 ) * CM( K )
        D( K ) = ( 2*K - 1 ) * DM( K )
20 CONTINUE
C
      ELSE
C      ** COMPUTE SEKERA C AND D ARRAYS
      CS( NTRM+2 ) = ( 0., 0. )
      DS( NTRM+2 ) = ( 0., 0. )
      CS( NTRM+1 ) = ( 0., 0. )
      DS( NTRM+1 ) = ( 0., 0. )
C
      DO 30 K = NTRM, 1, -1
        CS( K ) = CS( K+2 ) + ( 2*K + 1 ) * ( CM( K+1 ) - B( K ) )

```

```

      DS( K ) = DS( K+2 ) + ( 2*K + 1 ) * ( DM( K+1 ) - A( K ) )
30  CONTINUE
C
      DO 40 K = 1, NTRM + 2
        C( K ) = ( 2*K - 1 ) * CS( K )
        D( K ) = ( 2*K - 1 ) * DS( K )
40  CONTINUE
C
      END IF
C
      NELEM = IABS ( IPOLZN )
      DO 50 J = 1, MAX0( 1, NELEM )
        DO 50 L = 0, NMOM
          PMOM( L,J ) = 0.0
50 CONTINUE
C
      IF( IPOLZN.LT.0 ) NUMMOM = MIN0( NMOM, 2*NTRM - 2 )
      IF( IPOLZN.GE.0 ) NUMMOM = MIN0( NMOM, 2*NTRM )
      IF ( NUMMOM .GT. MAXMOM )
$      CALL ERRMSG( 'LPCOE--PARAMETER MAXTRM TOO SMALL', .TRUE. )
C
C      ** LOOP OVER MOMENTS
      DO 500 L = 0, NUMMOM
        LD2 = L / 2
        EVENL = MOD( L,2 ) .EQ. 0
C
C      ** CALCULATE NUMERICAL COEFFICIENTS
C      ** A-SUB-M AND B-SUB-I IN DAVE
C      ** DOUBLE-SUMS FOR MOMENTS
      IF( L.EQ.0 ) THEN
C
        IDEL = 1
        DO 60 M = 0, NTRM
          AM( M ) = 2.0 * RECIP( 2*M + 1 )
60  CONTINUE
          BI( 0 ) = 1.0
C
        ELSE IF( EVENL ) THEN
C
          IDEL = 1
          DO 70 M = LD2, NTRM
            AM( M ) = ( 1. + RECIP( 2*M-L+1 ) ) * AM( M )
70  CONTINUE
          DO 75 I = 0, LD2-1
            BI( I ) = ( 1. - RECIP( L-2*I ) ) * BI( I )
75  CONTINUE
            BI( LD2 ) = ( 2. - RECIP( L ) ) * BI( LD2-1 )
C
          ELSE
C
            IDEL = 2
            DO 80 M = LD2, NTRM
              AM( M ) = ( 1. - RECIP( 2*M-L+2 ) ) * AM( M )
80  CONTINUE
            DO 85 I = 0, LD2
              BI( I ) = ( 1. - RECIP( L+2*I+1 ) ) * BI( I )
85  CONTINUE
C

```

```

C      END IF
C
C      ** ESTABLISH UPPER LIMITS FOR SUMS
C      ** AND INCORPORATE FACTOR CAPITAL-
C      ** DEL INTO B-SUB-I
C
C      MMAX = NTRM - IDEL
C      IF( IPOLZN.GE.0 ) MMAX = MMAX + 1
C      IMAX = MIN0( LD2, MMAX - LD2 )
C      IF( IMAX.LT.0 ) GO TO 600
C      DO 90 I = 0, IMAX
C          BIDL( I ) = BI( I )
90      CONTINUE
C      IF( EVENL ) BIDL( 0 ) = 0.5 * BIDL( 0 )
C
C      ** PERFORM DOUBLE SUMS JUST FOR
C      ** PHASE QUANTITIES DESIRED BY USER
C
C      IF( NELEM.EQ.0 ) THEN
C
C          DO 110 I = 0, IMAX
C
C              ** VECTORIZABLE LOOP (CRAY)
C
C              SUM = 0.0
C              DO 100 M = LD2, MMAX - I
C                  SUM = SUM + AM( M ) *
C                      ( REAL( C(M-I+1) * CONJG( C(M+I+IDEL) ) )
C                      + REAL( D(M-I+1) * CONJG( D(M+I+IDEL) ) ) )
C              CONTINUE
100          PMOM( L,1 ) = PMOM( L,1 ) + BIDL( I ) * SUM
110          CONTINUE
C          PMOM( L,1 ) = 0.5 * PMOM( L,1 )
C          GO TO 500
C      END IF
C
C      DO 160 I = 0, IMAX
C
C          ** VECTORIZABLE LOOP (CRAY)
C
C          SUM = 0.0
C          DO 150 M = LD2, MMAX - I
C              SUM = SUM + AM( M ) *
C                  REAL( C(M-I+1) * CONJG( C(M+I+IDEL) ) )
C          CONTINUE
150          PMOM( L,1 ) = PMOM( L,1 ) + BIDL( I ) * SUM
160          CONTINUE
C
C      IF( NELEM.GT.1 ) THEN
C
C          DO 210 I = 0, IMAX
C
C              ** VECTORIZABLE LOOP (CRAY)
C
C              SUM = 0.0
C              DO 200 M = LD2, MMAX - I
C                  SUM = SUM + AM( M ) *
C                      REAL( D(M-I+1) * CONJG( D(M+I+IDEL) ) )
C              CONTINUE
200          PMOM( L,2 ) = PMOM( L,2 ) + BIDL( I ) * SUM
210          CONTINUE
C      END IF
C
C      IF( NELEM.GT.2 ) THEN
C
C          DO 310 I = 0, IMAX

```

```

C                                ** VECTORIZABLE LOOP (CRAY)
      SUM = 0.0
      DO 300 M = LD2, MMAX - I
        SUM = SUM + AM( M ) *
$          ( REAL( C(M-I+1) * CONJG( D(M+I+IDEL) ) )
$          + REAL( C(M+I+IDEL) * CONJG( D(M-I+1) ) ) )
300      CONTINUE
      PMOM( L,3 ) = PMOM( L,3 ) + BIDEI( I ) * SUM
310      CONTINUE
      PMOM( L,3 ) = 0.5 * PMOM( L,3 )
      END IF
C
      IF( NELEM.GT.3 ) THEN
C
      DO 410 I = 0, IMAX
C                                ** VECTORIZABLE LOOP (CRAY)
        SUM = 0.0
        DO 400 M = LD2, MMAX - I
          SUM = SUM + AM( M ) *
$            ( AIMAG( C(M-I+1) * CONJG( D(M+I+IDEL) ) )
$            + AIMAG( C(M+I+IDEL) * CONJG( D(M-I+1) ) ) )
400      CONTINUE
        PMOM( L,4 ) = PMOM( L,4 ) + BIDEI( I ) * SUM
410      CONTINUE
        PMOM( L,4 ) = - 0.5 * PMOM( L,4 )
      END IF
C
500 CONTINUE
C
C
600 RETURN
      END
      SUBROUTINE LPCOIT ( NMOM, IPOLZN, MOMDIM, A, B, PMOM )
C
C      CALCULATE LEGENDRE POLYNOMIAL EXPANSION COEFFICIENTS (ALSO
C      CALLED MOMENTS) FOR PHASE QUANTITIES IN SPECIAL CASE WHERE
C      NO. TERMS IN MIE SERIES = 1
C
      INTEGER IPOLZN, MOMDIM, NMOM
      REAL PMOM( 0:MOMDIM, * )
      COMPLEX A( * ), B( * ), CTMP, A1B1C
      SQ( CTMP ) = REAL( CTMP )**2 + AIMAG( CTMP )**2
C
C
      NELEM = IABS ( IPOLZN )
      DO 10 J = 1, MAX0( 1, NELEM )
        DO 10 L = 0, NMOM
          PMOM( L,J ) = 0.0
10 CONTINUE
C
      A1SQ = SQ( A(1) )
      B1SQ = SQ( B(1) )
      A1B1C = A(1) * CONJG( B(1) )
C
      IF( IPOLZN.LT.0 ) THEN
C
      PMOM( L,1 ) = 2.25 * B1SQ

```

```

      IF( NELEM.GT.1 ) PMOM( L,2 ) = 2.25 * A1SQ
      IF( NELEM.GT.2 ) PMOM( L,3 ) = 2.25 * REAL( A1B1C )
      IF( NELEM.GT.3 ) PMOM( L,4 ) = 2.25 * AIMAG( A1B1C )
C
      ELSE
C
      NUMMOM = MINO( NMOM, 2 )
C
      ** LOOP OVER MOMENTS
      DO 100 L = 0, NUMMOM
C
        IF( NELEM.EQ.0 ) THEN
          IF( L.EQ.0 ) PMOM( L,1 ) = 1.5 * ( A1SQ + B1SQ )
          IF( L.EQ.1 ) PMOM( L,1 ) = 1.5 * REAL( A1B1C )
          IF( L.EQ.2 ) PMOM( L,1 ) = 0.1 * PMOM( L,1 )
          GO TO 100
        END IF
C
        IF( L.EQ.0 ) PMOM( L,1 ) = 2.25 * ( A1SQ + B1SQ / 3. )
        IF( L.EQ.1 ) PMOM( L,1 ) = 1.5 * REAL( A1B1C )
        IF( L.EQ.2 ) PMOM( L,1 ) = 0.3 * B1SQ
C
        IF( NELEM.GT.1 ) THEN
          IF( L.EQ.0 ) PMOM( L,2 ) = 2.25 * ( B1SQ + A1SQ / 3. )
          IF( L.EQ.1 ) PMOM( L,2 ) = PMOM( 1,1 )
          IF( L.EQ.2 ) PMOM( L,2 ) = 0.3 * A1SQ
        END IF
C
        IF( NELEM.GT.2 ) THEN
          IF( L.EQ.0 ) PMOM( L,3 ) = 3.0 * REAL( A1B1C )
          IF( L.EQ.1 ) PMOM( L,3 ) = 0.75 * ( A1SQ + B1SQ )
          IF( L.EQ.2 ) PMOM( L,3 ) = 0.1 * PMOM( L,3 )
        END IF
C
        IF( NELEM.GT.3 ) THEN
          IF( L.EQ.0 ) PMOM( L,4 ) = - 1.5 * AIMAG( A1B1C )
          IF( L.EQ.1 ) PMOM( L,4 ) = 0.0
          IF( L.EQ.2 ) PMOM( L,4 ) = - 0.2 * PMOM( L,4 )
        END IF
C
      100 CONTINUE
C
      END IF
C
      RETURN
      END
      SUBROUTINE LP2T2T ( NMOM, IPOLZN, MOMDIM, A, B, PMOM )
C
      CALCULATE LEGENDRE POLYNOMIAL EXPANSION COEFFICIENTS (ALSO
      CALLED MOMENTS) FOR PHASE QUANTITIES IN SPECIAL CASE WHERE
      NO. TERMS IN MIE SERIES = 2
C
      INTEGER IPOLZN, MOMDIM, NMOM
      REAL PMOM( 0:MOMDIM, * )
      COMPLEX A( * ), B( * )
      COMPLEX A2C, B2C, CTMP, CA, CAC, CAT, CB, CBC, CBT, CG, CH,
      S
      S2S1C( 0:4 ), T1T2C( 0:2 )
      SQ( CTMP ) = REAL( CTMP )**2 + AIMAG( CTMP )**2

```

```

C
C
NELEM = IABS ( IPOLZN )
DO 10 J = 1, MAX0( 1, NELEM )
  DO 10 L = 0, NMOM
    PMOM( L,J ) = 0.0
10 CONTINUE
C
CA = 3. * A(1) - 5. * B(2)
CAT= 3. * B(1) - 5. * A(2)
CAC = CONJG( CA )
A2SQ = SQ( A(2) )
B2SQ = SQ( B(2) )
A2C = CONJG( A(2) )
B2C = CONJG( B(2) )
C
IF( IPOLZN.LT.0 ) THEN
C
C      ** LOOP OVER SEKERA MOMENTS
NUMMOM = MIN0( NMOM, 2 )
DO 50 L = 0, NUMMOM
C
  IF( L.EQ.0 ) THEN
    PMOM( L,1 ) = 0.25 * ( SQ(CAT) + (100./3.) * B2SQ )
  ELSE IF( L.EQ.1 ) THEN
    PMOM( L,1 ) = (5./3.) * REAL( CAT * B2C )
  ELSE
    PMOM( L,1 ) = (10./3.) * B2SQ
  END IF
C
  IF( NELEM.GT.1 ) THEN
    IF( L.EQ.0 ) THEN
      PMOM( L,2 ) = 0.25 * ( SQ(CA) + (100./3.) * A2SQ )
    ELSE IF( L.EQ.1 ) THEN
      PMOM( L,2 ) = (5./3.) * REAL( CA * A2C )
    ELSE
      PMOM( L,2 ) = (10./3.) * A2SQ
    END IF
  END IF
C
  IF( NELEM.GT.2 ) THEN
    IF( L.EQ.0 ) THEN
      T1T2C(L) = 0.25 * ( CAT*CAC + (100./3.)*B(2)*A2C )
      PMOM( L,3 ) = REAL( T1T2C(L) )
    ELSE IF( L.EQ.1 ) THEN
      T1T2C(L) = (5./6.) * ( B(2)*CAC + CAT*A2C )
      PMOM( L,3 ) = REAL( T1T2C(L) )
    ELSE
      T1T2C(L) = (10./3.) * B(2) * A2C
      PMOM( L,3 ) = REAL( T1T2C(L) )
    END IF
  END IF
C
  IF( NELEM.GT.3 ) PMOM( L,4 ) = - AIMAG( T1T2C(L) )
C
50 CONTINUE
C
ELSE

```

```

C      CB = 3. * B(1) + 5. * A(2)
      CBT = 3. * A(1) + 5. * B(2)
      CBC = CONJG( CB )
      CG = (1./3.) * ( CBC*CBT + 10.*( CAC*A(2) + B2C*CAT ) )
      CH = 2.*( CBC*A(2) + B2C*CBT )

C
C      ** LOOP OVER MUELLER MOMENTS
      NUMMOM = MIN0( NMOM, 4 )
      DO 100 L = 0, NUMMOM
        IF( L.EQ.0 ) THEN
          PMOM( L,1 ) = 0.25 * SQ(CA) + (1./12.) * SQ(CB)
          + (5./3.) * REAL(CA*B2C) + 5.*B2SQ
        ELSE IF( L.EQ.1 ) THEN
          PMOM( L,1 ) = REAL( CB * ( (1./6.)*CAC + B2C ) )
        ELSE IF( L.EQ.2 ) THEN
          PMOM( L,1 ) = (1./30.) * SQ(CB) + (20./7.) * B2SQ
          + (2./3.) * REAL( CA * B2C )
        ELSE IF( L.EQ.3 ) THEN
          PMOM( L,1 ) = (2./7.) * REAL( CB * B2C )
        ELSE
          PMOM( L,1 ) = (40./63.) * B2SQ
        END IF
      END IF

C
      IF( NELEM.NE.1 ) THEN
        IF( L.EQ.0 ) THEN
          PMOM( L,2 ) = 0.25*SQ(CAT) + (1./12.) * SQ(CBT)
          + (5./3.) * REAL(CAT*A2C) + 5.*A2SQ
        ELSE IF( L.EQ.1 ) THEN
          PMOM( L,2 ) = REAL( CBT * ((1./6.)*CONJG(CAT) + A2C) )
        ELSE IF( L.EQ.2 ) THEN
          PMOM( L,2 ) = (1./30.) * SQ(CBT) + (20./7.) * A2SQ
          + (2./3.) * REAL( CAT * A2C )
        ELSE IF( L.EQ.3 ) THEN
          PMOM( L,2 ) = (2./7.) * REAL( CBT * A2C )
        ELSE
          PMOM( L,2 ) = (40./63.) * A2SQ
        END IF
      END IF

C
      IF( NELEM.EQ.0 ) THEN
        PMOM( L,1 ) = 0.5 * ( PMOM(L,1) + PMOM(L,2) )
        GO TO 100
      END IF

C
      IF( NELEM.GT.2 ) THEN
        IF( L.EQ.0 ) THEN
          S2S1C(L) = 0.25 * ( CAC*CAT + CG + 20.*B2C*A(2) )
          PMOM( L,3 ) = REAL( S2S1C(L) )
        ELSE IF( L.EQ.1 ) THEN
          S2S1C(L) = (1./12.) * ( CAC*CBT + CBC*CAT + 3.*CH )
          PMOM( L,3 ) = REAL( S2S1C(L) )
        ELSE IF( L.EQ.2 ) THEN
          S2S1C(L) = 0.1 * ( CG + (200./7.) * B2C * A(2) )
          PMOM( L,3 ) = REAL( S2S1C(L) )
        ELSE IF( L.EQ.3 ) THEN
          S2S1C(L) = CH / 14.
        END IF
      END IF

```

```

      PMOM( L,3 ) = REAL( S2S1C(L) )
    ELSE
      S2S1C(L) = (40./63.) * B2C * A(2)
      PMOM( L,3 ) = REAL( S2S1C(L) )
    END IF
  END IF
C
      IF( NELEM.GT.3 ) PMOM( L,4 ) = AIMAG( S2S1C(L) )
C
100  CONTINUE
C
      END IF
C
      RETURN
    END
  COMPLEX FUNCTION CONFRA( N, ZINV, XX )
C
C      COMPUTE BESSEL FUNCTION RATIO CAPITAL-A-SUB-N FROM ITS
C      CONTINUED FRACTION USING LENTZ METHOD ( REF. 1, PP. 17-20 )
C
C      ZINV = RECIPROCAL OF ARGUMENT OF CAPITAL-A
C
C      I N T E R N A L   V A R I A B L E S
C      -----
C
C      CAK      TERM IN CONTINUED FRACTION EXPANSION OF CAPITAL-A
C               ( REF. 1, EQ. 25 )
C      CAPT     FACTOR USED IN LENTZ ITERATION FOR CAPITAL-A
C               ( REF. 1, EQ. 27 )
C      CDENOM   DENOMINATOR IN -CAPT- ( REF. 1, EQ. 28B )
C      CNUMBER  NUMERATOR IN -CAPT- ( REF. 1, EQ. 28A )
C      CDTD     PRODUCT OF TWO SUCCESSIVE DENOMINATORS OF -CAPT-
C               FACTORS ( REF. 1, EQ. 34C )
C      CNTN     PRODUCT OF TWO SUCCESSIVE NUMERATORS OF -CAPT-
C               FACTORS ( REF. 1, EQ. 34B )
C      EPS1     ILL-CONDITIONING CRITERION
C      EPS2     CONVERGENCE CRITERION
C      KK       SUBSCRIPT K OF -CAK- ( REF. 1, EQ. 25B )
C      KOUNT    ITERATION COUNTER ( USED ONLY TO PREVENT RUNAWAY )
C      MAXIT    MAX. ALLOWED NO. OF ITERATIONS
C      MM       + 1 AND - 1, ALTERNATELY
C
C      INTEGER  N
C      REAL     XX
C      COMPLEX  ZINV
C      COMPLEX  CAK, CAPT, CDENOM, CDTD, CNUMBER, CNTN
C      DATA  EPS1 / 1.E - 2 /, EPS2 / 1.E - 8 /
C      DATA  MAXIT / 10000 /
C
C
C      *** REF. 1, EQS. 25A, 27
C
      CONFRA = ( N + 1 ) * ZINV
      MM      = - 1
      KK      = 2 * N + 3
      CAK     = ( MM * KK ) * ZINV
      CDENOM  = CAK
      CNUMBER = CDENOM + 1.0 / CONFRA
      KOUNT   = 1

```

```

C
20 KOUNT = KOUNT + 1
  IF ( KOUNT.GT.MAXIT )
    $ CALL ERRMSG( 'CONFRA--ITERATION FAILED TO CONVERGES', .TRUE.)
C
C                                     *** REF. 2, EQ. 25B
C
MM = - MM
KK = KK + 2
CAK = ( MM * KK ) * ZINV
C                                     *** REF. 2, EQ. 32
  IF ( CABS( CNUMER/CAK ).LE.EPS1
    $ .OR. CABS( CDENOM/CAK ).LE.EPS1 ) THEN
C
C                                     ** ILL-CONDITIONED CASE -- STRIDE
C                                     ** TWO TERMS INSTEAD OF ONE
C
C                                     *** REF. 2, EQS. 34
CNTN = CAK * CNUMER + 1.0
CTD = CAK * CDENOM + 1.0
CONFRA = ( CNTN / CTD ) * CONFRA
C                                     *** REF. 2, EQ. 25B
MM = - MM
KK = KK + 2
CAK = ( MM * KK ) * ZINV
C                                     *** REF. 2, EQS. 35
CNUMER = CAK + CNUMER / CNTN
CDENOM = CAK + CDENOM / CTD
KOUNT = KOUNT + 1
GO TO 20
C
ELSE
C                                     ** WELL-CONDITIONED CASE
C
C                                     *** REF. 2, EQS. 26, 27
CAPT = CNUMER / CDENOM
CONFRA = CAPT * CONFRA
C                                     ** CHECK FOR CONVERGENCE
C                                     ** ( REF. 2, EQ. 31 )
C
  IF ( ABS( REAL( CAPT ) - 1.0 ).GE.EPS2
    $ .OR. ABS( AIMAG(CAPT) ) .GE.EPS2 ) THEN
C
C                                     *** REF. 2, EQS. 30A-B
CNUMER = CAK + 1.0 / CNUMER
CDENOM = CAK + 1.0 / CDENOM
GO TO 20
  END IF
END IF
C
RETURN
C
END
SUBROUTINE MIPRNT( PRNT, XX, PERFCT, CREFIN, NUMANG, XMU,
$ QEXT, QSCA, QQSC, SELECT, NMOM, IPOLZN,
$ MOMDIM, PMOM, SFORW, SBACK, TFORW,
$ TBACK, S1, S2 )
C

```

```

C      PRINT SCATTERING QUANTITIES OF A SINGLE PARTICLE
C
C      LOGICAL PERFCT, PRNT( * ), SELECT
C      INTEGER IPOLZN, MOMDIM, NMOM, NUMANG
C      REAL QQSC, PMOM( 0:MOMDIM, * ), QEXT, QSCA, XX, XMU( * )
C      COMPLEX CREFIN, SFORW, SBACK, TFORW( * ), TBACK( * ),
C      $      S1( * ), S2( * )
C
C
C      IF ( PERFCT )      WRITE ( *,1001 ) XX
C      IF ( .NOT.PERFCT ) WRITE ( *,1002 ) CREFIN, XX
C
C      IF ( PRNT(1) .AND. NUMANG.GT.0 ) THEN
C
C      WRITE ( *,1010 )
C      DO 10 I = 1, NUMANG
C          FI1 = REAL( S1(I) ) **2 + AIMAG( S1(I) ) **2
C          FI2 = REAL( S2(I) ) **2 + AIMAG( S2(I) ) **2
C          WRITE( *,1011 ) I, XMU(I), S1(I), S2(I), S1(I)*CONJG(S2(I)),
C      $          FI1, FI2, 0.5*(FI1+FI2), (FI2-FI1)/(FI2+FI1)
C          WRITE(10,1111) ACOS(XMU(I))*57.29578, FI1
C          WRITE(11,1111) ACOS(XMU(I))*57.29578, FI2
1111      FORMAT(2G16.8)
10      CONTINUE
C
C      .END IF
C
C
C      IF ( PRNT(2) ) THEN
C
C      WRITE ( *,1020 ) 0.0, SFORW, TFORW(1), TFORW(2),
C      $      180., SBACK, TBACK(1), TBACK(2)
C      WRITE ( *,1022 ) QEXT, QSCA, QEXT-QSCA, QEXT-QQSC
C
C      IF ( SELECT ) THEN
C          IF ( IPOLZN.EQ.0 ) THEN
C              NPQUAN = 1
C              WRITE ( *,1024 ) ' PHASE FCN '
C          ELSE IF ( IPOLZN.GT.0 ) THEN
C              NPQUAN = IPOLZN
C              WRITE ( *,1024 ) '      M1      ', M2      ',
C      $              '      S21      ', D21      '
C          ELSE
C              NPQUAN = - IPOLZN
C              WRITE ( *,1024 ) '      R1      ', R2      ',
C      $              '      R3      ', R4      '
C          END IF
C
C      FNORM = 4. / ( XX**2 * QSCA )
C      DO 20 M = 0, NMOM
C          WRITE ( *,1026 ) M, ( FNORM*PMOM(M,NP), NP = 1, NPQUAN )
20      CONTINUE
C      END IF
C
C      END IF
C
C      RETURN

```

```

C
C 1001 FORMAT( //, ' PERFECTLY CONDUCTING CASE, SIZE PARAMETER =',
C $      1P, E11.4 )
C 1002 FORMAT( //, ' REFRACTIVE INDEX:  REAL ', 1P, E11.4, '  IMAG ', E11.4,
C $      ', SIZE PARAMETER =', E11.4 )
C 1010 FORMAT( /, '      COS(ANGLE)  ----- S1 ----- S2 ',
C $      '----- S1*CONJG(S2) --- I1=S1**2',
C $      ' I2=S2**2 (I1+I2)/2  DEG POLZN' )
C 1011 FORMAT( I4, F10.6, 1P, 10E11.3 )
1020 FORMAT( /, '  ANGLE', 9X, 'S-SUB-1', 17X, 'T-SUB-1', 17X, 'T-SUB-2',
$      /, (OP, F7.2, 1P, 6E12.3) )
1022 FORMAT( /, ' EFFICIENCY FACTORS,  EXTINCTION:', 1P, E11.4,
$      '  SCATTERING:', E11.4, '  ABSORPTION:', E11.4,
$      '  RAD. PRESSURE:', E11.4 )
1024 FORMAT( /, ' NORMALIZED MOMENTS OF :', 4A )
1026 FORMAT( '      MOMENT NO.', I4, 4X, 1P, 4E13.4 )
END
SUBROUTINE SMALL1 ( XX, NUMANG, XMU, QEXT, QSCA, GQSC, SFORW,
$      SBACK, S1, S2, TFORW, TBACK, A, B )
C
C      SMALL-PARTICLE LIMIT OF MIE QUANTITIES IN TOTALLY REFLECTING
C      LIMIT ( MIE SERIES TRUNCATED AFTER 2 TERMS )
C
C      A,B      FIRST TWO MIE COEFFICIENTS, WITH NUMERATOR AND
C      DENOMINATOR EXPANDED IN POWERS OF -XX- ( A FACTOR
C      OF XX**3 IS MISSING BUT IS RESTORED BEFORE RETURN
C      TO CALLING PROGRAM ) ( REF. 2, P. 1508 )
C
C      INTEGER  NUMANG
C      REAL     GQSC, QEXT, QSCA, XX, XMU( * )
C      COMPLEX  A( 2 ), B( 2 ), SFORW, SBACK, S1( * ), S2( * ),
C $      TFORW( * ), TBACK( * )
C
C      PARAMETER ( TWOTHR = 2./3., FIVTHR = 5./3., FIVNIN = 5./9. )
C      COMPLEX  CTMP
C      SQ( CTMP ) = REAL( CTMP )**2 + AIMAG( CTMP )**2
C
C      A( 1 ) = CMPLX ( 0., TWOTHR * ( 1. - 0.2 * XX**2 ) )
C $      / CMPLX ( 1. - 0.5 * XX**2, TWOTHR * XX**3 )
C
C      B( 1 ) = CMPLX ( 0., - ( 1. - 0.1 * XX**2 ) / 3. )
C $      / CMPLX ( 1. + 0.5 * XX**2, - XX**3 / 3. )
C
C      A( 2 ) = CMPLX ( 0., XX**2 / 30. )
C      B( 2 ) = CMPLX ( 0., - XX**2 / 45. )
C
C      QSCA = 6. * XX**4 * ( SQ( A(1) ) + SQ( B(1) )
C $      + FIVTHR * ( SQ( A(2) ) + SQ( B(2) ) ) )
C      QEXT = QSCA
C      GQSC = 6. * XX**4 * REAL( A(1) * CONJG( A(2) + B(1) )
C $      + ( B(1) + FIVNIN * A(2) ) * CONJG( B(2) ) ) )
C
C      RTMP = 1.5 * XX**3
C      SFORW = RTMP * ( A(1) + B(1) + FIVTHR * ( A(2) + B(2) ) )
C      SBACK = RTMP * ( A(1) - B(1) - FIVTHR * ( A(2) - B(2) ) )
C      TFORW( 1 ) = RTMP * ( B(1) + FIVTHR * ( 2.*B(2) - A(2) ) )

```

```

TFORW( 2 ) = RTMP * ( A(1) + FIVTHR * ( 2.*A(2) - B(2) ) )
TBACK( 1 ) = RTMP * ( B(1) - FIVTHR * ( 2.*B(2) + A(2) ) )
TBACK( 2 ) = RTMP * ( A(1) - FIVTHR * ( 2.*A(2) + B(2) ) )
C
DO 10 J = 1, NUMANG
  S1( J ) = RTMP * ( A(1) + B(1) * XMU(J) + FIVTHR *
$      ( A(2) * XMU(J) + B(2) * ( 2.*XMU(J)**2 - 1. ) ) )
  S2( J ) = RTMP * ( B(1) + A(1) * XMU(J) + FIVTHR *
$      ( B(2) * XMU(J) + A(2) * ( 2.*XMU(J)**2 - 1. ) ) )
10 CONTINUE
C
C      ** RECOVER ACTUAL MIE COEFFICIENTS
A( 1 ) = XX**3 * A( 1 )
A( 2 ) = XX**3 * A( 2 )
B( 1 ) = XX**3 * B( 1 )
B( 2 ) = XX**3 * B( 2 )
C
RETURN
END
SUBROUTINE SMALL2 ( XX, CIOR, CALCQE, NUMANG, XMU, QEXT, QSCA,
$      QGSC, SFORW, SBACK, S1, S2, TFORW, TBACK,
$      A, B )
C
C      SMALL-PARTICLE LIMIT OF MIE QUANTITIES FOR GENERAL REFRACTIVE
C      INDEX ( MIE SERIES TRUNCATED AFTER 2 TERMS )
C
C      A,B      FIRST TWO MIE COEFFICIENTS, WITH NUMERATOR AND
C      DENOMINATOR EXPANDED IN POWERS OF -XX- ( A FACTOR
C      OF XX**3 IS MISSING BUT IS RESTORED BEFORE RETURN
C      TO CALLING PROGRAM ) ( REF. 2, P. 1508 )
C
C      CIORSQ   SQUARE OF REFRACTIVE INDEX
C
C      LOGICAL  CALCQE
C      INTEGER  NUMANG
C      REAL     QGSC, QEXT, QSCA, XX, XMU( * )
C      COMPLEX  A( 2 ), B( 2 ), CIOR, SFORW, SBACK, S1( * ), S2( * ),
$      TFORW( * ), TBACK( * )
C
C      PARAMETER ( TWOTHR = 2./3., FIVTHR = 5./3. )
C      COMPLEX  CTMP, CIORSQ
C      SQ( CTMP ) = REAL( CTMP )**2 + AIMAG( CTMP )**2
C
C      CIORSQ = CIOR**2
C      CTMP = CMPLX( 0., TWOTHR ) * ( CIORSQ - 1.0 )
C      A(1) = CTMP * ( 1.0 - 0.1 * XX**2 + (CIORSQ/350. + 1./280.)*XX**4 )
$      / ( CIORSQ + 2.0 + ( 1.0 - 0.7 * CIORSQ ) * XX**2
$      - ( CIORSQ**2/175. - 0.275 * CIORSQ + 0.25 ) * XX**4
$      + XX**3 * CTMP * ( 1.0 - 0.1 * XX**2 ) )
C
C      B(1) = (XX**2/30.) * CTMP * ( 1.0 + (CIORSQ/35. - 1./14.) *XX**2 )
$      / ( 1.0 - ( CIORSQ/15. - 1./6. ) * XX**2 )
C
C      A(2) = ( 0.1 * XX**2 ) * CTMP * ( 1.0 - XX**2 / 14. )
$      / ( 2. * CIORSQ + 3. - ( CIORSQ/7. - 0.5 ) * XX**2 )
C
C      QSCA = 6. * XX**4 * ( SQ(A(1)) + SQ(B(1)) + FIVTHR * SQ(A(2)) )

```

```

      GQSC = 6. * XX**4 * REAL( A(1) * CONJG( A(2) + B(1) ) )
      QEXT = QSCA
      IF ( CALCQE ) QEXT = 6. * XX * REAL( A(1) + B(1) + FIVTHR * A(2) )

C
      RTMP = 1.5 * XX**3
      SFORW      = RTMP * ( A(1) + B(1) + FIVTHR * A(2) )
      SBACK      = RTMP * ( A(1) - B(1) - FIVTHR * A(2) )
      TFORW( 1 ) = RTMP * ( B(1) - FIVTHR * A(2) )
      TFORW( 2 ) = RTMP * ( A(1) + 2. * FIVTHR * A(2) )
      TBACK( 1 ) = TFORW( 1 )
      TBACK( 2 ) = RTMP * ( A(1) - 2. * FIVTHR * A(2) )

C
      DO 10 J = 1, NUMANG
        S1( J ) = RTMP * ( A(1) + ( B(1) + FIVTHR * A(2) ) * XMU(J) )
        S2( J ) = RTMP * ( B(1) + A(1) * XMU(J) + FIVTHR * A(2)
$              * ( 2. * XMU(J)**2 - 1. ) )
10 CONTINUE

C
C              ** RECOVER ACTUAL MIE COEFFICIENTS
      A( 1 ) = XX**3 * A( 1 )
      A( 2 ) = XX**3 * A( 2 )
      B( 1 ) = XX**3 * B( 1 )
      B( 2 ) = ( 0., 0. )

C
      RETURN
      END
      SUBROUTINE TESTMI ( QEXT, QSCA, GQSC, SFORW, SBACK, S1, S2,
$      TFORW, TBACK, PMOM, MOMDIM, OK )

C
C      COMPARE MIE CODE TEST CASE RESULTS WITH CORRECT ANSWERS
C      AND RETURN OK=FALSE IF EVEN ONE RESULT IS INACCURATE.
C
C      THE TEST CASE IS : MIE SIZE PARAMETER = 10
C                        REFRACTIVE INDEX   = 1.5 - 0.1 I
C                        SCATTERING ANGLE = 140 DEGREES
C                        1 SEKERA MOMENT
C      RESULTS FOR THIS CASE MAY BE FOUND AMONG THE TEST CASES
C      AT THE END OF REFERENCE (1).
C
C      *** NOTE *** WHEN RUNNING ON SOME COMPUTERS, ESP. IN SINGLE
C      PRECISION, THE 'ACCUR' CRITERION BELOW MAY HAVE TO BE RELAXED.
C      HOWEVER, IF 'ACCUR' MUST BE SET LARGER THAN 10**-4 IN ORDER
C      TO PASS THESE TESTS, YOUR COMPUTER IS PROBABLY NOT ACCURATE
C      ENOUGH TO DO NON-TRIVIAL MIE COMPUTATIONS.
C
      REAL      QEXT, QSCA, GQSC, PMOM( 0:MOMDIM, * )
      COMPLEX   SFORW, SBACK, S1( * ), S2( * ), TFORW( * ), TBACK( * )
      LOGICAL   OK, WRONG

C
      PARAMETER ( MAXMOM = 1 )
      REAL      ACCUR, TESTQE, TESTQS, TESTGQ, TESTPM( 0:MAXMOM )
      COMPLEX   TESTSF, TESTSB, TESTS1, TESTS2, TESTTF( 2 ), TESTTB( 2 )
      DATA     TESTQE / 2.459791 /, TESTQS / 1.235144 /,
$      TESTGQ / 1.139235 /, TESTSF / ( 61.49476, -3.177994 ) /,
$      TESTSB / ( 1.493434, 0.2963657 ) /,
$      TESTS1 / ( -0.1548380, -1.128972 ) /,
$      TESTS2 / ( 0.05669755, 0.5425681 ) /,
$      TESTTF / ( 12.95238, -136.6436 ), ( 48.54238, 133.4656 ) /,

```

```

$      TESTTB / ( 41.88414, -15.57833 ), ( 43.37758, -15.28196 )/,
$      TESTPM / 227.1975, 183.6898 /
DATA   ACCUR / 1.E-4 /
C      DATA ACCUR / 1.E-6 /
      WRONG( CALC, EXACT ) = ABS( (CALC - EXACT) / EXACT ) .GT. ACCUR
C
C
      OK = .TRUE.
      IF ( WRONG( QEXT,TESTQE ) ) CALL TSTBAD( 'QEXT', -99, OK )
      IF ( WRONG( QSCA,TESTQS ) ) CALL TSTBAD( 'QSCA', -99, OK )
      IF ( WRONG( GQSC,TESTGQ ) ) CALL TSTBAD( 'GQSC', -99, OK )
C
      IF ( WRONG( REAL(SFORW), REAL(TESTSF) ) .OR.
$      WRONG( AIMAG(SFORW), AIMAG(TESTSF) ) )
$      CALL TSTBAD( 'SFORW', -99, OK )
C
      IF ( WRONG( REAL(SBACK), REAL(TESTSB) ) .OR.
$      WRONG( AIMAG(SBACK), AIMAG(TESTSB) ) )
$      CALL TSTBAD( 'SBACK', -99, OK )
C
      IF ( WRONG( REAL(S1(1)), REAL(TESTS1) ) .OR.
$      WRONG( AIMAG(S1(1)), AIMAG(TESTS1) ) )
$      CALL TSTBAD( 'S1', 1, OK )
C
      IF ( WRONG( REAL(S2(1)), REAL(TESTS2) ) .OR.
$      WRONG( AIMAG(S2(1)), AIMAG(TESTS2) ) )
$      CALL TSTBAD( 'S2', 1, OK )
C
      DO 20 N = 1, 2
        IF ( WRONG( REAL(TFORW(N)), REAL(TESTTF(N)) ) .OR.
$        WRONG( AIMAG(TFORW(N)), AIMAG(TESTTF(N)) ) )
$        CALL TSTBAD( 'TFORW', N, OK )
        IF ( WRONG( REAL(TBACK(N)), REAL(TESTTB(N)) ) .OR.
$        WRONG( AIMAG(TBACK(N)), AIMAG(TESTTB(N)) ) )
$        CALL TSTBAD( 'TBACK', N, OK )
20 CONTINUE

      DO 30 M = 0, MAXMOM
        IF ( WRONG( PMOM(M,1), TESTPM(M) ) )
$        CALL TSTBAD( 'PMOM', M, OK )
30 CONTINUE

      RETURN
C
      END
      COMPLEX FUNCTION CSUM ( N, CX, ISKIP )

C      FORTRAN EQUIVALENT OF BLAS SUMMING ROUTINE FOR COMPLEX NUMBERS
C      WITH A SKIP DISTANCE OF UNITY

      COMPLEX CX( * )

      CSUM = ( 0., 0. )
      DO 1 I = 1, N
        CSUM = CSUM + CX( I )
1     CONTINUE
      RETURN

```

```

END
SUBROUTINE ERRMSG( MESSAG, FATAL )
C
C     PRINT OUT AN ERROR MESSAGE AND ABORT IF ERROR IS FATAL
C
C     LOGICAL      FATAL
C     CHARACTER*(*) MESSAG
C     INTEGER      MAXMSG, NUMMSG
C     SAVE         MAXMSG, NUMMSG
C     DATA NUMMSG / 0 /, MAXMSG / 50 /
C
C     NUMMSG = NUMMSG + 1
C     IF ( NUMMSG.GT.MAXMSG ) THEN
C         WRITE ( *,99 )
C         STOP
C     END IF
C
C     IF ( FATAL ) THEN
C         WRITE ( *,100 ) MESSAG
C         STOP
C     ELSE
C         WRITE ( *,101 ) MESSAG
C         RETURN
C     ENDIF
C
C     99 FORMAT( ///,' >>>>>> TOO MANY ERRORS -- ABORTING <<<<<< ',/// )
C     100 FORMAT( /,' ***** ERROR >>>>>> ', A )
C     101 FORMAT( /,' ***** WARNING >>>>>> ', A )
C     END
C     SUBROUTINE WRTBAD ( VARNAM, ERFLAG )
C
C         WRITE NAMES OF ERRONEOUS VARIABLES
C
C     INPUT :   VARNAM = NAME OF ERRONEOUS VARIABLE TO BE WRITTEN
C               ( CHARACTER, LENGTH 8 )
C
C     OUTPUT :  ERFLAG = LOGICAL FLAG, SET TRUE BY THIS ROUTINE
C     -----
C     CHARACTER*(*) VARNAM
C     LOGICAL      ERFLAG
C     INTEGER      MAXMSG, NUMMSG
C     SAVE NUMMSG, MAXMSG
C     DATA NUMMSG / 0 /, MAXMSG / 50 /
C
C     NUMMSG = NUMMSG + 1
C     WRITE ( *, 10 ) VARNAM
C     ERFLAG = .TRUE.
C     IF ( NUMMSG.EQ.MAXMSG )
C $     CALL ERRMSG ( 'TOO MANY INPUT ERRORS. ABORTING...$', .TRUE. )
C     RETURN
C
C     10 FORMAT( ' **** INPUT VARIABLE ', A, ' IN ERROR **** ' )
C     END
C     SUBROUTINE TSTBAD( VARNAM, INDX, OK )

```

```

C
C  WRITE NAME (-VARNAM-) AND INDEX (-INDX-) OF VARIABLE FAILING
C  SELF-TEST ( USE  INDX = - 99  TO INDICATE A SCALAR -VARNAM- )
C
C  CHARACTER*(*)  VARNAM
C  LOGICAL        OK
C  INTEGER        INDX
C
C      OK = .FALSE.
C      IF( INDX.EQ.-99 ) THEN
C          WRITE( *, 101 )  VARNAM
C      ELSE
C          WRITE( *, 101 )  VARNAM, INDX
C      ENDIF
C
C      RETURN
C
101 FORMAT( /, ' SELF-TEST FAILED FOR OUTPUT VARIABLE ', A,
$          :, ' WITH INDEX', I4 )
END

```

PROGRAM SLICE

C THIS PROGRAM WILL DRAW IN COLOR, SLICES OF THE PRESSURE PROFILE
C IN THE CELL.

```

      IMPLICIT REAL(A-Z)

      PARAMETER (NUM_COLORS=255)

      INTEGER I
      CHARACTER*80 FILENAME
      CHARACTER*1 CH
      LOGICAL NORM
      REAL PK_PK(1000),PK_PK_2(1000),X(1000),Y(1000)

      COMMON VD_ID,WD_ID,VCM_ID

      TYPE *
      TYPE *, 'SLICE V1.2'
      TYPE *

100    TYPE 1
1      FORMAT(' ENTER X DATA FILE NAME -> '$)
      ACCEPT 2,FILENAME
2      FORMAT(A80)

      OPEN (UNIT=1, NAME=FILENAME, STATUS='OLD', READONLY)

      TYPE 3
3      FORMAT(' ENTER Y DATA FILE NAME -> '$)
      ACCEPT 2,FILENAME

      OPEN (UNIT=2, NAME=FILENAME, STATUS='OLD', READONLY)

      TYPE 30
30     FORMAT(' NORMALIZE TO 1.0 (Y/N)? '$)
      ACCEPT ' (A1) ',CH
      NORM=CH.EQ. 'Y'.OR.CH.EQ. 'Y'

      READ(1,4) ORIG_X,ORIG_Y,ORIG_Z,ORIG_Y1,ORIG_Z1,ORIG_PK_PK,
2          ORIG_FREQ,ORIG_TEMP1,ORIG_TEMP2,ORIG_VOLT
4      FORMAT(F4.1,2X,F4.1,2X,F5.1,2X,F5.3,2X,F5.3,2X,F5.2,2X,
2          F5.2,2X,F4.1,2X,F4.1,2X,F4.2)

      I=1
      DO WHILE(.TRUE.)
          I=I+1
          READ(1,4,END=5) X(I),TY,TZ,TY1,TZ1,PK_PK(I)
          X(I)=IFIX(X(I)+0.5)
          IF (I.EQ.1) Z_SLICE=TZ
          IF (NORM) THEN
              PK_PK(I)=PK_PK(I)/ORIG_PK_PK
          ELSE
              PK_PK_MAX=MAX(PK_PK_MAX,PK_PK(I))
          ENDIF
      END DO

5      X(1)=I-2

```

```

        PK_PK(1)=I-2

        CLOSE(UNIT=1)

        READ(2,4) ORIG_X,ORIG_Y,ORIG_Z,ORIG_Y1,ORIG_Z1,ORIG_PK_PK,
        2         ORIG_FREQ,ORIG_TEMP1,ORIG_TEMP2,ORIG_VOLT

        I=1
        DO WHILE(.TRUE.)
            I=I+1
            READ(2,4,END=6) TX,Y(I),TZ,TY1,TZ1,PK_PK_2(I)
            Y(I)=IFIX(Y(I)+0.5)
            IF(I.EQ.1) Z_SLICE1=TZ
            IF(NORM) THEN
                PK_PK_2(I)=PK_PK_2(I)/ORIG_PK_PK
            ELSE
                PK_PK_2_MAX=MAX(PK_PK_2_MAX,PK_PK_2(I))
            ENDIF
        END DO

6       Y(1)=I-2
        PK_PK_2(1)=I-2

        CLOSE(UNIT=2)

        IF(Z_SLICE.NE.Z_SLICE1) STOP 'SLICES ARE NOT THE SAME.'

        IF(NORM) THEN
            PK_PK_MAX=1.0
        ELSE
            PK_PK_MAX=MAX(PK_PK_MAX,PK_PK_2_MAX)
        ENDIF

        DO I=2,IFIX(X(1)+1),1
            PK_PK(I)=MIN(PK_PK_MAX,PK_PK(I))*((NUM_COLORS+1)/PK_PK_MAX)
        END DO

        DO I=2,IFIX(Y(1)+1),1
            PK_PK_2(I)=MIN(PK_PK_MAX,PK_PK_2(I))*((NUM_COLORS+1)/PK_PK_MAX)
        END DO

        CALL INIT_GRAPH

        CALL GRAPH_X_SLICE(X,PK_PK)

        CALL GRAPH_Y_SLICE(Y,PK_PK_2)

        IF(X(1).NE.Y(1)) TYPE *, 'WARNING; SLICE LENGTHS ARE DIFFERENT.'
        TYPE *, 'SLICE DATA'
        DO I=2,IFIX(MIN(X(1),Y(1)))+1,2
            IF(X(I).NE.Y(I)) TYPE *,X(I), ' <> ',Y(I)
            IF(X(I+1).NE.Y(I+1)) TYPE *,X(I+1), ' <> ',Y(I+1)
            TYPE 10,IFIX(X(I)),IFIX(PK_PK(I)+0.5),IFIX(PK_PK_2(I)+0.5),
            2         IFIX(X(I+1)),IFIX(PK_PK(I+1)+0.5),IFIX(PK_PK_2(I+1)+0.5)
        C10    FORMAT(' I2' 'I2' , 'I2'          'I2' 'I2' , 'I2)
        C      END DO

200    TYPE *
        TYPE *, 'ENTER ONE OF THE FOLLOWING:'

```

```
TYPE *, ' 1.  SAVE PICTURE TO DISK'
TYPE *, ' 2.  RESTART PROGRAM'
TYPE *, ' 3.  EXIT PROGRAM'
ACCEPT '(A1)', CH
```

```
IF (CH.EQ.'1') THEN
    CALL SAVE_TO_DISK
    GOTO 200
```

```
ELSE IF (CH.EQ.'2') THEN
    CALL ERASE_DISPLAY
    GOTO 100
```

```
C      CALL ERASE_DISPLAY
      GOTO 100
```

```
ELSE
    CALL ERASE_DISPLAY
    CALL EXIT
ENDIF
```

```
END
```

PROGRAM SHAPE

```
VIRTUAL DRIVE(16384),WAVE1(16384),WAVE2(16384),DC(16384)
INTEGER DRIVE,WAVE1,WAVE2,DC
```

```
INTEGER ROW,COL,NCHAR,I,OSCADR,GENADR,INFO(50),MAXVAL,MAXV
INTEGER KOUNT
REAL GEN,ATEMP,WTEMP,WATT,FREQ,RMS,RVAL,PERCNT
BYTE CTIME(9),FILE(80),STR(80),CFILE(15),CH
LOGICAL THRESH,ENABLE
```

```
COMMON /MAIN/ CTIME
```

```
DATA CTIME/9*0/,THRESH/.FALSE./,INFO/50*0/,MAXVAL/4097/
DATA PERCNT/0.25/,KOUNT/0/,ENABLE/.TRUE./
```

```
OSCADR = 4                !HP SCOPE ADDRESS
GENADR = 17               !GENERATOR ADDRESS
```

```
CALL IBIFC
CALL IBTERM(10)
CALL SETOSC(OSCADR)
CALL HEADER('SHAPE OSCILLATION DATA COLLECTION')
CALL IBSEOI('HEAD 0',,GENADR)
CALL GETGEN(GEN,FREQ,GENADR,OSCADR)
CALL TIME(CTIME)
CALL UTIME(CTIME,1,1)
CALL OPT(1,'CURRENT TIME: ',,CTIME)
CALL OPT(2,'LASER WATTAGE:',WATT,'WATTS')
CALL OPT(3,'GENERATOR: ',GEN,'VOLTS')
CALL OPT(4,'GENERATOR: ',FREQ/1000.,'KHZ')
CALL OPT(5,'AMBIENT TEMP: ',ATEMP,'DEG C')
CALL OPT(6,'WATER TEMP: ',WTEMP,'DEG C')
CALL OPT(7,'GET OSCILLOSCOPE WAVEFORMS',,,)
CALL OPT(8,'WRITE ACQUIRED WAVEFORMS',,,)
CALL OPT(9,'START THRESHOLD TRIGGER',,,)
CALL OPT(10,'GET NEW THRESHOLD',,,)
CALL OPT(11,'THRESHOLD: ',FLOAT(MAXVAL),'UNITS')
CALL OPT(12,'CRITERIA %: ',PERCNT*100.,'%')
CALL OPT(13,'ENABLE PANEL CONTROL',,,)
CALL OPT(14,'RESET "CURRENT FILE"',,,)
CALL OPT(15,'CURRENT FILE: ',,CFILE)
CALL OPT(16,'EXIT PROGRAM',,,)
```

```
CALL PROMPT('ENTER FIRST "CURRENT STATE" FILE NAME -> ')
```

```
ACCEPT S,NCHAR,(CFILE(I),I=1,14,1)
```

```
FORMAT(Q,14A1)
```

```
FILE(NCHAR+1)=0
```

```
CALL NXTFIL(CFILE)
```

```
OPEN(UNIT=1,NAME=CFILE,TYPE='NEW')
```

```
WRITE(1,6)
```

```
6 FORMAT('          SCATTER  GEN   LASER  GEN   AMB.  WAT.'/
2      ' TIME    MV DC    VOLTS  WATTS  FREQ  TEMP  TEMP')
```

```
MAXROW=8
```

```
ROW=1
```

```
COL=1
```

```

8      CALL IPOKE("44,IPEEK("44).OR."50100)
      CALL POS(18,1)
      TYPE 80,27
80     FORMAT('+'A1'[J<RETURN> CHOOSES SELECTION'/
2       ' <ARROWS> MOVES TO SELECTION'/
2       ' <SPACE> GETS CURRENT VALUES, WRITES THEM AND ',
2       'GETS A NEW THRESHOLD'/
2       ' <NUMBER> INCEREMENTS VOLTAGE BY <NUMBER>*0.001'/
2       '      R      PERFORMS: <SPACE> <1>'/
2       '      F      FORCES A TRIGGER'S)
      IF(ENABLE)CALL CLRESR(OSCADR)
      IF(ENABLE)CALL CLROER(OSCADR)

9      CALL POS(ROW+3,COL)
10     ICH=ITTINR()
      CALL UTIME(CTIME,ROW,COL)

      CALL GETADO(IVAL,8)

      IF(IVAL.GT.MAXVAL.AND.THRESH)GOTO 5000

      IF(ICH.EQ.70.OR.ICH.EQ.102)GOTO 4000
      IF(ICH.EQ.32.OR.ICH.EQ.82)GOTO 1800
      IF(ICH.GE.49.AND.ICH.LE.57)GOTO 6000
      IF(ICH.LE.0.OR.ICH.EQ.27.OR.ICH.EQ.91.OR.ICH.EQ.10)GOTO 10

      GOTO(100,200,300,400) ICH-64
      IF(ICH.EQ.13)GOTO 500
      IF(ICH.EQ.23)GOTO 7000
11     TYPE 12,7
12     FORMAT('+'A1$)
      GOTO 10

C UP ARROW

100    ROW=ROW-1
      IF(ROW.LE.0) ROW=MAXROW
      GOTO 9

C DOWN ARROW

200    ROW=ROW+1
      IF(ROW.GT.MAXROW) ROW=1
      GOTO 9

C RIGHT ARROW

300    COL=COL+40
      IF(COL.GT.41) COL=1
      GOTO 9

C LEFT ARROW

400    COL=COL-40
      IF(COL.LE.0) COL=41
      GOTO 9

```

C RETURN

```

500  CALL IPOKE("44,IPEEK("44).AND."127677)
      GOTO(1100,1500,1200,1700,1300,1400,1600,1900,2000,2200,11
          2      ,2100,2400,2500,11,2300)(ROW-1)*2+1+MAX0(0,COL-40)
      CALL ERRMSG('ILLEGAL SELECTION.')
      GOTO 11

```

C TIME

```

1100  CALL ERRMSG('CAN'T CHANGE THE TIME.')
      GOTO 8

```

C SET GENERATOR VOLTAGE

```

1200  CALL PROMPT('ENTER VOLTAGE (VOLTS) -> ')
      READ(5,1201,ERR=1210)RVAL
1201  FORMAT(G)
      GEN=RVAL
1202  CALL SETGEN(GEN,FREQ,GENADR,OSCADR)
      CALL GETGEN(GEN,FREQ,GENADR,OSCADR)
      CALL OPT(3,'GENERATOR:      ',GEN,'VOLTS')
      GOTO 8
1210  CALL ERRMSG('YOU MUST USE A REAL NUMBER.')
      GOTO 8

```

C AMBIENT TEMP

```

1300  CALL PROMPT('ENTER AMBIENT TEMPERATURE (DEG C) -> ')
      READ(5,1201,ERR=1210)RVAL
      ATEMP=RVAL
      CALL OPT(5,'AMBIENT TEMP: ',ATEMP,'DEG C')
      GOTO 8

```

C WATER TEMP

```

1400  CALL PROMPT('ENTER WATER TEMPERATURE (DEG C) -> ')
      READ(5,1201,ERR=1210)RVAL
      WTEMP=RVAL
      CALL OPT(6,'WATER TEMP:    ',WTEMP,'DEG C')
      GOTO 8

```

C LASER WATTAGE

```

1500  CALL PROMPT('ENTER LASER WATTAGE (WATTS) -> ')
      READ(5,1201,ERR=1210)RVAL
      WATT=RVAL
      CALL OPT(2,'LASER WATTAGE:',WATT,'WATTS')
      GOTO 8

```

C GET WAVEFORMS

```

1600  IF(.NOT.ENABLE)GOTO 1610
      CALL PROMPT('TRANSFER RECORDS 1 OR 2? ')
      ACCEPT *,IREC
      IF(IREC.NE.2)IREC=1

```

```

CALL GETWAV(1,DC,OSCADR,IREC)
CALL GETWAV(2,DRIVE,OSCADR,IREC)
CALL GETWAV(3,WAVE1,OSCADR,IREC)
CALL GETWAV(4,WAVE2,OSCADR,IREC)
WRITE(1,*) 'TRANSFERRING WAVEFORMS, IREC =',IREC
GOTO 8
1610 CALL ERRMSG('YOU MUST DISABLE PANEL CONTROL!')
GOTO 8

C SET GENERATOR FREQUENCY

1700 CALL PROMPT('ENTER FREQUENCY (KILOHERTZ) -> ')
READ(5,1201,ERR=1210) RVAL
FREQ=RVAL*1000.
CALL SETGEN(GEN,FREQ,GENADR,OSCADR)
CALL GETGEN(GEN,FREQ,GENADR,OSCADR)
CALL OPT(3,'GENERATOR: ',GEN,'VOLTS')
CALL OPT(4,'GENERATOR: ',FREQ/1000.,'KHZ')
GOTO 8

C GET/WRITE CURRENT STATE

1800 CALL UTIME(CTIME,ROW,COL)
CALL PROMPT('TRIGGERING ...')

CALL GETRMS(RMS)

CALL GETGEN(GEN,FREQ,GENADR,OSCADR)
CALL OPT(3,'GENERATOR: ',GEN,'VOLTS')
CALL OPT(4,'GENERATOR: ',FREQ/1000.,'KHZ')
WRITE(1,1807) (CTIME(I),I=1,8,1),RMS,GEN,WATT,FREQ/1000.,
2 ATEMP,WTEMP
1807 FORMAT('8A1' 'F7.3' 'F6.3' 'F5.3' 'F5.2' 'F4.1' 'F4.1')
CALL POS(13,1)
TYPE 1805,27,RMS
1805 FORMAT('+A1'[KCURRENT RMS VALUE: 'F7.3' MV'S)
IF(ICH.EQ.32.OR.ICH.EQ.82)GOTO 2200 !GET NEW THRESHOLD ON <SPACE> ALSO
GOTO 8

1810 TYPE 1811,NCHAR,(STR(I),I=1,LEN(STR),1)
1811 FORMAT(' ERROR: NCHAR='I'STR='80A1)
CALL EXIT

C WRITE WAVEFORMS

1900 CALL PROMPT('HOW MANY DATA POINTS? ')
ACCEPT *,N
IF(N.LE.0)GOTO 8
WRITE(1,*) 'WRITING WAVEFORMS, SIZE =',N
CALL NXTFIL(CFILE)
OPEN(UNIT=2,NAME=CFILE,TYPE='NEW')
WRITE(2,1901) (DC(I),I=1,N,1)
CLOSE(UNIT=2)

CALL NXTFIL(CFILE)
OPEN(UNIT=2,NAME=CFILE,TYPE='NEW')
WRITE(2,1901) (DRIVE(I),I=1,N,1)

```

```

1901  FORMAT(' 'I5)
      CLOSE (UNIT=2)

      CALL NXTFIL(CFILE)
      OPEN (UNIT=2, NAME=CFILE, TYPE='NEW')
      WRITE (2, 1901) (WAVE1 (I), I=1, N, 1)
      CLOSE (UNIT=2)

      CALL NXTFIL(CFILE)
      OPEN (UNIT=2, NAME=CFILE, TYPE='NEW')
      WRITE (2, 1901) (WAVE2 (I), I=1, N, 1)
      CLOSE (UNIT=2)

      CLOSE (UNIT=1)
      CALL NXTFIL(CFILE)
      GOTO 2510

C START THRESHOLD TRIGGERING

2000  THRESH=.NOT.THRESH
      IF (.NOT.THRESH) CALL OPT (9, 'START THRESHOLD TRIGGER',,,)
      IF (THRESH) CALL OPT (9, 'STOP THRESHOLD TRIGGER ',,,)
      IF (THRESH) WRITE (1, *) 'THRESHOLD TRIGGERING STARTED'
      IF (.NOT.THRESH) WRITE (1, *) 'THRESHOLD TRIGGERING STOPPED
2 DUE TO USER'
      GOTO 8

C CRITERIA PERCENTAGE

2100  CALL PROMPT('ENTER NEW CRITERIA PERCENT (IE 0.12 = 12%) -> ')
      ACCEPT *, PERCNT
      WRITE (1, *) 'CRITERIA PERCENTAGE:', IFIX (PERCNT*100.)
      CALL OPT (12, 'CRITERIA %: ', PERCNT*100., '%')
      GOTO 8

C GET NEW THRESHOLD VALUE

2200  CALL PROMPT('GETTING NEW THRESHOLD ...')
      MAXVAL=9999
      DO 2220 J=1, 20, 1
          MAXV=0
          DO 2210 I=1, 200, 1
              CALL GETADO (IVAL, 8)
              MAXV=MAX0 (IVAL, MAXV)
2210  CONTINUE
          MAXVAL=MIN0 (MAXVAL, MAXV)
2220  CONTINUE
      MAXVAL=MAX0 (MAXVAL+IFIX (PERCNT*FLOAT (MAXVAL)), 20)
      CALL OPT (11, 'THRESHOLD: ', FLOAT (MAXVAL), ' UNITS')
      WRITE (1, *) 'CURRENT THRESHOLD:', MAXVAL
      IF (ICH.EQ.82) GOTO 6000 !R MEANS INCREMENT GEN ALSO
      GOTO 8

C EXIT PROGRAM

2300  CLOSE (UNIT=1)
      CALL POS (24, 1)

```

CALL EXIT

C ENABLE/DISABLE REMOTE

```
2400  ENABLE=.NOT.ENABLE
      IF(ENABLE) II=IBREMO(1)
      IF(.NOT.ENABLE) II=IBREMO(0)
      IF(ENABLE) CALL OPT(13, 'ENABLE PANEL CONTROL ',,)
      IF(.NOT.ENABLE) CALL OPT(13, 'DISABLE PANEL CONTROL ',,)
      IF(ENABLE) WRITE(1,*) 'PANEL CONTROL USER DISABLED'
      IF(.NOT.ENABLE) WRITE(1,*) 'PANEL CONTROL USER ENABLED'
      GOTO 8
```

C RESET "CURRENT FILE"

```
2500  CALL PROMPT('DO YOU WANT TO KEEP THE CURRENT FILE (Y/N)?')
      ACCEPT 2501, CH
2501  FORMAT(A1)
      IF(CH.EQ.'Y'.OR.CH.EQ.'Y') GOTO 2502
      IF(.NOT.(CH.EQ.'N'.OR.CH.EQ.'N')) GOTO 2500
      CLOSE(UNIT=1, DISPOSE='DELETE')
      GOTO 2510

2502  CLOSE(UNIT=1)
      DO 2503 I=1,5,1
2503  CALL NXTFIL(CFILE)

2510  OPEN(UNIT=1, NAME=CFILE, TYPE='NEW')
      WRITE(1,6)
      GOTO 8
```

C FORCED TRIGGER

```
4000  CALL IBSEOI('START',, OSCADR)
      CALL MWAIT(128, OSCADR)
      CALL PROMPT('TRIGGER FORCED!')
      WRITE(1,*) '*** TRIGGER FORCED BY USER ***'
      IF(THRESH) WRITE(1,*)
1      'THRESHOLD TRIGGERING STOPPED DUE TO FORCING'
      THRESH=.FALSE.
      CALL WRTSET(OSCADR, GENADR)
      CALL OPT(9, 'START THRESHOLD TRIGGER',,)
      GOTO 8
```

C THRESHOLD CRITERIA MET

```
5000  TYPE 12,7
      CALL IBSEOI('START',, OSCADR)
      CALL MWAIT(128, OSCADR)
      TYPE 12,7
      CALL PROMPT('CRITERIA MET!')
      WRITE(1,*) '*** CRITERIA MET ** THRESHOLD:', IVAL
      THRESH=.FALSE.
      WRITE(1,*) 'THRESHOLD TRIGGERING STOPPED DUE TO CRITERIA'
      CALL WRTSET(OSCADR, GENADR)
      CALL OPT(9, 'START THRESHOLD TRIGGER',,)
      GOTO 8
```

C INCREMENT GEN

```

6000  IF(ICH.EQ.82)GEN=GEN+0.001
      IF(ICH.NE.82)GEN=GEN+FLOAT(ICH-48)*0.001
      GOTO 1202

```

C REPAINT SCREEN

```

7000  CALL HEADER('SHAPE OSCILLATION DATA COLLECTION')
      CALL OPT(1,'CURRENT TIME: ',,CTIME)
      CALL OPT(2,'LASER WATTAGE:',WATT,'WATTS')
      CALL OPT(3,'GENERATOR: ',,GEN,'VOLTS')
      CALL OPT(4,'GENERATOR: ',,FREQ/1000.,'KHZ')
      CALL OPT(5,'AMBIENT TEMP: ',ATEMP,'DEG C')
      CALL OPT(6,'WATER TEMP: ',WTEMP,'DEG C')
      CALL OPT(7,'GET OSCILLOSCOPE WAVEFORMS',,,)
      CALL OPT(8,'WRITE ACQUIRED WAVEFORMS',,,)
      CALL OPT(9,'START THRESHOLD TRIGGER',,,)
      CALL OPT(10,'GET NEW THRESHOLD',,,)
      CALL OPT(11,'THRESHOLD: ',,FLOAT(MAXVAL), 'UNITS')
      CALL OPT(12,'CRITERIA %: ',,PERCNT*100., '%')
      CALL OPT(13,'ENABLE PANEL CONTROL',,,)
      CALL OPT(14,'SET NEW "CURRENT FILE"',,,)
      CALL OPT(15,'CURRENT FILE: ',,CFILE)
      CALL OPT(16,'EXIT PROGRAM',,,)
      CALL POS(13,1)
      TYPE 1805,27,RMS
      GOTO 8

```

END

SUBROUTINE HEADER(STR)

BYTE STR(1)

TYPE 1,27,27,27

1 FORMAT('+'A1'[H'A1'[J'A1'#3'S)

TYPE 2,(STR(I),I=1,LEN(STR),1)

2 FORMAT('+'A1\$)

TYPE 3,27,27

3 FORMAT('+'A1'[2H'A1'#4'S)

TYPE 2,(STR(I),I=1,LEN(STR),1)

RETURN

END

SUBROUTINE OPT(NUM,STR1,VALUE,STR2)

INTEGER NUM,ROW,COL

BYTE STR1(1),STR2(1)

REAL VALUE

ROW=(NUM-1)/2+4

COL=1

IF(MOD(NUM,2).EQ.0)COL=41

CALL POS(ROW,COL)

IF(IADDR(STR1).EQ.-1)GOTO 100

TYPE 1,27

1 FORMAT('+'A1'[1M'S)

TYPE 2,(STR1(I),I=1,LEN(STR1),1)

```

2      FORMAT('+'A1$)
      TYPE 3,27
3      FORMAT('+'A1'(M'$)

100    IF (IADDR(VALUE).EQ.-1)GOTO 200
      IF (ABS(VALUE).GT.32767.)GOTO 150
      IF (IFIX(VALUE).NE.VALUE)GOTO 150
      TYPE 101,IFIX(VALUE)
101    FORMAT('+'I5$)
      GOTO 200
150    TYPE 102,VALUE
102    FORMAT('+'F7.3$)

200    IF (IADDR(STR2).EQ.-1)GOTO 300
      TYPE 2,' ',(STR2(I),I=1,LEN(STR2),1)

300    RETURN
      END
      SUBROUTINE POS(ROW,COL)
      INTEGER ROW,COL

      TYPE 1,27,ROW/10+48,MOD(ROW,10)+48,COL/10+48,MOD(COL,10)+48
1      FORMAT('+'A1'('2A1';'2A1'H'$)

      RETURN
      END
      SUBROUTINE PROMPT(STR)
      BYTE STR(1)

      CALL POS(22,1)
      TYPE 1,27
1      FORMAT('+'A1'(7M'$)
      TYPE 2,(STR(I),I=1,LEN(STR),1)
2      FORMAT('+'A1$)
      TYPE 3,27,27
3      FORMAT('+'A1'(M'A1'(J '$)

      RETURN
      END
      SUBROUTINE ERRMSG(STR)
      BYTE STR(1)

      CALL POS(23,1)
      TYPE 1,27,7
1      FORMAT('+'A1'(7M'A1$)
      TYPE 2,(STR(I),I=1,LEN(STR),1)
2      FORMAT('+'A1$)
      TYPE 3,27,27
3      FORMAT('+'A1'(M'A1'(J'$)
      CALL WAIT

      RETURN
      END
      INTEGER FUNCTION LEN(STR)
      BYTE STR(1)

      I=1
1      IF (STR(I).EQ.0)GOTO 2
      I=I+1

```

```

GOTO 1

2   LEN=I-1

RETURN
END
SUBROUTINE UTIME (CTIME, ROW, COL)
INTEGER ROW, COL
BYTE CTIME (9), OTIME (9)
DATA OTIME/9*0/

CALL TIME (CTIME)
IF (CTIME (8) .EQ. OTIME (8)) RETURN
DO 1 I=1, 8, 1
1   OTIME (I) =CTIME (I)
    OTIME (9) =0

CALL OPT (1, 'CURRENT TIME:      ', OTIME)
CALL POS (ROW+3, COL)

RETURN
END
SUBROUTINE GETGEN (GEN, FREQ, GENADR, OSCADR)
REAL GEN, FREQ
INTEGER GENADR, NCHAR
BYTE STR (80)

CALL IBSEOI ('FR?', , GENADR)
NCHAR=IBRECV (STR, 80, GENADR) -2
DECODE (NCHAR, 1, STR) FREQ
1   FORMAT (F12.3)

CALL IBSEOI ('AM?', , GENADR)
NCHAR=IBRECV (STR, 80, GENADR) -2
DECODE (NCHAR, 2, STR) GEN
2   FORMAT (F12.6)

IF (GEN.GE.0.03) CALL IBSEOI ('CHANNEL 2; RANGE 200.0 MV', , OSCADR)
IF (GEN.LT.0.03) CALL IBSEOI ('CHANNEL 2; RANGE 100.0 MV', , OSCADR)

RETURN
END
SUBROUTINE SETGEN (GEN, FREQ, GENADR, OSCADR)
REAL GEN, FREQ
INTEGER GENADR, NCHAR, OSCADR
BYTE STR1 (16), STR2 (16)
DATA STR1/'A', 'M', 5*0, '.', 6*0, 'V', 'R'/
DATA STR2/'F', 'R', 8*0, '.', 3*0, 'H', 'Z'/

ENCODE (12, 1, STR2 (3)) FREQ
1   FORMAT (F12.3)
CALL IBSEOI (STR2, 16, GENADR)

ENCODE (12, 2, STR1 (3)) GEN
2   FORMAT (F12.6)
CALL IBSEOI (STR1, 16, GENADR)

```

```

IF (GEN.GE.0.03) CALL IBSEOI('CHANNEL 2; RANGE 200.0 MV',, OSCADR)
IF (GEN.LT.0.03) CALL IBSEOI('CHANNEL 2; RANGE 100.0 MV',, OSCADR)

RETURN
END
SUBROUTINE GETWAV (CHAN, WAVE1, OSCADR, IREC)
VIRTUAL WAVE1 (16384)
INTEGER CHAN, OSCADR, WAVE1, I, J, K, LENG, NCHAR, IREC
BYTE STR (80), STEMP (4), WAVE (4096)

CALL IBSEOI('WAVEFORM FORMAT WORD',, OSCADR)
CALL CFILL('WAVEFORM SOURCE CHANNEL 0 RECORD 1', STR)
STR (25) = CHAN + 48
STR (34) = IREC + 48
CALL IBSEOI (STR,, OSCADR)

CALL IBTERM()
CALL IBSEOI('WAVEFORM DATA?',, OSCADR)
NCHAR = IBRECV (STEMP, 4, OSCADR)
LENG = (FLOAT (STEMP (3) .AND. "377") * 256. + FLOAT (STEMP (4) .AND. "377")) / 2.
CALL POS (14, 1)
TYPE 1701, LENG
1701  FORMAT ('+TOTAL NUMBER OF DATA POINTS:      'IS)
      K = 1
      DO 1720 I = 1, LENG, 2048
        CALL POS (15, 1)
        TYPE 1702, K - 1
1702  FORMAT ('+NUMBER OF DATA POINTS TRANSFERRED: 'IS)
      NCHAR = IBRECV (WAVE, 4096, OSCADR)
      DO 1710 J = 1, NCHAR, 2
        WAVE1 (K) = (WAVE (J) .AND. "377") * 256 + (WAVE (J + 1) .AND. "377")
        K = K + 1
1710  CONTINUE
1720  CONTINUE
      CALL POS (15, 1)
      TYPE 1702, K - 1
      CALL POS (14, 1)
      TYPE 1725, 27
1725  FORMAT ('+'A1'[{J}'$)
      CALL IBTERM (10)

RETURN
END
SUBROUTINE CFILL (STR, STR1)
BYTE STR (1), STR1 (1)
INTEGER I

I = 1
STR1 (1) = 0
1  IF (STR (I) .EQ. 0) GOTO 2
  STR1 (I) = STR (I)
  I = I + 1
  GOTO 1

2  STR1 (I) = 0
  RETURN
END
SUBROUTINE SETOSC (OSCADR)

```

```

INTEGER OSCADR
BYTE CH

CALL HEADER('HP 5183 INSTRUMENT SETTINGS')
CALL IBSEOI('PRESET;TRACE 4',,OSCADR)
CALL IBSEOI('SWEEP SINGLE;TRIGGER SOURCE EXTERNAL',,OSCADR)
CALL POS(5,1)
TYPE 2
2  FORMAT('+SINGLE SWEEP, EXTERNAL TRIGGERING'S)

CALL IBSEOI('MAIN 1.0 US',,OSCADR)
CALL POS(5,40)
TYPE 3
3  FORMAT('+SAMPLING RATE: 1.0 MICROSECONDS/SAMPLE'S)

CALL IBSEOI('TRIGGER LEVEL EXTERNAL 3.0 V',,OSCADR)
CALL IBSEOI('TRIGGER MODE EXTERNAL POS',,OSCADR)
C  CALL IBSEOI('TRIGGER POSITION -50 PERCENT',,OSCADR)
CALL POS(7,1)
TYPE 4
4  FORMAT('+TRIGGER SETTINGS'S)
CALL POS(8,1)
TYPE 5
5  FORMAT('+EXTERNAL LEVEL 3.0V'S)
CALL POS(8,40)
TYPE 6
6  FORMAT('+POSITIVE SLOPE TRIGGERING'S)
CALL POS(9,1)
C  TYPE 7
C7  FORMAT('+50% PRE-TRIGGERING'S)
CALL POS(9,40)
TYPE 8
8  FORMAT('+TRIGGER SOURCE: HP GENERATOR SYNC'S)

CALL POS(11,1)
TYPE 80
80  FORMAT('+CHANNEL SETTINGS'S)
CALL POS(12,1)
TYPE 9
9  FORMAT('+ CHANNEL 1      CHANNEL 2      CHANNEL 3      ',
2'CHANNEL 4'S)

CALL IBSEOI('CHANNEL 1;CHANNEL CONFIGURE SINGLE',,OSCADR)
CALL IBSEOI('CHANNEL 2;CHANNEL CONFIGURE SINGLE',,OSCADR)
CALL IBSEOI('CHANNEL 3;CHANNEL CONFIGURE SINGLE',,OSCADR)
CALL IBSEOI('CHANNEL 4;CHANNEL CONFIGURE DIFF',,OSCADR)
CALL POS(13,1)
TYPE 10
10  FORMAT('+SINGLE ENDED   SINGLE ENDED   SINGLE ENDED   ',
?'DIFFERENTIAL'S)

CALL IBSEOI('CHANNEL 1;RANGE 1.0 V;COUPLING DC',,OSCADR)
CALL IBSEOI('CHANNEL 2;RANGE 100.0 MV;COUPLING AC',,OSCADR)
CALL IBSEOI('CHANNEL 3;RANGE 200.0 MV;COUPLING AC',,OSCADR)
CALL IBSEOI('CHANNEL 4;RANGE 200.0 MV;COUPLING AC',,OSCADR)
CALL POS(14,1)
TYPE 11
11  FORMAT('+ +/-1.0V      +/-100MV      +/-200MV      ',

```

```

2'+/~200MV'$)
CALL POS(15,1)
TYPE 12
12  FORMAT('+      DC      AC      AC      ',
2'AC'$)

CALL IBSEOI('LENGTH 32768',,OSCADR)
CALL POS(16,1)
TYPE 13
13  FORMAT('+32768 SAMPLES 32768 SAMPLES 32768 SAMPLES ',
2'32768 SAMPLES'$)

CALL CLRESR(OSCADR)
CALL CLROER(OSCADR)

CALL PROMPT('PRESS <RETURN> TO CONTINUE')
ACCEPT 100,CH
100  FORMAT(A1)

RETURN
END
SUBROUTINE MWAIT(MASK,INS)
IMPLICIT INTEGER (A-Z)

BYTE ERR(100)
BYTE STR(19)
DATA STR/'O','E','E',9*' ',';','E','S','E',' ','3','6'/

ENCODE(8,1,STR(5))MASK
1  FOPMAT(I8)

CALL IBSEOI(STR,19,INS)

10  ISTAT=IBSTS(INS).AND."377
IF (MOD(ISTAT,64)/32.EQ.1)GOTO 1000
IF (MOD(ISTAT,2).EQ.0)GOTO 10
RETURN

1000 CALL IBSEOI('ERR STRING ?',,INS)
N=IBRECV(ERR,100,INS)
TYPE *,'ERROR!!!!'
TYPE 1001,(ERR(I),I=1,N,1)
1001 FORMAT(' '100A1)
STOP

END

SUBROUTINE CLRESR(OSCADR)
IMPLICIT INTEGER (A-Z)
BYTE STR(80)

CALL IBSEOI('ESR?',,OSCADR)
N=IBRECV(STR,80,OSCADR)

RETURN
END
SUBROUTINE CLROER(OSCADR)
IMPLICIT INTEGER (A-Z)

```

```

BYTE STR(80)

CALL IBSEOI('OER?',,OSCADR)
N=IBRECV(STR,80,OSCADR)

RETURN
END
SUBROUTINE GETAD0(IVAL,CHAN)
INTEGER IVAL,CHAN

IVAL=IPEEK("171002)
CALL IPOKE("171000,CHAN*256+1)
1 IF(IPEEK("171000).LT.0)GOTO 1
IVAL=IABS(IPEEK("171002)-2048)

RETURN
END
SUBROUTINE GETRMS(RMS)
INTEGER I,IVAL
REAL RMS,FACTOR
DATA FACTOR/2.5/          !CONVERT A/D TO MV

RMS=0.0

DO 1810 I=1,500,1
    CALL GETAD0(IVAL,0)
    RMS=RMS+IVAL
1810 CONTINUE

RMS=(RMS/500.)*FACTOR

RETURN
END
SUBROUTINE NXTFIL(FILE)
BYTE FILE(15),PREFIX
INTEGER I,N,START
DATA N/0/

IF(N.GE.99)GOTO 100
IF(N.LE.0)GOTO 200

N=N+1
5 I=START
FILE(I)=PREFIX
I=I+1
IF(N.LE.9)GOTO 10
FILE(I)=N/10+48
I=I+1
FILE(I)=MOD(N,10)+48
I=I+1
GOTO 20

10 FILE(I)=N+48
I=I+1

20 FILE(I)='.'
FILE(I+1)='D'
FILE(I+2)='A'

```

```

FILE(I+3)='T'
FILE(I+4)=0

CALL OPT(15,'CURRENT FILE: ',,')
CALL OPT(15,'CURRENT FILE: ',,FILE)
RETURN

100  N=1
      PREFIX=PREFIX+1
      IF(PREFIX.GT.'Z') PREFIX='A'
      GOTO 5

200  START=1
      DO 210 I=1,4,1
210  IF(FILE(I).EQ.':') START=I+1

      PREFIX=FILE(START)
      N=FILE(START+1)-48
      IF(FILE(START+2).EQ.':') GOTO 5
      N=N*10+(FILE(START+2)-48)
      GOTO 5

      END
      SUBROUTINE WRTSET(OSCADR,GENADR)
      INTEGER OSCADR,GENADR
      BYTE STR(80)

      WRITE(1,*)'-----'
      WRITE(1,*)'GENERATOR SETTINGS'

      CALL IBSEOI('FR?',,GENADR)
      NCHAR=IBRECV(STR,80,GENADR)-2
      WRITE(1,*)'      FREQ: ',(STR(I),I=1,NCHAR,1)

      CALL IBSEOI('AM?',,GENADR)
      NCHAR=IBRECV(STR,80,GENADR)-2
      WRITE(1,*)'      VOLT: ',(STR(I),I=1,NCHAR,1)

      WRITE(1,*)'HP SCOPE SETTINGS'

      CALL IBSEOI('CHANNEL 1;RANGE?',,OSCADR)
      NCHAR=IBRECV(STR,80,OSCADR)-2
      WRITE(1,*)'      CHANNEL 1 RANGE: ',(STR(I),I=1,NCHAR,1)
      CALL IBSEOI('CHANNEL 2;RANGE?',,OSCADR)
      NCHAR=IBRECV(STR,80,OSCADR)-2
      WRITE(1,*)'      CHANNEL 2 RANGE: ',(STR(I),I=1,NCHAR,1)
      CALL IBSEOI('CHANNEL 3;RANGE?',,OSCADR)
      NCHAR=IBRECV(STR,80,OSCADR)-2
      WRITE(1,*)'      CHANNEL 3 RANGE: ',(STR(I),I=1,NCHAR,1)
      CALL IBSEOI('CHANNEL 4;RANGE?',,OSCADR)
      NCHAR=IBRECV(STR,80,OSCADR)-2
      WRITE(1,*)'      CHANNEL 4 RANGE: ',(STR(I),I=1,NCHAR,1)

      CALL IBSEOI('MAIN?',,OSCADR)
      NCHAR=IBRECV(STR,80,OSCADR)-2
      WRITE(1,*)'      SAMPLING RATE: ',(STR(I),I=1,NCHAR,1)
      WRITE(1,*)'-----'

```

RETURN
END

REFERENCES

- G. Mie. "Beitrage fur optik truber medier, speziell kolloidaler metallosungen". Ann. Phys. **25**, 377-445 (1908).
- P. Debye. "Der lichtdruck auf kugeln von beliebigem material". Ann. Phys. **30**, 57-136 (1909).
- J. Stratton. Electromagnetic Theory. (McGraw Hill, New York, 1941).
- H. Lamb. Hydrodynamics, (6th ed. Dover Publications, New York, 1945).
- N.W. McLachlan. Theory and application of Mathieu functions. (Clarendon, Oxford, 1947).
- G.I. Taylor. Proc. Roy. Soc. (London) **A201**, 192 (1950).
- M.S. Plesset. "On the stability of fluid flows with spherical symmetry". J. Appl. Phys. **25**, 96-98 (1954).
- C. Devin. "Survey of thermal, radiation, and viscous damping of pulsating air bubbles in water". JASA **31**, 1654-1667 (1959).
- M. Borne and E. Wolf. Principles of Optics. (Pergamon, New York, 1959).

C. Hayashi. Nonlinear Oscillations in Physical Systems. (McGraw-Hill, New York, 1964).

W.L. Nyborg. "Radiation pressure on a small rigid sphere". JASA 42, 947-952 (1967).

A.I. Eller. "Force on a bubble in a standing acoustic wave". JASA 43, 170-171 (1968).

L.M. Milne-Thomson. Theoretical Hydrodynamics. (Macmillan, New York, 1968).

L.A. Crum and A.I. Eller. "The motion of bubbles in a stationary sound field". Tech. Mem. 61, ARL Harvard Univ. (1969).

M. Kerker. The scattering of light and other electromagnetic radiation. (Academic, New York, 1969).

A.I. Eller and L.A. Crum. "Instability of the motion of a pulsating bubble in a sound field". JASA 47, 762-767 (1970).

A.I. Eller. "Damping constants of pulsating bubbles". JASA 47, 1469-1470 (1970).

L.A. Crum. "The acoustic radiation pressure on a liquid droplet in a stationary sound field". Tech. Report, U.S. Naval Academy (1970).

R.K. Gould. "Rectified diffusion in the presence of, and absence of, acoustic streaming". JASA **56**, 1740-1746 (1974).

A. Prosperetti. "Viscous effects on perturbed spherical flows". Q. Appl. Math. 339-352 (1977).

R. Clift, J.R. Grace, and M.E. Weber. Bubbles, Drops and Particles. (Academic, New York, 1978).

P. Hall and G. Seminara. "Nonlinear oscillations of non-spherical cavitation bubbles in acoustic fields". J. Fluid. Mech. **101**, 423 (1980).

W. J. Wiscombe. "Improved Mie scattering algorithms". Appl. Opt. **19**, 1505-1509 (1980).

L.A. Crum and G.M. Hansen. "Generalized equations for rectified diffusion". JASA **72**, 1586-1592 (1982).

L.A. Crum. "The polytropic exponent of gas contained within air bubbles pulsating in a liquid". JASA **73**, 116-120 (1983).

L.A. Crum and A. Prosperetti. "Nonlinear oscillations of gas bubbles in liquids: An interpretation of some experimental results". JASA **73**, 121-127 (1983).

G.M. Hansen. "Mie scattering as a technique for the sizing of air bubbles". Dissertation, U. of Mississippi (1983).

A. A. Atchley. "Acoustic cavitation and bubble dynamics". Dissertation, U. of Mississippi (1985).

A. Proseperetti and Lezzi. "Bubble dynamics in a compressible liquid. Part 1. First-order theory". J. Fluid Mech. **168**, 457-478 (1986).

P.M. Morse and K.U. Ingard. Theoretical acoustics. (McGraw-Hill, New Jersey, 1986).

IMSL. Mathematical software library. (IMSL, Houston, 1987).

A. Prosperetti, L.A. Crum and K. Commander. "Nonlinear bubble dynamics". JASA **83**, 502-514 (1988).

B.T. Unger and P.L. Marston. "Optical levitation of bubbles in water by the radiation pressure of a laser beam: an acoustically quiet levitator". JASA **83**, 970-975 (1988).

R.G. Holt. "Experimental observation of the nonlinear response of single bubbles to an applied acoustic field". Dissertation, U. of Mississippi (1988).

C.C. Church. Private communication, (1989).

D.F. Gaitan. Private communication, (1989).

R.G. Holt. Private communication, (1989).

BIOGRAPHICAL SKETCH OF THE AUTHOR

Steven DeWitt Horsburgh, [REDACTED]

[REDACTED]. Attended high school in Mobile, Alabama at Independent Methodist School. Graduated in 1978. Bachelor of Science degree in Physics and Computer Science at the University of Mississippi in 1983. Masters of Science degree in Atmospheric Physics at the University of Mississippi in 1984. Doctor of Philosophy degree in Physics with a dissertation in Nonlinear Dynamics/Acoustics at the University of Mississippi in 1990. Career objectives include directing a Computational Physics department in a research laboratory environment. Employed by the National Center for Physical Acoustics from 1987 to present. Former employer was the Physics department at the University of Mississippi in conjunction with the National Severe Storms Laboratory in Norman, Oklahoma from 1979 to 1986.

Permanent address [REDACTED]

Steven D. H. [REDACTED]

P.O. [REDACTED]

University of Mississippi [REDACTED]

January 1990

DISTRIBUTION LIST FOR
NCPA LC.01.90
TECHNICAL REPORT UNDER CONTRACT N0014-87-K-0019

COPY NO.

1	Chief of Naval Research Office of Naval Research Department of the Navy Attn: Marshall Orr
2	Director Naval Research Laboratory Washington, D.C. 20375 Attn: Library
3	Commander David Taylor Naval Ship Research and Development Center Target Physics Branch Bethesda, MD 20084 Attn: S. Schreppler (Code 1965)
4	Superintendent Naval Postgraduate School Department of Physics Monterey, CA 93940 Attn: Anthony Atchley
5-9	Commanding Officer and Director Defense Technical Information Center Bldg. 5, Camerson Station 5010 Duke Street Alexandria, VA 22314
10-20	Dr. L.A. Crum Associate Director for Basic Research National Center for Physical Acoustics University, MS 38677
21-30	Dr. Steve Horsburgh National Center for Physical Acoustics University of Mississippi University, MS 38677

- 31 Jane E. Crum
Acting Business Manager
National Center for Physical Acoustics
University of Mississippi
University, MS 38677
- 32 J.E. Matthews
NOARL Numerical Modeling Division
NSTL, MS 39529-5004
- 33 Phil Marston
Department of Physics
Washington State University
Pullman, WA 99164
- 34 Bob Gould
Department of Physics
Middlebury College
Middlebury, VT 05753