

ADA 027578

Nonlinear Scaling Laws for Parametric Receiving Arrays

Part II Numerical Analysis

prepared under:

Contract N00039-75-C-0259
ARPA Order 2910
Program Code 5G10

COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION

Authors:

J. W. Kesner

Westinghouse Electric Corporation
Oceanic Division
Annapolis, Maryland

F. H. Fenlon

Applied Research Laboratory
Pennsylvania State University
State College, Pennsylvania

June 30, 1976

Westinghouse Electric Corporation
✓ Oceanic Division
P. O. Box 1488
Annapolis, Maryland 21404

DDC
RECEIVED
JUL 30 1976
A

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

6 Nonlinear Scaling Laws for Parametric Receiving Arrays.

Part II. Numerical Analysis,

prepared under:

15
Contract N00039-75-C-0259,
✓ ARPA Order-2910
Program Code 5G10

Authors:

10 J. W. / Kesner

Westinghouse Electric Corporation
Oceanic Division
Annapolis, Maryland

F. H. / Fenlon

Applied Research Laboratory
Pennsylvania State University
State College, Pennsylvania

11 30 June 30, 1976

Westinghouse Electric Corporation
Oceanic Division
P. O. Box 1488
Annapolis, Maryland 21404

12 120p.

407381

LB

**BEST
AVAILABLE COPY**

Nonlinear Scaling Laws for Parametric Receiving Arrays

Part II Numerical Analysis

by

J. W. Kesner and F. H. Fenlon

ABSTRACT

This report outlines the procedures used in a computer program for calculating nonlinear scaling laws for parametric receiving arrays. The basic equations, the program input and output, the program listing, and sample computer runs are included. The program output is in the form of curves for harmonic amplitudes as a function of range, the "extra dB loss" as a function of range, and parametric receiving array beam patterns at given ranges. Both detailed analytical solutions and approximate solutions are available in the computer program.

ACCESSION (G)	
NTIS	White Section ☒
DDO	Buff Section ☐
UNANNOUNCED	☐
JUSTIFICATION	
<i>Letter on file</i>	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
<i>A</i>	

TABLE OF CONTENTS

	PAGE
THEORY	1
APPROXIMATE EXPRESSIONS	4
PROGRAM INPUT	6
PROGRAM OUTPUT	8
COMMENTS ON THE PROGRAM	10
REFERENCE	12
APPENDIX A	A1
FIGURES	F1
PROGRAM LISTING	L1

THEORY

Part I of this report covers the theoretical analysis that forms the basis of the computer program. The salient equations are presented here to indicate what the computer program does.

The spectral amplitudes ψ_n of the axial pressure field are given by the solutions to the differential equations

$$\frac{d}{dR}(\psi_n) = -(A+C)\psi_n - j(B-SWCN)\psi_n - j \sum_{m=-\infty}^{\infty} D \psi_{n-m} \psi_m \quad (1)$$

where ψ_n is the complex spectral amplitude of the n^{th} harmonic in the pressure field

R is the normalized range = range/Rayleigh distance

$A = 0$ for plane wave cases

$= 1/R$ for spherical wave cases

$= R/(1+R^2)$ for mixed (plane and spherical wave) cases

$B = 1/(1+R^2)$ for mixed cases with phase

$= 0$ for plane, spherical, and mixed-no phase cases

$C = n^2(f_s/f_o)^2(\alpha_o r_o)ATTF$

f_s/f_o = source frequency to pump frequency ratio

$\alpha_o r_o$ = attenuation coefficient at f_o times the Rayleigh distance

$ATTF = 1 + 0.69145 \cdot SALN / (1 + n^2 \cdot SWK^2)$

$SALN$ = water salinity in parts per 1000

SWK = sea water K

$D = (n/4)(f_s/f_o)\sigma_o$

σ_o = scaled source tone

$SWCN = 0.69145 \cdot SALN \cdot DDT \cdot C_o \cdot (k_o r_o) \cdot n / (1 + n^2 \cdot SWK^2)$

C_o = small signal speed of sound (=1500 m/s in program)
 $k_o r_o$ = wavenumber times Rayleigh distance at f_o
 $DDT = 0.538 \cdot 10^{-9} \cdot (1 - 6.54 \cdot 10^{-4} \cdot PRES)$
 $PRES$ = pressure in atmospheres.

A differential equation solving routine, VODQ, which is described in Appendix A, is used to solve equation (1) for ψ_n . Only those equations which are non-zero are included in the solution vector. The program user specifies the number (NHL) of lower harmonics (of f_s) and the number (NHH) of higher harmonics (of f_o) that he wants to consider. These harmonics are the only ones that are used in the differential equation. For example if $NHL = 3$, $NHH = 2$, and $f_o/f_s = 10$, then the solution vector would only contain

$$\psi_1, \psi_2, \psi_3, \psi_7, \psi_8, \psi_9, \psi_{10}, \psi_{11}, \psi_{12}, \psi_{13}, \psi_{17}, \psi_{18}, \psi_{19}, \psi_{20}, \psi_{21}, \psi_{22}, \psi_{23}.$$

By definition $\psi_o = 0$.

and $\psi_{-n} = \psi_n^* = \text{complex conjugate of } \psi_n$.

By specifying NHL, NHH, and f_o/f_s the infinite sum in equation (1) becomes a finite sum over at most

$$NHL + NHH \cdot (2 \cdot NHL + 1)$$

different positive values of m . For harmonics other than the ones determined by NHL, NHH, and (f_o/f_s) , $\psi_n = 0$.

The initial values of ψ_n that are used depend on the type of problem that is being solved. If the user simply wants to see the harmonics generated by a single frequency then the initial value of $\psi_1 = (0+j, 1)$ and all other initial ψ_n values are zero.

If the user wants an up-conversion case then

$$\psi_{\text{NFR}} = (0 + j.1) \quad \text{where } \text{NFR} = f_o/f_s$$

$$\text{and } \psi_1 = (0 + j.P)$$

where $P = \text{PSO}$ for a plane wave case

$$= \text{PSO} * \text{EXP}(-\alpha_o r_o / \text{NFR}^2) \quad \text{otherwise.}$$

$\text{PSO} =$ signal to pump amplitude ratio.

At each normalized range R the incoming signal is fed into the differential equation as

$$\psi_1 = \text{PSO} * \text{EXP}(-\alpha_o r_o * R / \text{NFR}^2) * \{\sin(\text{ARG}) + j \cos(\text{ARG})\}$$

$$\text{where } \text{ARG} = 2 * (k_o r_o) * R * (f_s / f_o) * \sin^2(\theta / 2)$$

and $\theta =$ angle of intersection between signal and pump wave normals.

If the user is interested in a difference frequency case then

$$\psi_{\text{NFR}-1} = (0 + j.0.5) \quad \text{and} \quad \psi_{\text{NFR}} = (0 + j.0.5)$$

for the two primary frequencies and the difference frequency amplitude will be generated in ψ_1 .

The extra dB loss for ψ_1 is computed from the formulas

$$\text{EXDB} = - \{20 * \log_{10}(\psi_1(R)) + 8.686 * \alpha_o r_o * R\} \quad \text{for plane wave cases}$$

$$\text{EXDB} = - \{20 * \log_{10}(\psi_1(R)) + 8.686 * \alpha_o r_o * (R-1) + 20 * \log_{10}(R)\} \quad \text{for spherical wave cases}$$

$$\text{EXDB} = - \{20 * \log_{10}(\psi_1(R)) + 8.686 * \alpha_o r_o * R + 20 * \log_{10}(\sqrt{1+R^2})\} \quad \text{for mixed cases.}$$

APPROXIMATE EXPRESSIONS

Approximate expressions for $\psi_1(R)$ and $\psi_{NFR+1}(R, \theta)$ were generated in part I of this report. These are also included in the computer program as an alternative to solving the differential equations. The user has the choice of two expressions for $\psi_1(R)$. One uses the El function and the other involves a definite integral. The integral form seems to give better results. These expressions are given below.

Approximate expression for $\psi_1(R)$ --- El form.

$$\psi_1(R) = \frac{\text{EXP}(-\alpha_o r_o * R)}{\text{EXP}(-\alpha_o r_o * R) + \text{RP} \sqrt{1 + (\sigma_o/2)^2} \text{EXP}(4\alpha_o r_o) \{ \text{El}(2\alpha_o r_o) - \text{El}(2\alpha_o r_o * \text{RP}) \}^2}$$

where $\text{RP} = R$ for spherical wave cases

and $\text{RP} = 1+R$ for mixed cases.

Approximate expression for $\psi_1(R)$ --- integral form.

$$\psi_1(R) = \frac{\text{EXP}(-\alpha_o r_o * R)}{\text{RP} \sqrt{1 + (\sigma_o/2)^2} \int_a^b \text{EXP}(-2\alpha_o r_o * R) / \text{RP} dR}$$

where $a = 0, b = R, \text{RP} = 1$ for plane wave cases

$a = 1, b = R, \text{RP} = R$ for spherical wave cases

$a = 0, b = R, \text{RP} = \sqrt{1+R^2}$ for mixed cases.

Approximate expression for $\psi_{NFR+1}(R, \theta)$

$$\psi_{NFR+1}(R, \theta) = \{ (\sigma_o/2)^2 \text{EXP}(-\alpha_{NFR+1} r_o * R) / \text{RP} \} * \int_a^b \text{XP} * \psi_1(X) * \text{EXP}(\alpha_o r_o * X) * \cos(KX) dX \quad (2)$$

where $\alpha_{NFR+1} r_o = (1 + f_s/f_o)^2 \alpha_o r_o$

$$K = 2(k_s r_o) \sin^2(\theta/2)$$

$$k_s r_o = (f_s/f_o) k_o r_o$$

$$k_o r_o = (k_o a)^2 / 2$$

$k_o a$ = wavenumber at f_o times radius of circular piston
projector

$a = 0, b = R, RP = 1, XP = 1$ for plane wave cases

$a = 1, b = R, RP = R, XP = X$ for spherical wave cases

$a = 0, b = R, RP = \sqrt{1+R^2}, XP = \sqrt{1+X^2}$ for mixed cases

θ = angle of intersection between signal and pump wave normals.

PROGRAM INPUT

The computer program may be run from a remote terminal or by cards as a batch job. The data for the program is in free field format which means that the data does not have to be in any particular column.

Input variables which begin with the letters I, J, K, L, M, N are integers and should not have any decimal point. All other variables must have a decimal point. For lines of input data that have more than one number the data values must be separated by commas.

Input 1 NOC = number of separate cases being run

Input 2 LUP, LPS, NFR, NHL, NHH, NAR, NSG, LAP, NSC, IBM

LUP = 1 for one frequency harmonic studies

 = 2 for up-conversion cases

 = 3 for difference frequency cases

LPS = 0 for plane wave cases

 = 1 for spherical wave cases

 = 2 for mixed cases with phase

 = 3 for mixed cases without phase

NFR = f_o/f_s = pump frequency to signal frequency ratio

NHL = number of lower harmonics (of f_s)

NHH = number of higher harmonics (of f_o)

NAR = number of $\alpha_o r_o$ values

NSG = number of σ_o values

LAP = 1 if the approximate expressions are to be used

 = 0 if the analytical solution is to be used

NSC = 0 for range scaling without α_o

 = 1 for range scaling with σ_o

IBM = 1 for beam pattern (only if LUP = 2 and LAP = 1)

 = 0 for no beam pattern

Input 3 XMAX = maximum range scale value

Input 4 AR(L), L = 1, ..., NAR $\alpha_o r_o$ values

Input 5 SG(L), L = 1, ..., NSG σ_o values

Input 6 SALN, PRES, TEMC, ITA, SWK, AKOA

 SALN = salinity of water in parts per 1000

 PRES = pressure in atmospheres

 TEMC = temperature in degrees centigrade
 (not used in present version of program)

 ITA = 1 for Marsh & Schulkin form

 = 2 for Russian form
 (not used in present version of program)

 SWK = sea water K

 AKOA = $k_o a$ = wavenumber at f_o times piston radius

Input 7 IAP (input only if LAP = 1)

 IAP = 1 for El approximate form

 = 2 for integral approximate form

Input 8 REM, TMAX (input only if IBM = 1)

 REM = normalized range R at which beam pattern is taken

 TMAX = maximum angle in degrees between pump and signal
 normals. The beam pattern is from 0° (on axis)
 to TMAX.

Input 9 PSO, LPP (input only if LUP = 2)

 PSO = signal to pump amplitude ratio

 LPP = 1 for only the ψ_{NFR-1} curve plotted

 = 2 for only the ψ_{NFR+1} curve plotted

 = 3 for both the ψ_{NFR-1} and ψ_{NFR+1} curves plotted

PROGRAM OUTPUT

The program output is in the form of curves for the axial harmonic amplitudes ($\psi_1(R)$, $\psi_2(R)$, $\psi_{NFR-1}(R)$, $\psi_{NFR+1}(R)$) as a function of range, the "extra dB loss" of $\psi_1(R)$ as a function of range, and parametric receiving array beam patterns ($\psi_{NFR+1}(R, \theta)$) at given ranges. A variety of range scales are used. For plane wave cases the range scale is $\sigma_0 R$ or R . For spherical wave cases the range scale is $\sigma_0 \ln(R)$ or $\ln(R)$. For mixed cases the range scale is $\sigma_0 \operatorname{arcsinh}(R)$ or $\operatorname{arcsinh}(R)$. The "extra dB loss" curves are plotted against R . The $\psi_{NFR-1}(R)$ and $\psi_{NFR+1}(R)$ curves are plotted against R for plane, spherical, and mixed cases.

Many cases were run during the development of the program in order to test the validity of the answers that the computer program was giving. Several of these test cases are included in the figures to show the comparisons between different assumptions, such as plane wave, spherical wave, and mixed wave cases, and to show the accuracy of the approximate expressions. Several figures are included to show the versatility of the computer program since it will handle harmonic studies, up-conversion cases, and beam pattern studies. All the figures shown were also chosen with the idea of demonstrating how the harmonic amplitudes and beam patterns change with the various parameters, such as $\alpha_0 r_0$, σ_0 , and f_0/f_s .

Figures 1 and 2 are plane wave cases that show $\psi_1(R)$, $\psi_2(R)$, and $\psi_3(R)$ for two values of $\alpha_0 r_0$ and four values of $\sigma_0/\alpha_0 r_0$.

Figures 3 and 4 are spherical wave cases that show $\psi_1(R)$, $\psi_2(R)$, and extra dB loss for two values of $\alpha_0 r_0$.

Figures 5, 6, and 7 are mixed wave cases that show $\psi_1(R)$, $\psi_2(R)$, and extra dB loss for three values of $\alpha_0 r_0$ and compare the curves generated by the approximate and analytical methods. In addition, figure 7 covers a range of five σ_0 values.

Figure 8 compares the spherical and mixed wave curves for ψ_2/ψ_1 and ψ_3/ψ_1 with similar curves previously generated by Fenlon¹.

Figure 9 shows the similarity of the two up-converted signals $\psi_{\text{NFR}-1}(\text{R})$ and $\psi_{\text{NFR}+1}(\text{R})$.

Figure 10 compares $\psi_{\text{NFR}+1}(\text{R})$ as generated with the mixed phase and mixed no phase assumptions and with the mixed approximate integral expressions.

Figure 11 shows $\psi_{\text{NFR}+1}(\text{R})$ for three values of f_o/f_s .

Figure 12 shows $\psi_{\text{NFR}+1}(\text{R})$ for two extreme values of $\alpha_o r_o$ and two extreme values of σ_o .

Figure 13 shows up-converted beam patterns for three values of f_o/f_s at two different ranges. One range is at the peak of the $\psi_{\text{NFR}+1}(\text{R})$ curve and the other range is in the far-field of the $\psi_{\text{NFR}+1}(\text{R})$ curve.

Figure 14 shows up-converted beam patterns for four value of σ_o at two different ranges. Some of these patterns are near-field patterns and some are far-field patterns because the peak of the $\psi_{\text{NFR}+1}(\text{R})$ curve changes with σ_o .

COMMENTS ON THE PROGRAM

The user should have a good idea of the normalized range scale he is interested in so that he can input an appropriate value for XMAX. It was found that using NSC = 0 (no σ_0 scale) was often more convenient when plotting curves for a range of σ_0 values. For up-conversion cases using the approximate expressions it is important to use a value of XMAX that is not too large because the $\psi_1(R)$ curve is precomputed at a finite number of points over the range up to XMAX and interpolation is used for values of R other than those prestored. Because of the steep rise and peak of the $\psi_1(R)$ curve it is necessary to have several points in this region. Otherwise, a poor representation of $\psi_1(R)$ will result and the resulting $\psi_{NFR+1}(R)$ curve will be no good. Program running cost was the main reason for prestoring the $\psi_1(R)$ curve.

One suggested change to the program is to include a multiplying factor G_m inside the summation of equation (1), but only for the $m = -\infty$ to $m = 0$ terms in the mixed with phase case. The suggested G_m term is

$$G_m = 1/(1 + 2m/((1+R^2)*(f_0/f_s))).$$

Another suggestion for making the program more complete is to replace the term $\cos(KR)$ in equation (2) with the term $\sin(KR) + j \cos(KR)$.

If the product KR in equation (2) is large enough then false lobes can appear because of the accuracy limitations in numerical integration of a rapidly varying function. These false lobes have shown up in figure 13 where the product KR is very large for some of the cases.

As an added feature, the computer program has been set up so that difference frequency cases can also be run. Most of the necessary changes to the program have been made, but have not been thoroughly checked out. If anyone would like to use the difference frequency part of the program it is suggested that he contact the authors first.

For more details of exactly what the computer program does, the reader is referred to the program listing which is included in this report.

REFERENCE

1. F.H. Fenlon, "A Recursive Procedure for Computing the Nonlinear Spectral Interactions of Progressive Finite-Amplitude Waves in Nondispersive Fluids", J. Acoust. Soc. Am. 50, 1299-1312 (1971).

APPENDIX A

Included in this appendix are descriptions of the packaged routines that are used in the program. The routines VODQ, ROMBS, and DEI are from the JPL Fortran V Subroutine Directory, Edition No. 4, October 1970. The routines TSCALE, TSETUP, and TPLOT are from the Westinghouse Research Laboratories Fortran library.

12.1. INITIAL VALUE PROBLEM

12.1.1. NUMERICAL SOLUTION, S.P. AND D.P.

12.1.1.1. IDENTIFICATION

NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS.

LANG	FILE	ELT/VERS	SIZE	ENTRY NAMES
F-V	LIB*JPLs	VOIGO/JPL	5257(8)=2735(10)	VODQ,VODQ1,VODQG

SUBROUTINE USED: #NOIE#

COGNIZANT PERSON: F. T. KROGH, SECTION 314, JPL, 1969 JUNE 10

12.1.1.2. PURPOSE

THIS SUBROUTINE COMPUTES THE NUMERICAL SOLUTION OF THE INITIAL VALUE PROBLEM FOR A SYSTEM OF ONE OR MORE ORDINARY DIFFERENTIAL EQUATIONS.

12.1.1.3. REFERENCE

1. F. T. KROGH, VODQ/SVDQ/DVDQ - VARIABLE ORDER INTEGRATORS FOR THE NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS. SECTION 314 SUBROUTINE WRITE-UP, JPL, MAY 1969.
2. F. T. KROGH, 'ON TESTING A SUBROUTINE FOR THE NUMERICAL INTEGRATION OF ORDINARY DIFFERENTIAL EQUATIONS', JPL, SECTION 314, TM NO. 217, MAY 1969.

12.1.1.4. PRECISION

THIS SUBROUTINE IS PRIMARILY SINGLE PRECISION, USING DOUBLE PRECISION ARITHMETIC ONLY AT A FEW CRITICAL POINTS. TWO OTHER VERSIONS OF THIS SUBROUTINE ARE ALSO AVAILABLE: SVDQ, WHICH IS ENTIRELY SINGLE PRECISION, AND DVDQ, WHICH IS ENTIRELY DOUBLE PRECISION. SEE THEIR WRITE-UPS FOR MINOR DIFFERENCES IN USAGE.

12.1.1.5. REMARKS

THE ORDINARY DIFFERENTIAL EQUATIONS MAY BE OF ORDERS 1,2,3, OR 4, AND NEED NOT ALL BE OF THE SAME ORDER. IT IS SUGGESTED THAT IF AN

ORDINARY DIFFERENTIAL EQUATION IS OF ORDER 2,3, OR 4, IT SHOULD BE TREATED AS SUCH, RATHER THAN BEING REFORMULATED AS A SET OF 2,3, OR 4 FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS. THIS PERMITS THE USE OF INTEGRATION FORMULAS WHICH HAVE BETTER NUMERICAL STABILITY.

12.1.1.6. METHOD

THIS SUBROUTINE USES LINEAR MULTISTEP PREDICTOR-CORRECTOR FORMULAS OF THE ADA-S-MOULTON TYPE WITH THE APPROPRIATELY GENERALIZED FORMULAS FOR ORDINARY DIFFERENTIAL EQUATIONS OF ORDERS 2,3, AND 4. THE SUBROUTINE SELECTS THE ORDER (1,2,..., OR 9) OF THE FORMULAS INDEPENDENTLY FOR EACH ORDINARY DIFFERENTIAL EQUATION OF THE SYSTEM AND THEN SELECTS A STEP SIZE TO BE USED FOR THE ENTIRE SYSTEM. THESE SELECTIONS OF ORDER AND STEPSIZE ARE RECONSIDERED AND POSSIBLY CHANGED AT EACH STEP. THE SELECTION IS BASED UPON MAINTAINING NUMERICAL STABILITY AND MEETING THE USER'S REQUESTED LOCAL ACCURACY.

THE SUBROUTINE PROVIDES SPECIAL RETURNS BASED UPON EITHER THE NUMBER OF STEPS, THE VALUE OF THE INDEPENDENT VARIABLE, VALUES OF THE DEPENDENT VARIABLES, OR VALUES OF AUXILIARY FUNCTIONS. ONLY THE FIRST TWO OF THESE FEATURES ARE DESCRIBED IN THIS WRITE-UP. SEE REFERENCE 1 FOR THE OTHER TWO FEATURES WHICH INVOLVE USE OF THE ENTRY POINT VODQG.

12.1.1.7. USAGE

INTEGER	NEQ,KD(#N1#),IFLAG,MXSTEP,KSTEP,KEMAX, KQ(#N7#)
REAL	EP(#N2#),HMINA,HMAXA,EMAX
REAL	F(#N3#),DT(10,#N4#)
DOUBLE PRECISION	T,Y(#N5#),H,DELT,TFINAL,YN(#N6#)

```
CALL VODQ (NEQ,T,Y,F,KD,EP,IFLAG,H,HMINA,HMAXA,DELT,TFINAL,
MXSTEP,KSTEP,KEMAX,EMAX,KQ,YN,DT)
GO TO 6
```

```
4 CALL VODQ1 6 GO TO (10,10,30,40,50,60,70,80),IFLAG
```

13 [COMPUTE F() AS A FUNCTION OF T AND Y(). IF IFLAG = 2 THE VALUE OF T WILL BE UNCHANGED FROM THE PREVIOUS RETURN. THUS THE USER CAN EFFECT SOME REDUCTION OF EXECUTION TIME BY EVALUATING ANY SUBEXPRESSIONS OF F() WHICH DEPEND ON T BUT NOT ON Y() ONLY WHEN IFLAG = 1, AND

REUSING THESE VALUES WITHOUT RECOMPUTING THEM WHEN
 IFLAG = 2.]
 GO TO 4

30 [AN OUTPUT POINT HAS BEEN REACHED UNDER CONTROL OF DELT.]
 GO TO 4

40 [T HAS REACHED THE VALUE TFINAL.]
 GO TO ...

50 [AN OUTPUT POINT HAS BEEN REACHED UNDER CONTROL OF
 MXSTEP.]
 GO TO 4

60 [ACCURACY APPARENTLY BEING LIMITED BY ROUND-OFF ERROR. SUG-
 GEST THAT EP() BE INCREASED.]
 GO TO ...

70 [ABS(H) NEEDS TO BE LESS THAN HMINA TO ACHIEVE THE
 ACCURACY SPECIFIED BY EP().]
 GO TO ...

80 [ERRONEOUS OR INCOMPATIBLE SETTING OF CONTROL PARAMETERS.]

THE DIMENSIONING PARAMETERS MUST SATISFY:

#N1# /.GE. 1 IF KD(1)>0,
 \.GE. NEG IF KD(1)<0

#N2# /.GE. 1 IF EP(1)>0,
 \.GE. #K# IF EP(1)<0, WHERE #K# IS THE SMALLEST INTEGER
 FOR WHICH EP(#K#) .GE. 0

#N3#, #N4#, #N7# .GE. NEG

#N5#, #N6# .GE. #KDTOTAL#

#KDTOTAL# /.EQ. KD(1)*NEG IF KD(1)>0
 \.EQ. SUM OF IABS(KD(I)) FOR I = 1,...,NEG IF
 KD(1)<0

BEFORE CALLING VODQ, THE USER MUST ASSIGN INITIAL VALUES TO

T AND Y() AND SET THE PARAMETERS NEQ,KD(),EP(),H,HMINA,HMAXA,DELT,TFINAL, AND MXSTEP. THE USER MAY CHANGE THE VALUES OF EP() WHENEVER IFLAG = 1, AND MAY CHANGE THE VALUES OF HMINA,HMAXA,DELT,TFINAL, AND MXSTEP AT ANY TIME.

THE USER MUST COMPUTE F() WHENEVER THE SUBROUTINE RETURNS WITH IFLAG = 1 OR 2.

THE SUBROUTINE WILL CHANGE THE VALUES OF T,Y(),IFLAG,H,KSTEP,KEMAX,EMAX,KG(),YN(), AND DT() DURING THE INTEGRATION. ON RETURNS WITH IFLAG = 3,4, OR 5 THE CURRENT VALUE OF THE SOLUTION WILL BE CONTAINED IN T,Y(), AND F().

THE SUBROUTINE PARAMETERS ARE DEFINED AS FOLLOWS:

NEQ NUMBER OF DIFFERENTIAL EQUATIONS IN THE SYSTEM.

T INDEPENDENT VARIABLE. MUST BE SET TO ITS INITIAL VALUE BY THE USER. THEREAFTER T WILL BE CHANGED BY THE SUBROUTINE.

(Y(I),I=1,#KDTOTAL#) DEPENDENT VARIABLES. FOR SYSTEM OF FIRST ORDER EQUATIONS Y(I) IS THE I**#TH# DEPENDENT VARIABLE. IN THE CASE OF HIGHER ORDER EQUATIONS, THE FIRST LOCATIONS OF Y CONTAIN THE DEPENDENT VARIABLE ASSOCIATED WITH THE FIRST EQUATION AND ITS DERIVATIVES UP TO ORDER ONE LESS THAN THE ORDER OF THE EQUATION (WITH THE LOWER ORDER DERIVATIVES STORED FIRST), THEN FOLLOW THE VARIABLES ASSOCIATED WITH THE SECOND EQUATION, ETC. FOR EXAMPLE, FOR THE SYSTEM:

$$\begin{aligned} U'' &= F_1(U,U',V,V',T) \\ V'' &= F_2(U,U',V,V',T) \end{aligned}$$

THE Y() ARRAY WOULD BE USED AS FOLLOWS:

$$\begin{aligned} Y(1) &= U \\ Y(2) &= U' \\ Y(3) &= V \\ Y(4) &= V' \end{aligned}$$

(F(I),I=1,NEQ) DERIVATIVE VALUES. IF #DI# DENOTES THE ORDER OF THE I**#TH# EQUATION, THEN F(I) IS THE #DI**#TH# DERIVATIVE OF THE I**#TH# COMPONENT WITH RESPECT TO T. THE USER MUST PROVIDE THE CODE TO COMPUTE F(), GIVEN T

AND Y().

KD() ORDER OF THE DIFFERENTIAL EQUATIONS IN SYSTEM. DIFFERENTIAL EQUATIONS WITH ORDER GREATER THAN 4 CAN ONLY BE INTEGRATED BY BREAKING THEM INTO LOWER ORDER EQUATIONS, OR BY CHANGING CERTAIN DATA STATEMENTS IN THE INTEGRATOR. IF ALL OF THE DIFFERENTIAL EQUATIONS ARE OF THE SAME ORDER THEN IT SUFFICES TO SET KD(1) EQUAL TO THAT ORDER (KD(1)>0). IF DIFFERENTIAL EQUATIONS OF MIXED ORDER ARE TO BE INTEGRATED, KD IS A VECTOR WITH KD(1)<0 TO INFORM THE INTEGRATOR THAT THIS IS THE CASE. THE ORDER OF THE I**NTHM EQUATION IS THEN GIVEN BY ABS(KD(I)). (FOR I>1, KD(I) MAY BE EITHER POSITIVE OR NEGATIVE).

EP() PARAMETER USED TO CONTROL THE LOCAL ERROR. IF ONE WANTS THE SAME LOCAL ERROR BOUND ON ALL COMPONENTS, THEN EP(1)>0 AND THE ESTIMATE OF THE LOCAL ERROR IN EACH COMPONENT IS KEPT LESS THAN EP(1)/10. FOR DIFFERENT ERROR BOUNDS ON DIFFERENT COMPONENTS, LET EP(1)<0 FOR I<NKH, AND LET EP(NKH) .GE. 0. THE LOCAL ERROR CONTROL FOR THE I**NTHM COMPONENT IS THEN BASED ON ABS(EP(I)) FOR I<NKH AND ON EP(NKH) FOR I .GE. NKH. SUGGESTIONS ON HOW TO SELECT EP() ARE GIVEN IN REFERENCE 1.

IFLAG PARAMETER USED FOR COMMUNICATION BETWEEN THE INTEGRATOR AND THE USER. THE INTEGRATOR SETS IFLAG AS FOLLOWS:

- =1 THE VALUE OF Y() FOR THE CURRENT STEP HAS BEEN PREDICTED. THE USER SHOULD COMPUTE F() AND CALL VODQ1. IF A RELATIVE ERROR TEST IS DESIRED, THE NEW VALUE(S) OF EP SHOULD ALSO BE COMPUTED HERE.
- =2 THE VALUE OF Y() FOR THE CURRENT STEP HAS BEEN CORRECTED. THE USER SHOULD COMPUTE F() AND CALL VODQ1.
- =3 AN OUTPUT POINT HAS BEEN REACHED (SEE USAGE OF DELT), TO CONTINUE THE INTEGRATION CALL VODQ1.
- =4 T=TFINAL. IF VODQ1 IS CALLED WITH T=TFINAL AND IFLAG=4, IFLAG IS SET EQUAL TO 8. IF THE VALUE OF TFINAL IS CHANGED THE INTEGRATION WILL CONTINUE.
- =5 KSTEP=KSOUT (SEE THE DESCRIPTION OF MXSTEP).
- =6 EMAX>0.1 AND IT APPEARS THAT REDUCING H WILL NOT HELP REDUCE THE GLOBAL ERROR BECAUSE OF ROUND-OFF ERROR. IF

THIS OCCURS, A LARGER VALUE OF $EP()$ (OR OF $ABS(EP(KEMAX))$ IF EP IS A VECTOR) SHOULD PROBABLY BE USED. IF EP IS NOT INCREASED, TOO SMALL A STEPSIZE IS LIABLE TO BE USED. WE HAVE FOUND THAT REPLACING $EP()$ WITH $32*EMAX*EP()$ WORKS REASONABLY WELL. (NOTE THAT $EMAX$ IN MOST CASES WILL BE ONLY SLIGHTLY LARGER THAN 0.1).

- =7 $ABS(H) < HMINA$. THIS MAY BE THE RESULT OF HALVING H , OF COMING TO THE END OF THE STARTING PHASE WITH $ABS(H) < HMINA$, OR OF THE USER INCREASING THE VALUE OF $HMINA$. IF ONE WISHES TO CONTINUE WITH THE CURRENT VALUE OF H , SET $HMINA .LE. H$ AND CALL $VODQ1$. IF THE STEPSIZE HAS JUST BEEN HALVED (IN WHICH CASE $EMAX > 0.1$), ONE MAY CONTINUE WITH THE OLD STEPSIZE BY SIMPLY CALLING $VODQ1$. (SUCH AN ACTION IS RISKY WITHOUT A CAREFUL ANALYSIS OF THE SITUATION.) IF THE STEPSIZE HAS NOT JUST BEEN HALVED, SIMPLY CALLING $VODQ1$ WILL RESULT IN THE INTEGRATION CONTINUING WITH THE CURRENT VALUE OF H , AND A RETURN TO THE USER WITH $IFLAG=7$ WILL OCCUR AT THE END OF EVERY STEP UNTIL $ABS(H) .GE. HMINA$.
- =8 ERROR INDICATION. THE CONDITIONS WHICH MAY CAUSE $IFLAG$ TO BE SET EQUAL TO 8 ARE LISTED BELOW.

WHEN CALLING $VODQ$:

$NEQ .LE. 0$
 $H*DELT .LE. 0$
 $KD=0$ OR $KD > 4$ (KD A SCALAR);
 $KD(NIH)=0$ OR $ABS(KD(NIH)) > 4$ FOR SOME $NIH, NIH=1,2,...$,
 NEQ (KD A VECTOR);

WHEN CALLING $VODQ1$:

$IFLAG=4$ AND $T=TFINAL$;
 $EP=0$ AND $HMAXA \neq 0$;
 $H*(TFINAL - (\text{INITIAL VALUE OF } T)) < 0$ AND THE ESTIMATED ERROR IN EXTRAPOLATING TO $TFINAL$ FROM THE INITIAL POINT (USING A FIRST ORDER METHOD) IS LARGER THAN PERMITTED BY EP ;

WHEN CALLING $VODQG$ WITH $NG \neq 0$:

$NSTOP < 0$ OR $NSTOP > ABS(NG)$ (SEE USAGE OF THE $VODQG$ ENTRY IN THE COMPLETE WRITE-UP)

IF VODQ1 IS CALLED WITH IFLAG SET EQUAL TO 8, THE PROGRAM IS STOPPED AND AN ERROR MESSAGE IS PRINTED.

H THE STEPSIZE. H CAN BE EITHER POSITIVE OR NEGATIVE. WHEN SELECTING THE INITIAL VALUE OF H, THE USER SHOULD REMEMBER THE FOLLOWING:

1. THE INTEGRATOR IS CAPABLE OF CHANGING H QUITE RAPIDLY AND THUS THE INITIAL CHOICE IS NOT CRITICAL.

2. IF IT DOES NOT LEAD TO PROBLEMS IN COMPUTING THE DERIVATIVES (E.G. BECAUSE OF OVERFLOW OR BECAUSE OF TRYING TO COMPUTE THE SQUARE ROOT OF A NEGATIVE NUMBER), IT IS BETTER TO CHOOSE H MUCH TOO LARGE THAN MUCH TOO SMALL.

HMINA MINIMUM STEPSIZE PERMITTED (AFTER THE INTEGRATION IS STARTED). AFTER THE INTEGRATION IS STARTED, AND WHEN H IS HALVED, $ABS(H)$ IS COMPARED WITH HMINA. IF $ABS(H) < HMINA$ CONTROL IS RETURNED TO THE USER WITH IFLAG=7.

HMAXA MAXIMUM STEPSIZE PERMITTED. THE STEPSIZE IS NOT DOUBLED IF DOING SO WOULD RESULT IN $ABS(H) > HMAXA$.

DELT OUTPUT INCREMENT. DELT MUST HAVE THE SAME SIGN AS H. INITIALLY TOUT IS SET EQUAL TO THE INITIAL VALUE OF T. WHENEVER $T = TOUT$ A RETURN IS MADE TO THE USER WITH IFLAG=3. (THUS, IN PARTICULAR, A RETURN IS ALWAYS MADE AT THE INITIAL POINT). WHENEVER VODQ1 IS CALLED WITH IFLAG=3, TOUT IS REPLACED WITH $T + DELT$. IF TOUT DOES NOT FALL ON AN INTEGRATION STEP, OUTPUT VALUES ARE OBTAINED BY INTERPOLATION ON THE FIRST STEP THAT $(T - TOUT) * H > 0$. INTERPOLATED VALUES FOR BOTH Y AND F ARE COMPUTED.

TFINAL FINAL VALUE OF T. WHEN T REACHES TFINAL, CONTROL IS RETURNED TO THE USER WITH IFLAG=4. OUTPUT VALUES ARE OBTAINED BY EXTRAPOLATION IF TFINAL DOES NOT FALL ON AN INTEGRATION STEP. IF ONE CHANGES TFINAL DURING THE INTEGRATION SO THAT $(TFINAL - T) * H < 0$, AN IMMEDIATE INTERPOLATION IS PERFORMED TO OBTAIN THE OUTPUT VALUES.

MXSTEP MAXIMUM NUMBER OF STEPS BETWEEN OUTPUT POINTS. WHEN

VODQ1 IS CALLED WITH 3 .LE. IFLAG .LE. 5, KSOUT IS SET EQUAL TO KSTEP + MXSTEP. AT THE END OF EACH STEP, KSTEP IS INCREMENTED AND COMPARED WITH KSOUT. IF KSTEP .GE. KSOUT CONTROL IS RETURNED TO THE USER WITH IFLAG=5. THUS IF DELT IS SUFFICIENTLY LARGE, CONTROL WILL BE RETURNED TO THE USER WITH IFLAG=5 EVERY MXSTEP STEPS.

KSTEP NUMBER OF INTEGRATION STEPS TAKEN.

KEMAX INDEX OF THE COMPONENT RESPONSIBLE FOR EMAX (SEE BELOW)

EMAX LARGEST VALUE IN ANY COMPONENT OF (ESTIMATED ERROR)/#EH WHERE #EH=ABS(EP) FOR THE COMPONENT UNDER CONSIDERATION. ORDINARILY THE STEPSIZE IS HALVED IF EMAX>0.1. HOWEVER, WITH A RECENT HISTORY OF ROUND-OFF ERROR LIMITING THE PRECISION, VALUES OF EMAX AS LARGE AS 1 ARE PERMITTED.

(KQ(I), I=1, NEQ) VECTOR USED TO STORE INTEGRATION ORDERS. KQ(I) GIVES THE INTEGRATION ORDER USED ON THE I**#TH# EQUATION

(YN(I), I=1, #KDTOTAL#) WORKING SPACE. (VECTOR USED TO STORE Y AT THE END OF EACH INTEGRATION STEP).

((DT(I, J), I=1, 10), J=1, NEQ) WORKING SPACE. (DIFFERENCE TABLES COMPUTED BY THE SUBROUTINE).

11.1. ONE-DIMENSIONAL

11.1.1. QUADRATURE, ONE-DIMENSION, S.P.

11.1.1.1. IDENTIFICATION

QUADRATURE, ONE-DIMENSIONAL, SINGLE PRECISION

LANG	FILE	BIT/VERS	SIZE	ENTRY NAMES
F-V	LIB*JPL\$	ROMBS/JPL	1251(8)=681(10)	ROMBS,ROM2

SUBROUTINES USED: NONE#

COGNIZANT PERSONS: W. R. BUNTON AND M. DIETHELM,
JPL, SECTION 314, 1969 SEPT 30

11.1.1.2. PURPOSE

OBTAIN APPROXIMATE EVALUATION OF A ONE-DIMENSIONAL DEFINITE
INTEGRAL BY NUMERICAL QUADRATURE.
$$ANS = \text{THE INTEGRAL FROM } A \text{ TO } B \text{ OF } F(X) \cdot DX$$

11.1.1.3. REFERENCES:

FOR A COMPLETE DESCRIPTION OF THIS SUBROUTINE, INCLUDING A
DISCUSSION OF A VARIETY OF TEST CASES, SEE:WILEY R. BUNTON, MICHAEL DIETHELM, AND KAREN HAIGLER, 'ROMBERG
QUADRATURE SUBROUTINE FOR SINGLE AND MULTIPLE INTEGRALS', JPLW. BUNTON, M. DIETHELM, G. WILDE, 'MODIFIED ROMBERG QUADRATURE: A
SUBROUTINE TO SUPPORT GENERAL SCIENTIFIC COMPUTING', JPL INTERNAL
MEMORANDUM TM 314-258, APRIL 1, 1970.W. BUNTON, M. DIETHELM, 'MODIFICATIONS TO THE JPL ROMBERG
SUBROUTINES', TM 314-247, 1 SEPT. 1970.

11.1.1.4. METHOD

THIS SUBROUTINE COMBINES TECHNIQUES FROM 'ROMBERG' AND 'ADAPTIVE
STEP' QUADRATURE METHODS. THE SUBROUTINE INITIALLY PICKS A
SUBINTERVAL $[A,B]$ OF THE TOTAL INTERVAL $[A,B]$ AND ATTEMPTS
TO APPROXIMATE THE INTEGRAL OVER THIS SUBINTERVAL BY USING THREE

STAGES OF ROMBERG QUADRATURE. THIS REQUIRES EVALUATION OF THE INTEGRAND AT FIVE EQUALLY SPACED POINTS.

IF THIS QUADRATURE IS REGARDED AS SUCCESSFUL (SEE: ERROR CONTROL) THE SUBROUTINE WILL ADD THE VALUE OBTAINED TO A RUNNING SUM AND PROCEED TO TREAT A NEW DISJOINT SUBINTERVAL OF THE SAME OR GREATER LENGTH.

IF THIS QUADRATURE IS NOT REGARDED AS SUCCESSFUL AND THE CURRENT STEP LENGTH IS GREATER THAN 10^{-10} THEN THE SUBROUTINE WILL REJECT THE RESULT FOR THE CURRENT SUBINTERVAL AND TAKE THE LEFT HALF OF THE SUBINTERVAL AS THE NEW SUBINTERVAL TO BE TREATED.

IF THE CURRENT STEP LENGTH IS 10^{-10} OR SMALLER, THEN THE SUBROUTINE WILL ACCEPT THE CURRENT RESULT AND PROCEED TO THE NEXT SUBINTERVAL, WRITING A MESSAGE ON FORTRAN UNIT 6 IDENTIFYING THE SUBINTERVAL ON WHICH THE ACCURACY TEST WAS NOT SATISFIED.

11.1.1.5. ERROR CONTROL

THE QUADRATURE IS REGARDED AS SATISFACTORY OVER A PARTICULAR SUBINTERVAL IF

1. THE ESTIMATED RELATIVE ERROR OF THE QUADRATURE OVER THAT SUBINTERVAL IS AT MOST $ERMAX$, OR
2. THE ESTIMATED ERROR IN THAT SUBINTERVAL RELATIVE TO THE ACCUMULATED VALUE OF THE INTEGRAL UP TO AND INCLUDING THAT SUBINTERVAL IS AT MOST $ERMAX$, OR
3. THE ESTIMATED ABSOLUTE ERROR OVER THAT SUBINTERVAL IS AT MOST $.1 * ABS(ERMAX)$.

11.1.1.6. PARAMETER CHECKING

THE SUBROUTINE WILL TERMINATE EXECUTION WITH A PRINTED MESSAGE AND A STANDARD EXEC 8 WALK-BACK (RETURN C) IF THE GIVEN PARAMETERS DO NOT SATISFY

$A < B$, OR $A > B$, BUT NOT $A = B$
 $0. < HMIN, LE, HSTAR, LE, HMAX$ AND

RELATIVE = $0. < ERMAX < 1.0$, BUT NOT $ERMAX = 0.$
 ABSOLUTE = ANY NEGATIVE NUMBER, BUT NOT $= 0.$

IF $A > B$ OR IF $HSTAR \geq (B-A)/4$, ONLY A WARNING MESSAGE IS PRINTED.

**COPY AVAILABLE TO DDC DOES NOT
 PERMIT FULLY LEGIBLE PRODUCTION**

11.1.1.7. USAGE

REAL A, B, X, FOFX, HSTAR, HMIN, HMAX, ERMAL, ANS
 INTEGER K, KEY

[ASSIGN VALUES TO A, B, HSTAR, HMIN, HMAX, ERMAL, AND KEY]

CALL ROMBS(A,B,X,FOFX,HSSTAR,HMIN,HMAX,ERMAL,ANS,K,KEY)

10 [EVALUATE THE INTEGRAND USING THE CURRENT VALUE OF X AND
 STORE THE RESULT IN FOFX.]

CALL ROM2

IF(K.EQ.1) GO TO 10

[QUADRATURE IS COMPLETED. RESULT IS IN ANS]

THE SUBROUTINE PARAMETERS ARE DEFINED AS FOLLOWS:

A,B LIMITS OF INTEGRATION. REQUIRE A NOT=B.

X VARIABLE SET BY THE SUBROUTINE FOR INTEGRAND EVALUATION
 IN THE USER'S PROGRAM.

FOFX VALUE OF INTEGRAND COMPUTED BY USER'S PROGRAM USING THE
 ARGUMENT X.

HSSTAR SUGGESTED INITIAL STEP SIZE. THE INITIAL STEP SIZE, H,
 WILL BE SET AT $H=.01*(B-A)$ IF $HSSTAR \geq (B-A)/4$. OR AT
 $H=HSSTAR$ IF $HSSTAR < (B-A)/4$. THE FIRST SUBINTERVAL WILL
 BE OF LENGTH $4.*H$ AND WILL REQUIRE FIVE EVALUATIONS OF
 THE INTEGRAND AT $X=A, A+H, \dots, A+4*H$. SUGGEST
 $HSSTAR \geq (B-A)/4$.

REQUIRE HMIN .LE. HSTAR .LE. HMAX.

HMIN MINIMUM ALLOWABLE STEP SIZE. REQUIRE $0. < HMIN \leq$
 HSTAR

HMAX MAXIMUM ALLOWABLE STEP SIZE. REQUIRE HSTAR .LE. HMAX.

ERMAL TOLERANCE ON RELATIVE OF ABSOLUTE ERROR. SEE DISCUSSION
 ABOVE UNDER 'ERROR CONTROL'. REASONABLE SETTINGS FOR
 ERMAL WOULD BE IN THE RANGE FROM $1.E-4$ TO $1.E-7$. IF
 GREATER ACCURACY IS REQUIRED, SEE THE WRITE-UP ON ROMBS.
 IT IS REQUIRED THAT $0. < ERMAL < 1$. FOR THE RELATIVE
 ERROR TEST. ONE SHOULD KNOW THE RANGE OF THE ANSWER

BEFORE HE USES ABSOLUTE ERROR.

- ANS THE FINAL VALUE OF THE INTEGRAL. AVAILABLE WHEN ROM2 RETURNS WITH K=2.
- K BRANCHING FLAG SET BY THE SUBROUTINE FOR USE IN THE USER'S PROGRAM. K=1 MEANS THE USER SHOULD EVALUATE THE INTEGRAND AT X, STORE THE VALUE IN FOEX, AND RE-ENTER ROM2. K=2 MEANS THE COMPUTATION IS COMPLETED AND THE VALUE IS IN ANS.
- KEY FLAG TO CONTROL PRINTING OF ERROR MESSAGES. PREVIOUSLY, ANY VALUE OF KEY NOT = 7 WOULD WRITE A DIAGNOSTIC MESSAGE WHEN H BECAME LT.HMIN. THE INPUT VALUES HAVE BEEN CHANGED SO THAT WHEN

ACTION

KEY=5 PRINT INTERMEDIATE T AND Y VALUES; PRINT THE HMIN DIAGNOSTIC IF DETECTED.

=6 PRINT INTERMEDIATE T AND Y VALUES; DO NOT PRINT THE HMIN DIAGNOSTIC.

=7 DO NOT PRINT THE T AND Y VALUES OR HMIN DIAGNOSTIC.

=ANY OTHER VALUE PRINT THE HMIN DIAGNOSTIC IF DETECTED.

THE T VALUES PRINTED ARE T(1,0), T(1,1), AND T(2,0). THE PRINTED Y VALUES ARE THE FUNCTIONAL EVALUATIONS Y(1) THRU Y(5) AT THE POINTS X,X+H,...,X+4H. SEE REFERENCES.

11.1.1.8. REMARKS:

THE FOLLOWING NOTES MAY BE APPLICABLE FOR DIFFICULT INTEGRANDS.

1. IF IN DOUBT, ONE SHOULD USE A SMALL VALUE OF HSTAR. THE STEP SIZE H CAN DOUBLE QUICKLY AND THE USER IS PENALIZED ONLY A SMALL NUMBER ON FUNCTIONAL EVALUATIONS WHILE HE INCREASES HIS CHANCES OF GETTING AN ACCURATE APPROXIMATION MANY FOLD.
2. BE CAUTIOUS WHEN RELATIVE ERMAX IS .GT. 10^{-5} . IF HSTAR.LT.(B-A)/4, BUT NOT SMALL ENOUGH, AND THE FUNCTION IS OSCILLATORY, VERY DIFFICULT, ETC., ROMDS CAN RETURN A WRONG ANSWER. EXAMPLE: $F(X)=X*\sin 30X*\cos X$ ON THE INTERVAL (0,2PI), HSTAR=1.57, TRUE ANSWER=-.20967248, A RELATIVE TOLERANCE OF .1 GAVE -.25E-5, AND A RELATIVE TOLERANCE OF .01 GAVE 4.188; BAD ANSWERS.

3. ALSO FOR RELATIVE $ERMAX \leq 10^{-6}$. IF THE RELATIVE ERROR IS ASKING FOR GREATER THAN 6 SIGNIFICANT DIGITS ONE IS PUSHING THE ACCURACY OF THE UNIVAC 1108. ON A HIGHLY OSCILLATORY, VERY DIFFICULT PROBLEM, ROMBS MAY BE TAKING THOUSANDS MORE FUNCTIONAL EVALUATIONS AND NOT REACHING THE ACCURACY IT DID AT 10^{-6} . EXAMPLE: SAME $F(x)$ AS ABOVE WHEN GIVEN TOLERANCE WAS 10^{-6} , $ANS = .20967380$ WITH A 2000 FUNCTIONAL EVALUATIONS; BUT WHEN WHEN $ERMAX = 10^{-8}$, $ANS = .20967765$ AND 10,267 FUNCTIONAL EVALUATIONS.
4. IF ONE WANTS A ROUGH APPROXIMATION OF THE INTEGRAL, HE CAN SET $HMIN = HSTAR = HMAX$. A FIXED STEP INTEGRATION OF THE FUNCTION WILL TAKE PLACE. BE SURE THAT $ILEV = 7$, OR MANY DIAGNOSTICS MAY BE PRINTED.

4.2.13. EXPONENTIAL INTEGRAL

4.2.13.1. IDENTIFICATION

EXPONENTIAL INTEGRAL

LANG	FILE	ELT/VERS	SIZE	ENTRY NAMES
F-V	LIB*JPLs	DEI/JPL		DEI

SUBROUTINES USED: DEXP,DLOG

COGNIZANT PERSON: E.W. NG, JPL, SECTION 314, 1970, SEPT. 14

4.2.13.2. PURPOSE

COMPUTE THE EXPONENTIAL INTEGRAL IN DOUBLE PRECISION ARITHMETIC. FOR X .GT. 0, THE EXPONENTIAL INTEGRAL, EI, IS DEFINED AS

$$EI(X) = \text{INTEGRAL FROM } T=-\text{INFINITY TO } T=X \text{ OF } (\exp(T)/T)*DT$$

WHERE THE INTEGRAL IS TO BE INTERPRETED AS THE CAUCHY PRINCIPAL VALUE. FOR X .LT. 0, $EI(X) = -EI(-X)$, WHERE

$$EI(Z) = \text{INTEGRAL FROM } T=Z \text{ TO } T=\text{INFINITY OF } (\exp(-T)/T)*DT$$

4.2.13.3. METHOD

THIS SUBROUTINE COMPUTES THE EXPONENTIAL INTEGRAL BY CHEBYSHEV RATIONAL APPROXIMATIONS FROM W.J. CODY AND H.C. THACHER, JR., MATH. COMP. VOL. 22, PP. 641-650, AND VOL. 23, PP. 289-303.

4.2.13.4. ACCURACY

EXTENSIVE TESTS WERE PERFORMED ON THE UNIVAC 1108 AND THE FOLLOWING ACCURACY STATISTICS WERE FOUND:

INTERVAL OF X	MAXIMUM RELATIVE	RMS RELATIVE
(-150, -4)	2.2D-16	5.1D-17

(-4, -1)	7.5D-17	1.2D-17
(-1, 0)	8.7D-18	1.4D-18
(0, 0.5)	1.8D-16*	3.2D-17*
(0.5, 6)	5.5D-17	1.0D-17
(6, 12)	1.6D-17	3.0D-18
(12, 24)	2.9D-17	7.1D-18
(24, 100)	8.9D-17	1.9D-17

CF. E.W. NG, COMM. ACM VOL. 13, #7, PP. 448-449.

4.2.13.5. USAGE

DOUBLE PRECISION DEI, X

USE THE FOLLOWING FUNCTION IN A FORTRAN ARITHMETIC STATEMENT:

DEI(X)

WHERE DEI IS THE COMPUTED VALUE OF THE EXPONENTIAL INTEGRAL.

4.2.13.6. LIMITATIONS

CI(0)=-INFINITY IS APPROXIMATED BY -7.2D75 AND EI(X).GT.
174.673) IS APPROXIMATED BY +7.2D75.

THESE LIMITATIONS WERE ORIGINALLY IMPOSED FOR THE IBM/360 AND HAVE
NOT BEEN MODIFIED FOR THE UNIVAC 1108.

4.2.13.7. ERROR EXITS

NONE

TSCALE

Specifications:

```
SUBROUTINE TSCALE (V,NPTS,VLOW,VHIGH,FIRST,IXORY)
INTEGER NPTS,IXORY
REAL V(NPTS),VLOW,VHIGH
LOGICAL FIRST
```

Purpose:

To scan the array V to obtain the minimum and maximum values, which are then used to establish a range which produces a rational scale for the axis.

Usage:

The subroutine should be called as

```
TSCALE(V,NPTS,VLOW,VHIGH,FIRST,IXORY)
```

where

V = array containing data to be scanned

NPTS = number of points in V to be scanned

VLOW,HIGH = receives upper and lower values for the range of V

FIRST = .TRUE. for first call for each plot,

.FALSE. for subsequent calls which may be made

for a multiple curve plot. Note that all curves

should be scanned via TSCALE for a multiple curve

plot.

IXORY = 0 for abscissa, 1 for ordinate

TSCALE (Cont'd)

Note:

The abscissa is eight grids down the page, and the ordinate is 5 grids across. TSCALE need not be used if the range of values is known.

TSETUP

Specifications:

```
SUBROUTINE TSETUP (XLOW,XHIGH,YLOW,YHIGH,IEORF,IGRID,LABELX,LABELY)
INTEGER IEORF,IGRID,LABELX(1),LABELY(1)
REAL XLOW,XHIGH,YLOW,YHIGH
```

Purpose:

To set up work area before data values to be plotted are stored (via TPLOT). Labelling information, ranges of values for the abscissa and ordinate variables, and options for grid and annotation are input parameters.

Usage:

The subroutine should be called as

```
TSETUP(XLOW,XHIGH,YLOW,YHIGH,IEORF,IGRID,LABELX,LABELY)
```

where

XLOW,XHIGH = lower and upper values for the range of
(YLOW,YHIGH)

the abscissa (ordinate) variable

IEORF = option to set the format for annotating the
axes; supply 1HE for E-format (1PE10.3), or
1HF for F-format (F10.3).

IGRID = option to select grid; supply 1HG for grid,
1Hb (blank) for no grid.

TSETUP (Cont'd)

LABELX(LABELY) = array containing the information to label the X(Y) axis. LABELX(1) (LABELY(1)) contains the number of characters, and the labelling data itself is stored six characters per word starting in LABELX(2) (LABELY(2)), with a maximum of 30 (24) characters.

TPLOT

Specifications:

```
SUBROUTINE TPLOT(PLOT, ICHAR, NPTS, XV, YV)
INTEGER ICHAR, NPTS
REAL XV(NPTS), YV(NPTS)
LOGICAL PLOT
```

Purpose:

To store plot data and initiate plotting via PLOT control.

Usage:

The subroutine should be called as

```
TPLOT (PLOT, ICHAR, NPTS, XV, YV)
```

where

PLOT = .FALSE. if data values are to be stored and
plotting deferred (for multiple curves on one plot);
.TRUE. if data values are to be stored and plotted.

ICHAR = single character used as plot symbol (e.g., 1H*)

NPTS = number of data points to be plotted

XV, YV = arrays containing the coordinate values to be
plotted, i.e., (XV(I)), YV(I)) is the Ith data point.

Restrictions:

1. If more than one data point occupies the same plot position, the most recent value processed will be used.
2. Data values outside range specified by TSETUP will not be plotted, but will be listed below the plot.

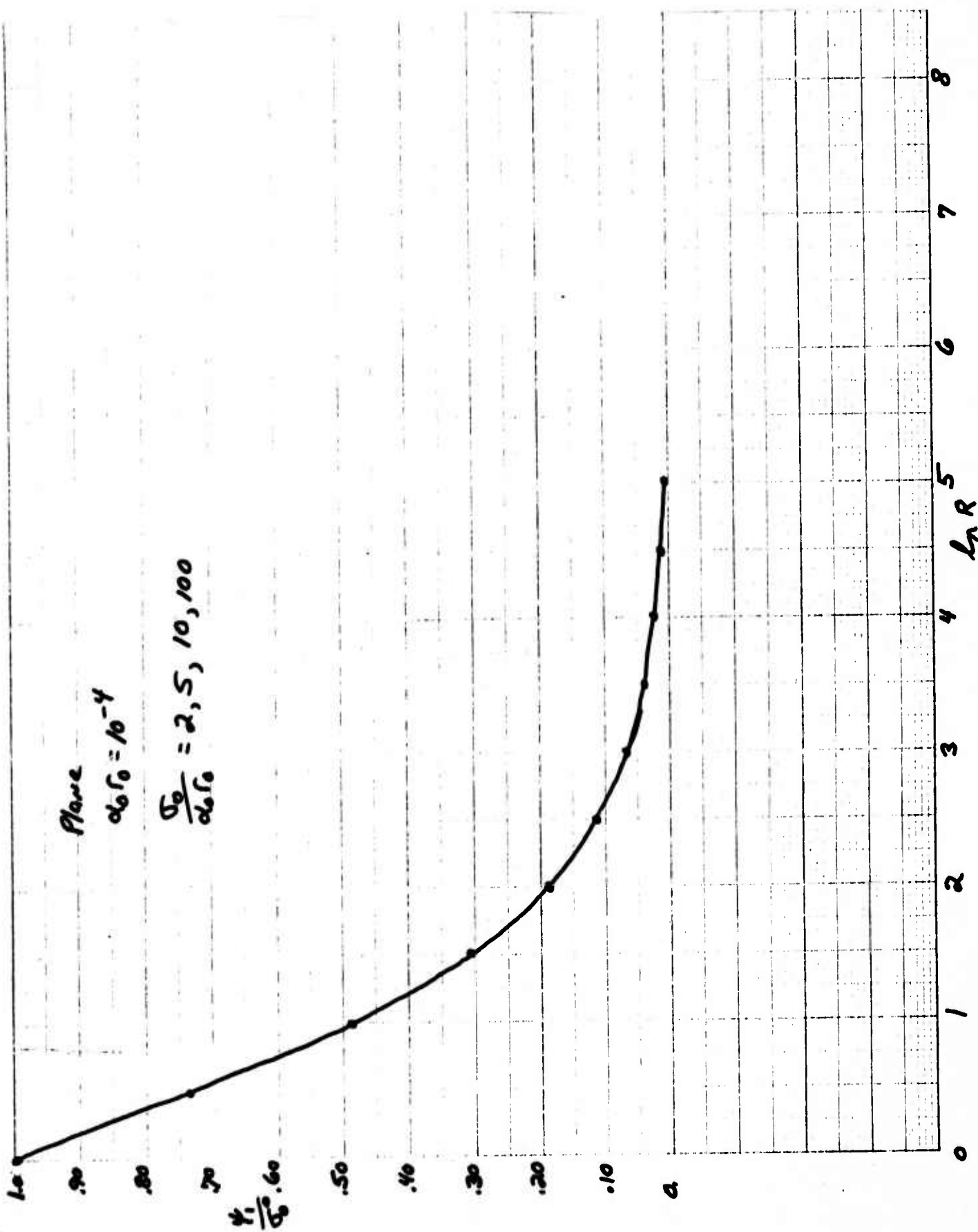
Figure 1
(pages F2 to F4)

Plane wave case

$$\alpha_o r_o = 10^{-4}$$

$$\sigma_o / \alpha_o r_o = 2, 5, 10, 100$$

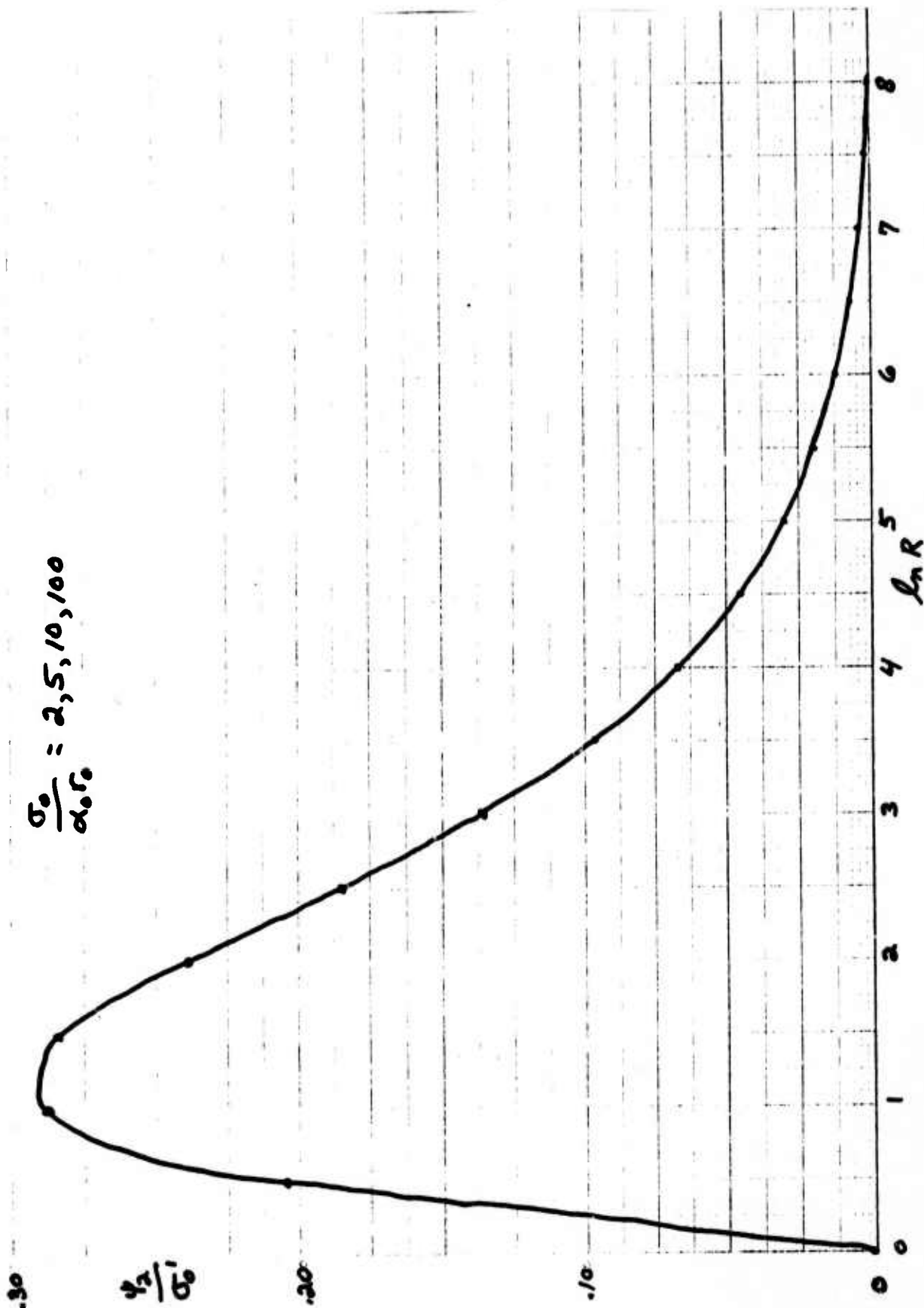
$$\psi_1, \psi_2, \psi_3$$



Plane

$$\alpha_0 r_0 = 10^{-4}$$

$$\frac{\sigma_0}{\alpha_0 r_0} = 2, 5, 10, 100$$



Plane
 $\alpha_0 r_0 = 10^{-4}$

$\frac{\sigma_0}{\alpha_0 r_0} = 2, 5, 10, 100$

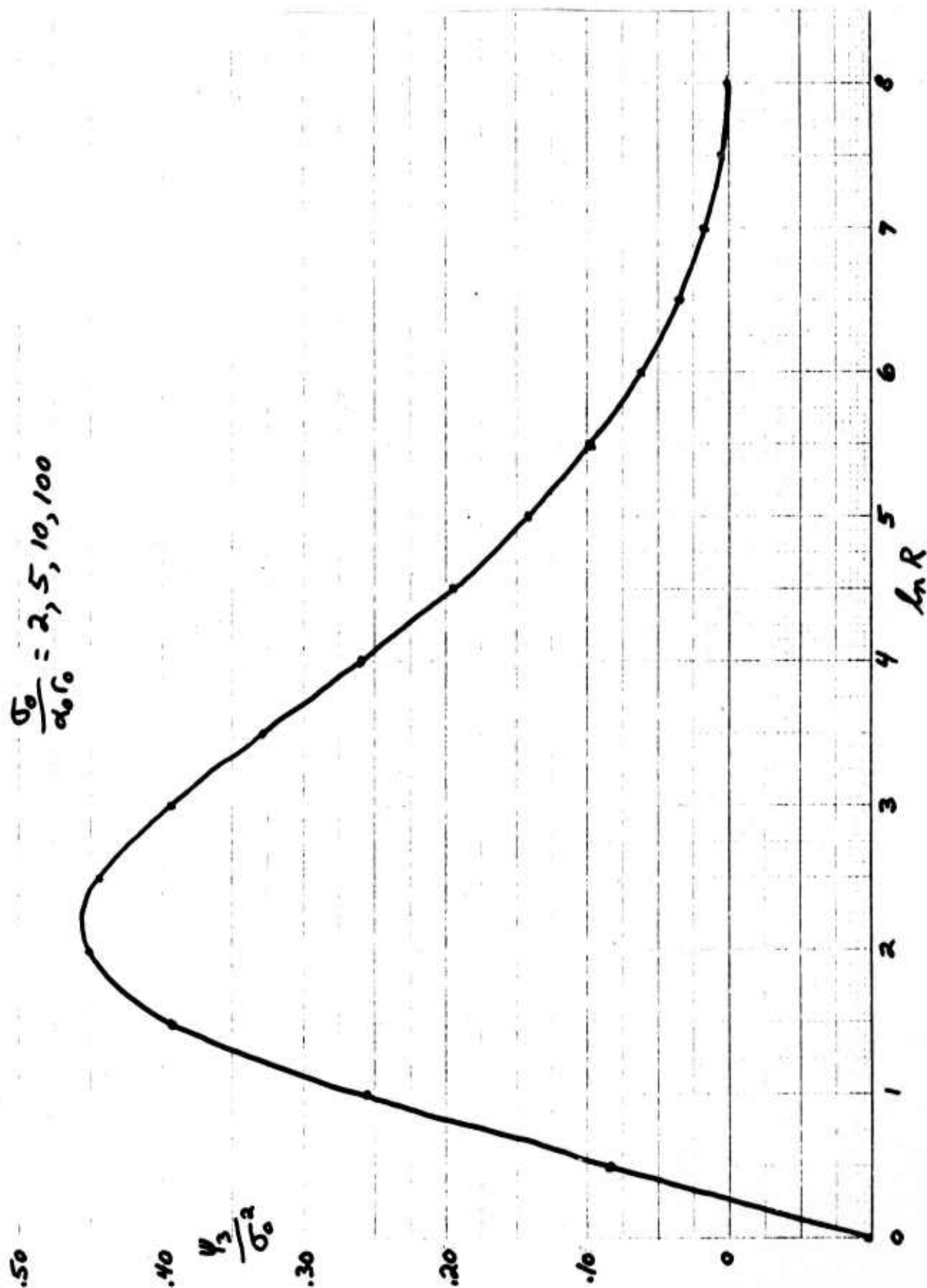


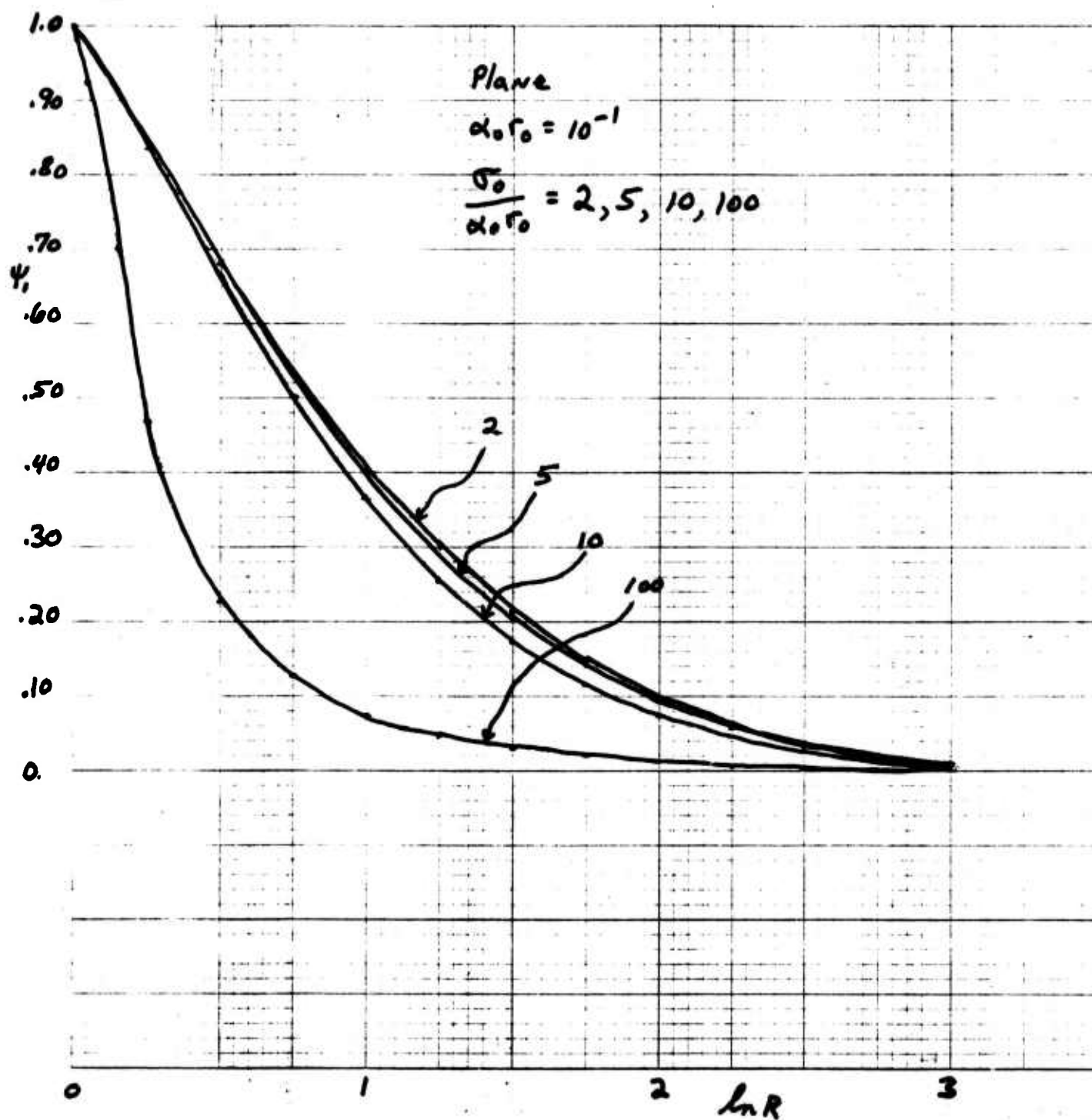
Figure 2
(pages F6 to F8)

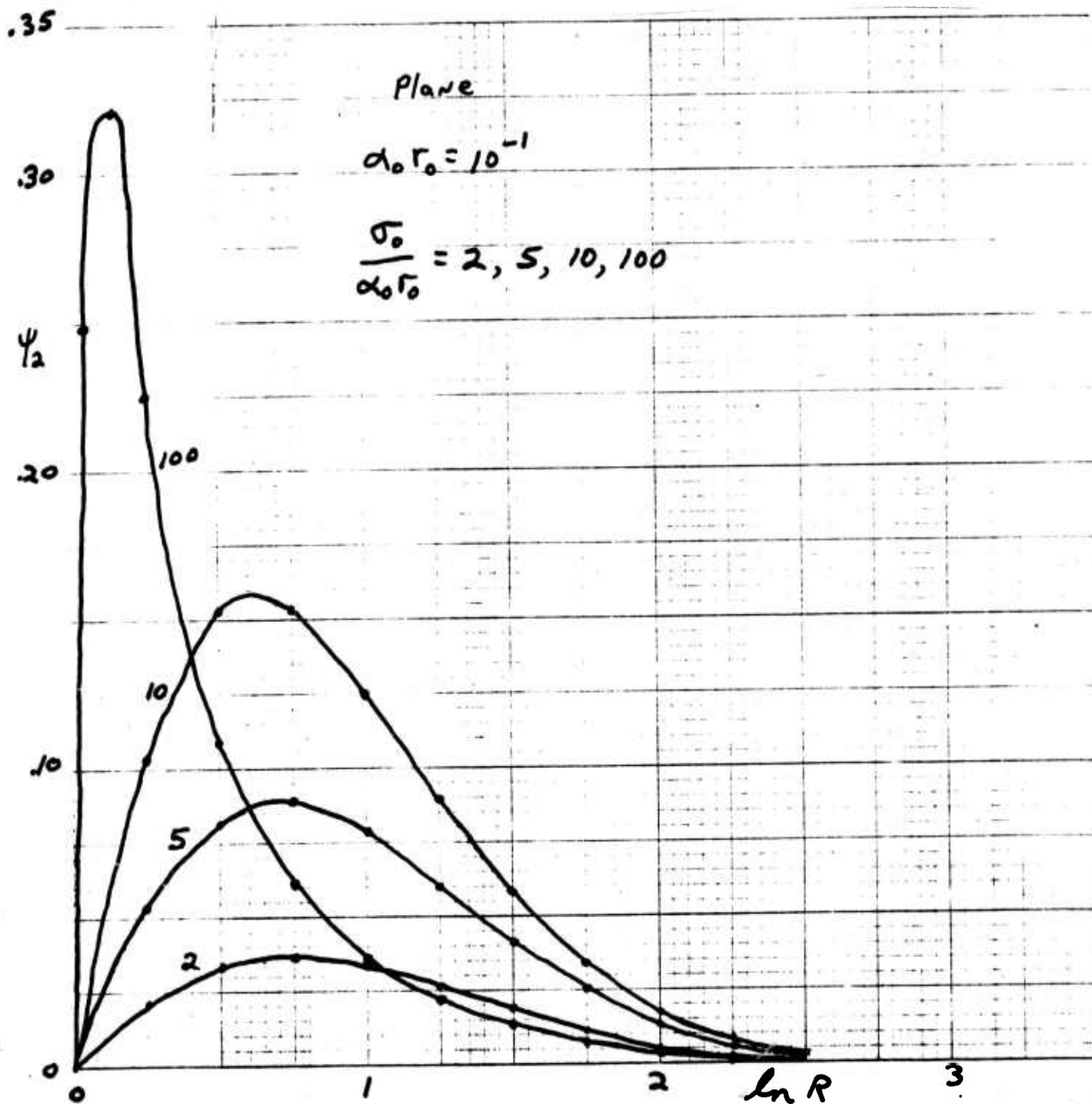
Plane wave case

$$\alpha_o r_o = 10^{-1}$$

$$\sigma_o / \alpha_o r_o = 2, 5, 10, 100$$

$$\psi_1, \psi_2, \psi_3$$





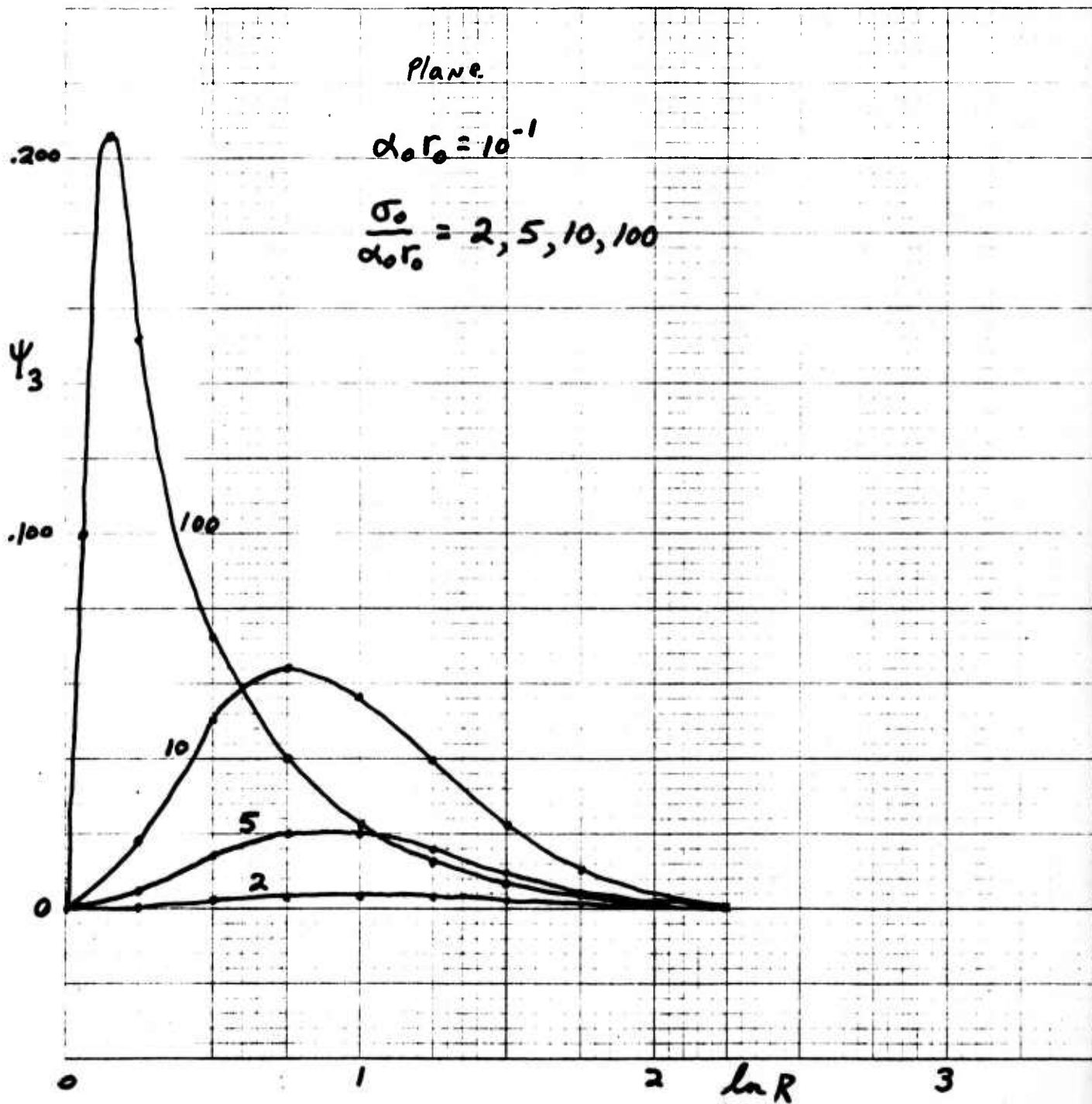


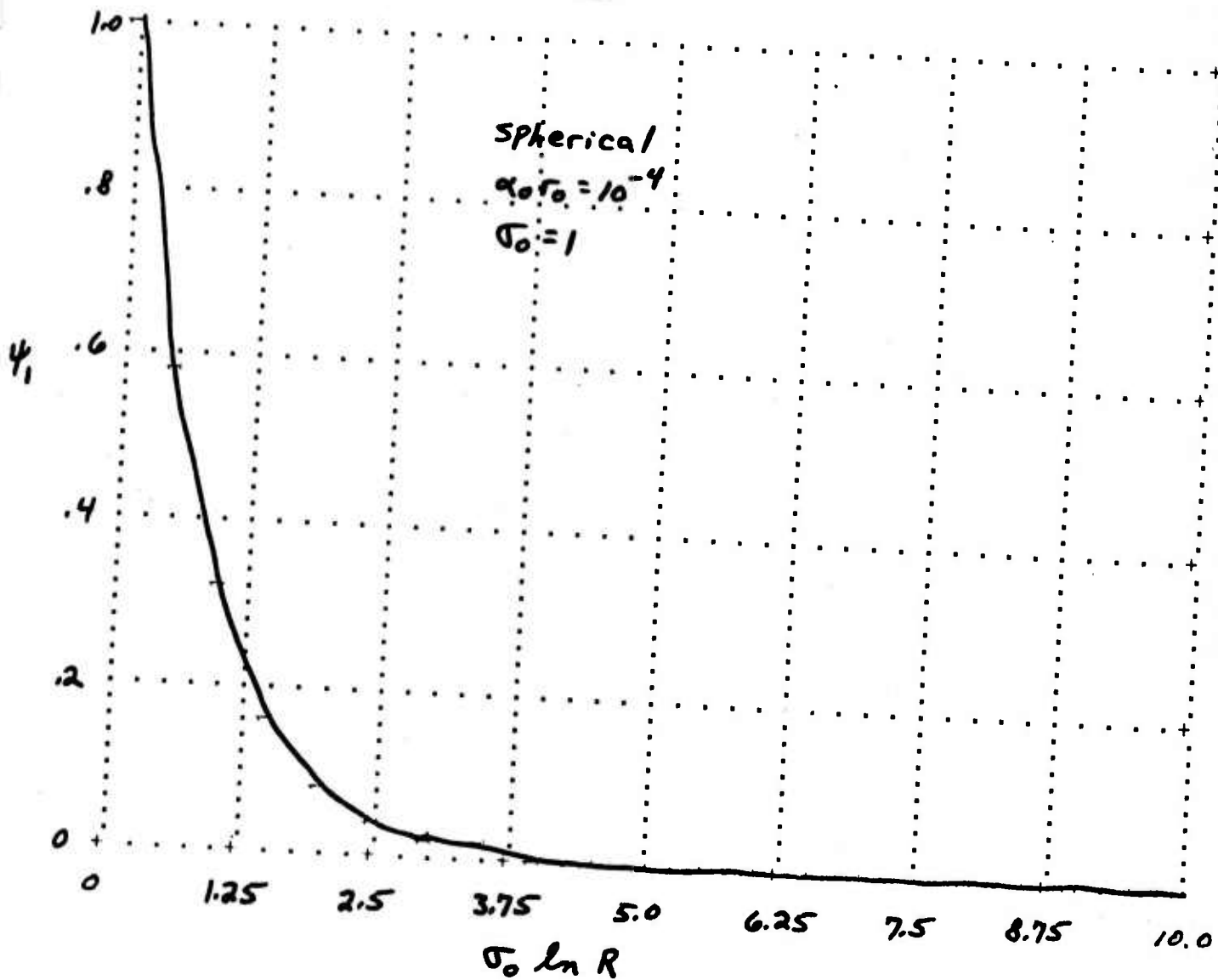
Figure 3
(pages F10to F12)

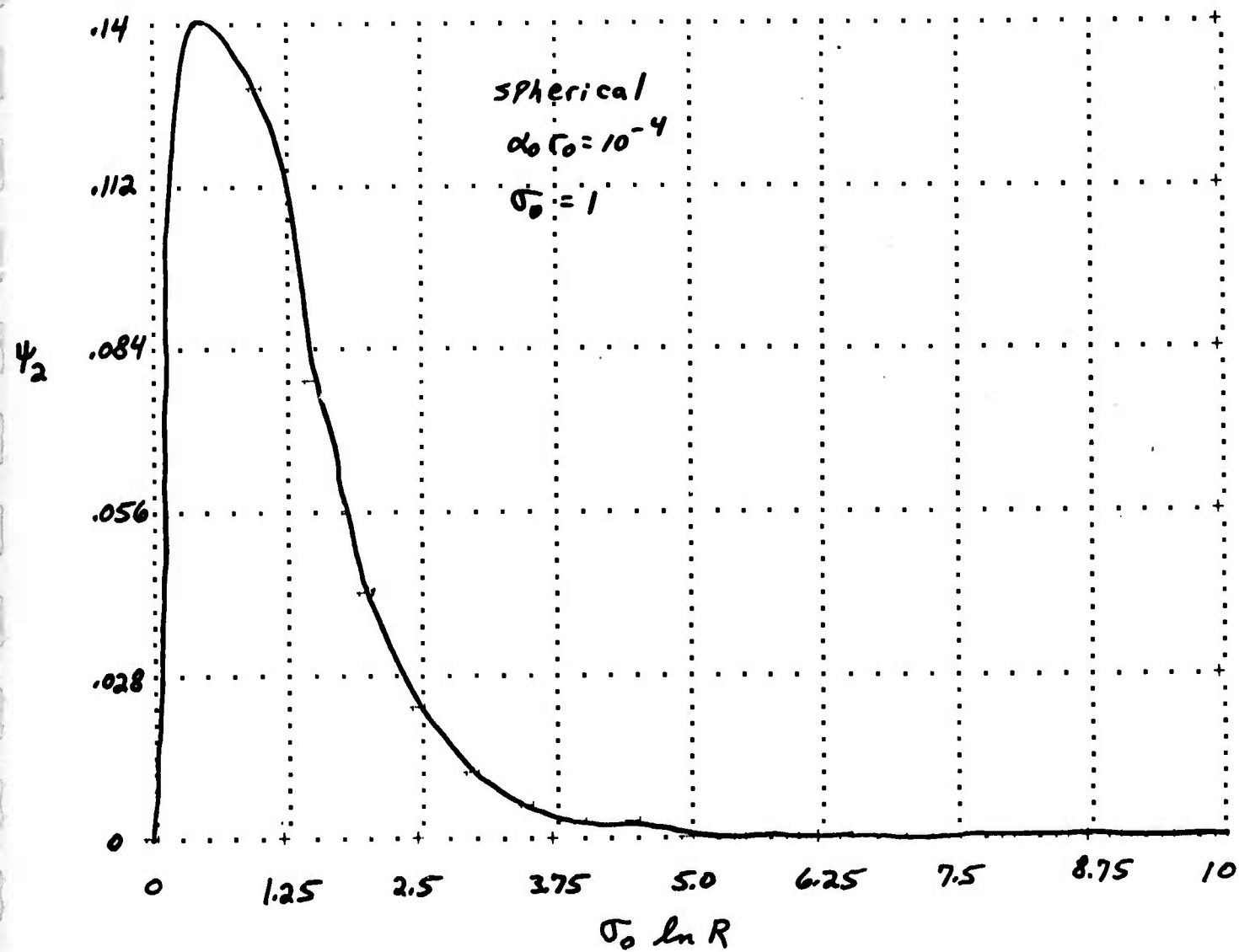
Spherical wave case

$$\alpha_o r_o = 10^{-4}$$

$$\sigma_o = 1$$

$$\psi_1, \psi_2, \text{EXDB}$$





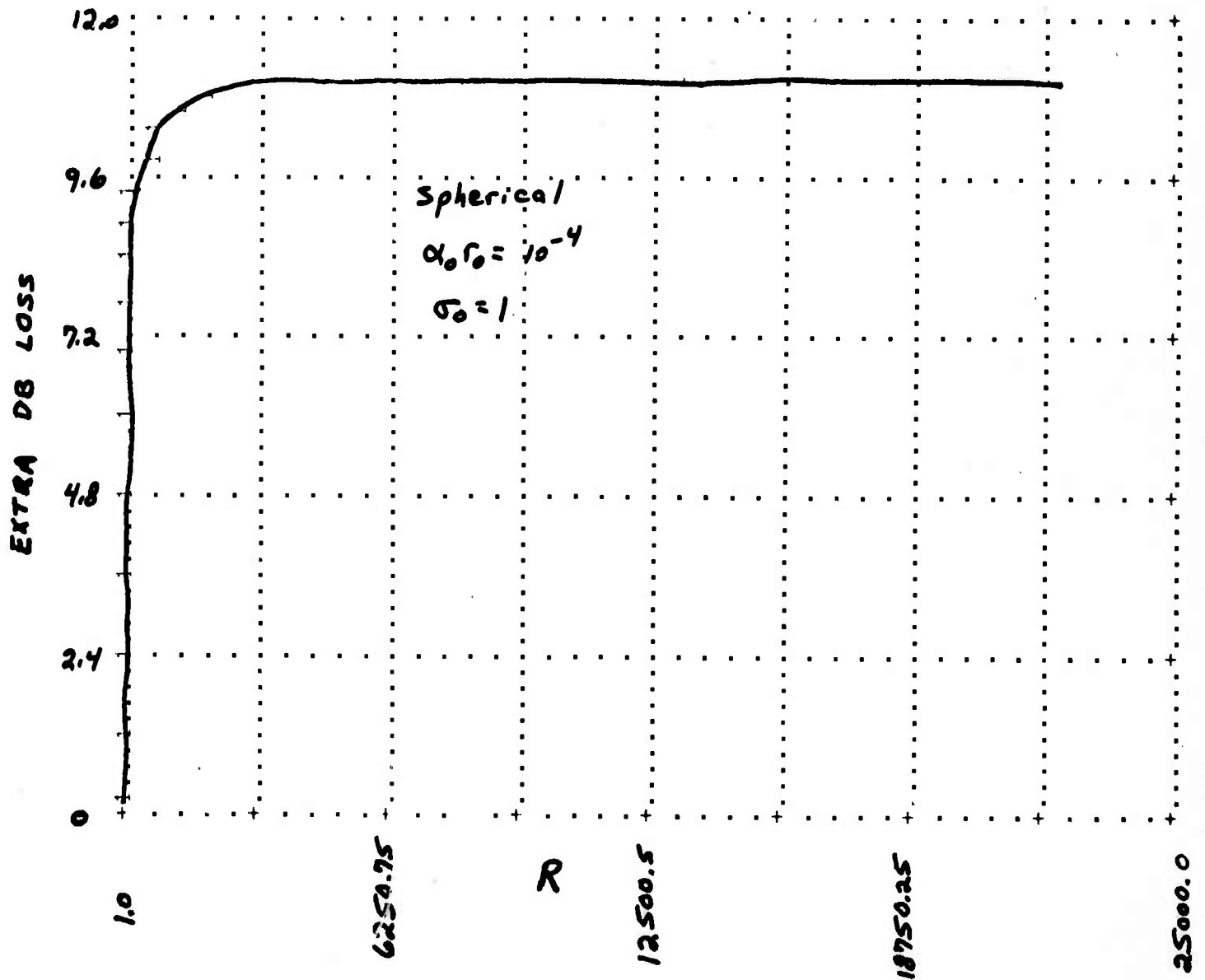


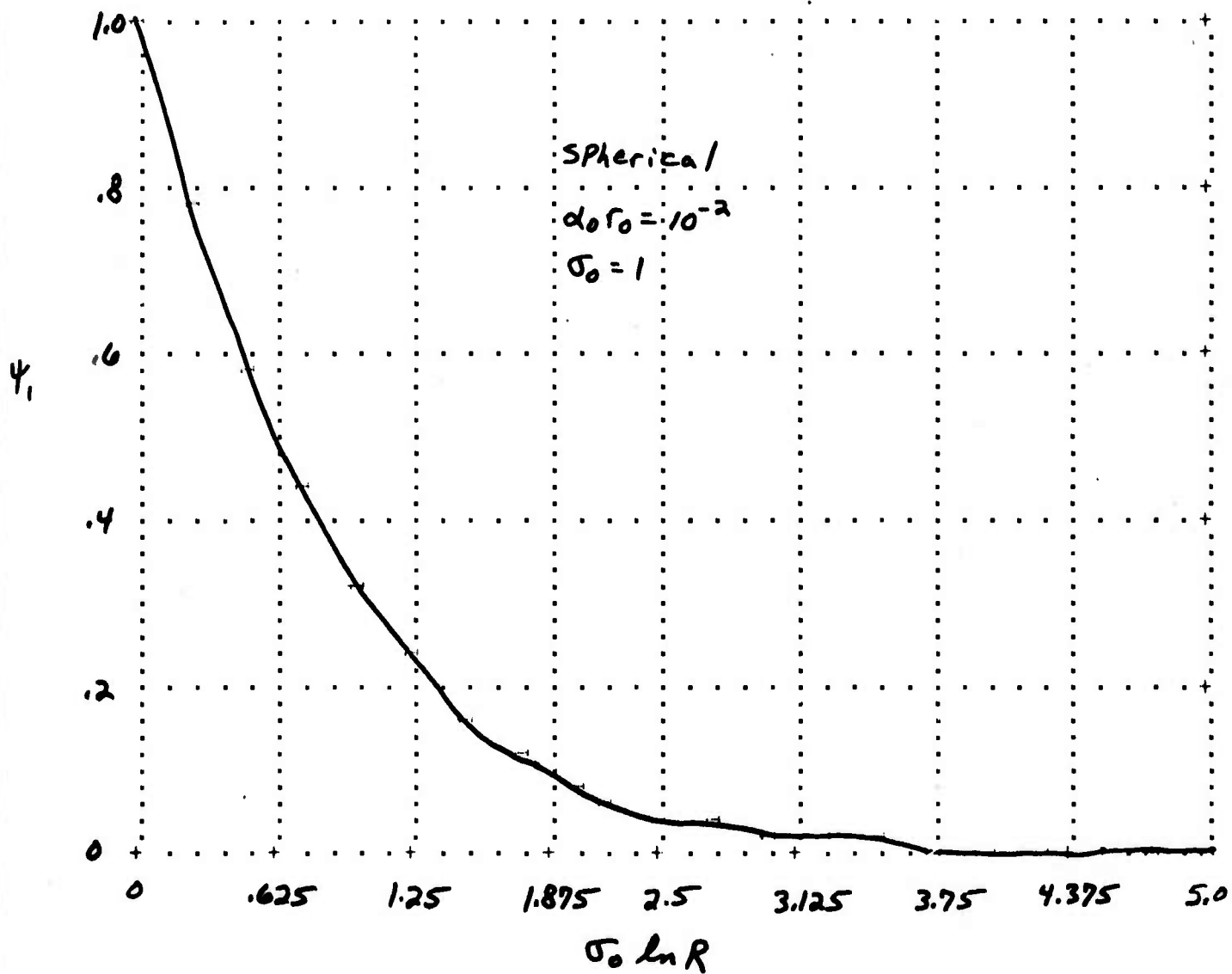
Figure 4
(pages F14to F16)

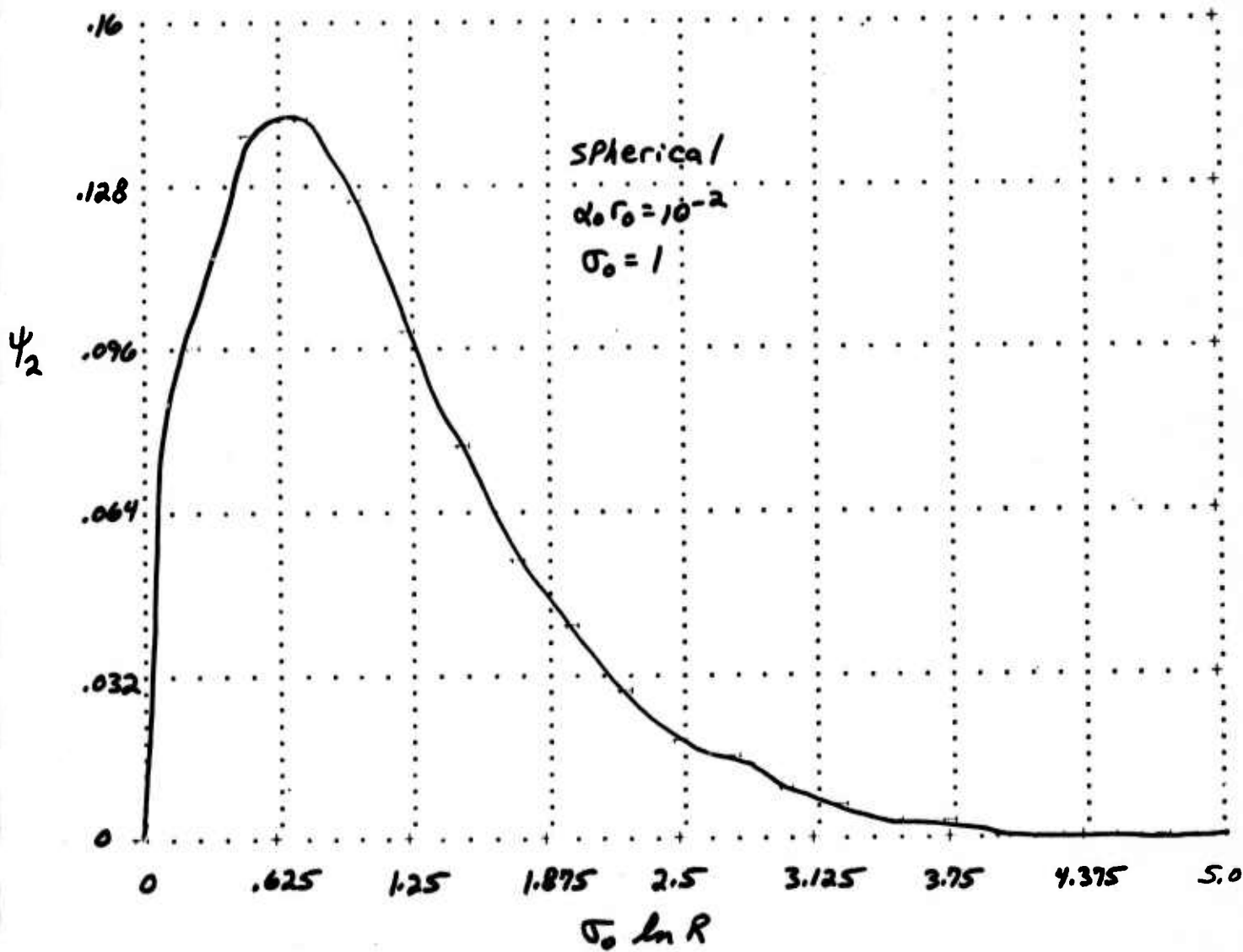
Spherical wave case

$$\alpha_o r_o = 10^{-2}$$

$$\sigma_o = 1$$

$$\psi_1, \psi_2, \text{ EXDB}$$





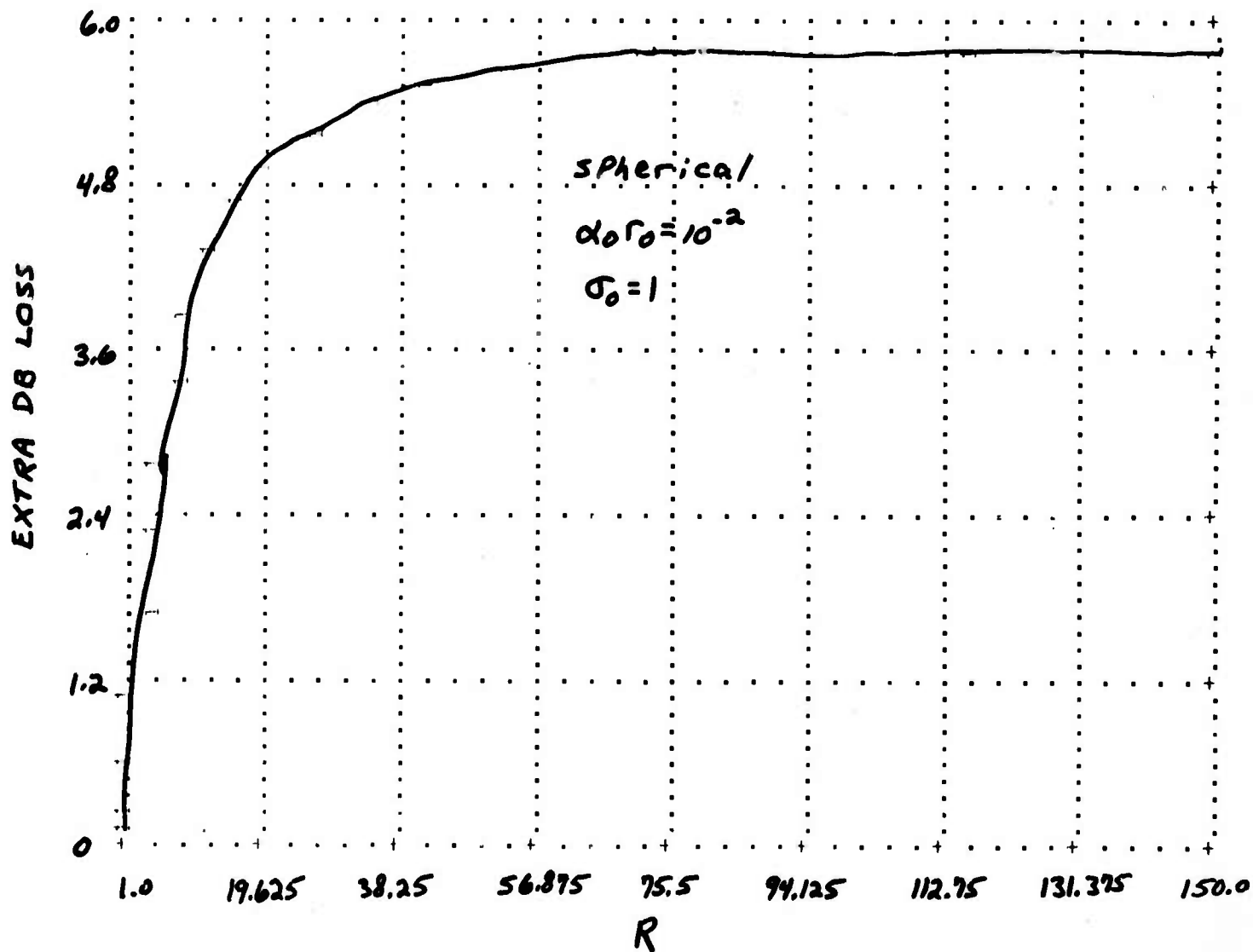


Figure 5
(pages F18to F22)

Mixed case

$$\sigma_0 = 10^{-4}$$

$$\sigma_0 = 1$$

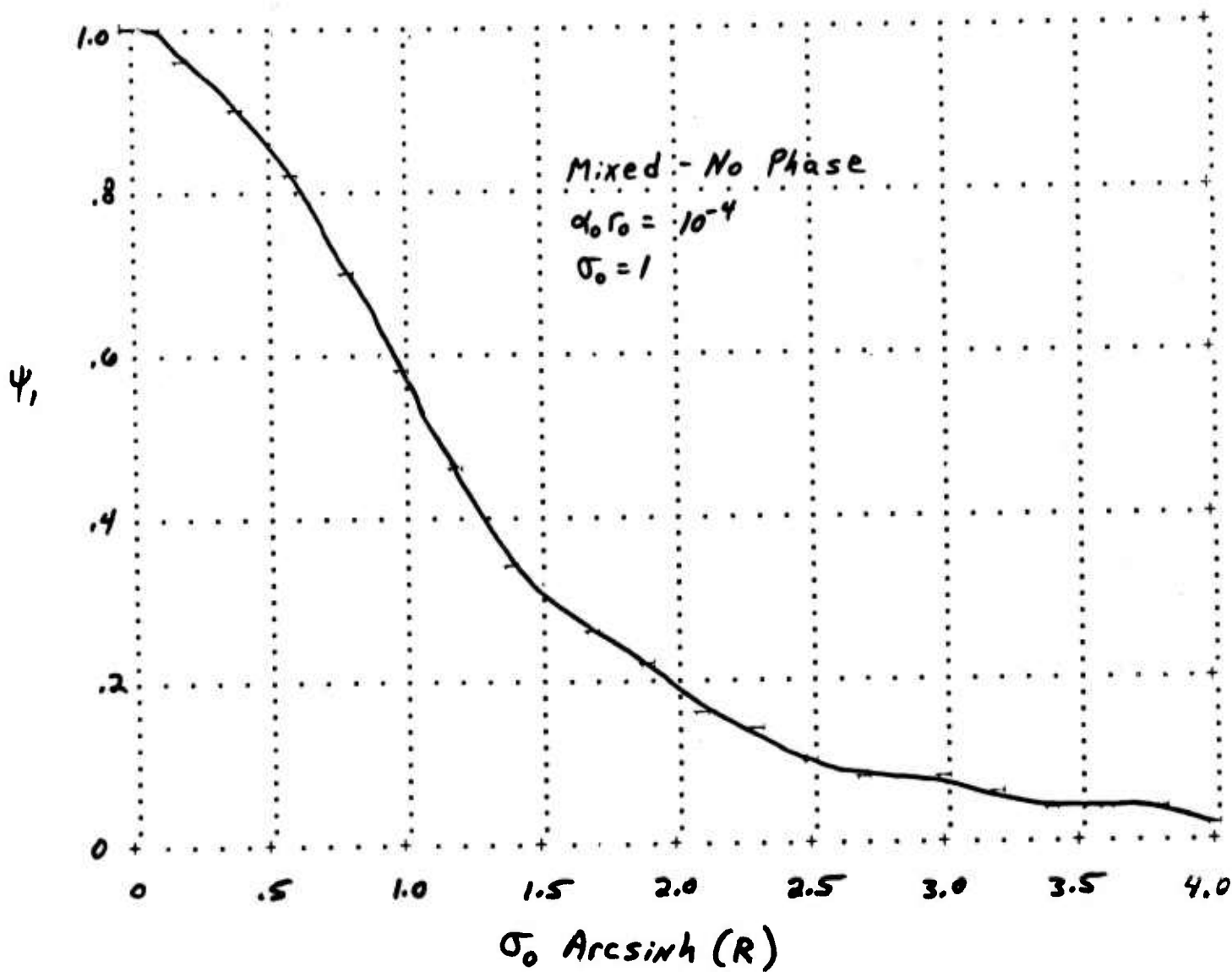
ψ_1 - mixed no phase

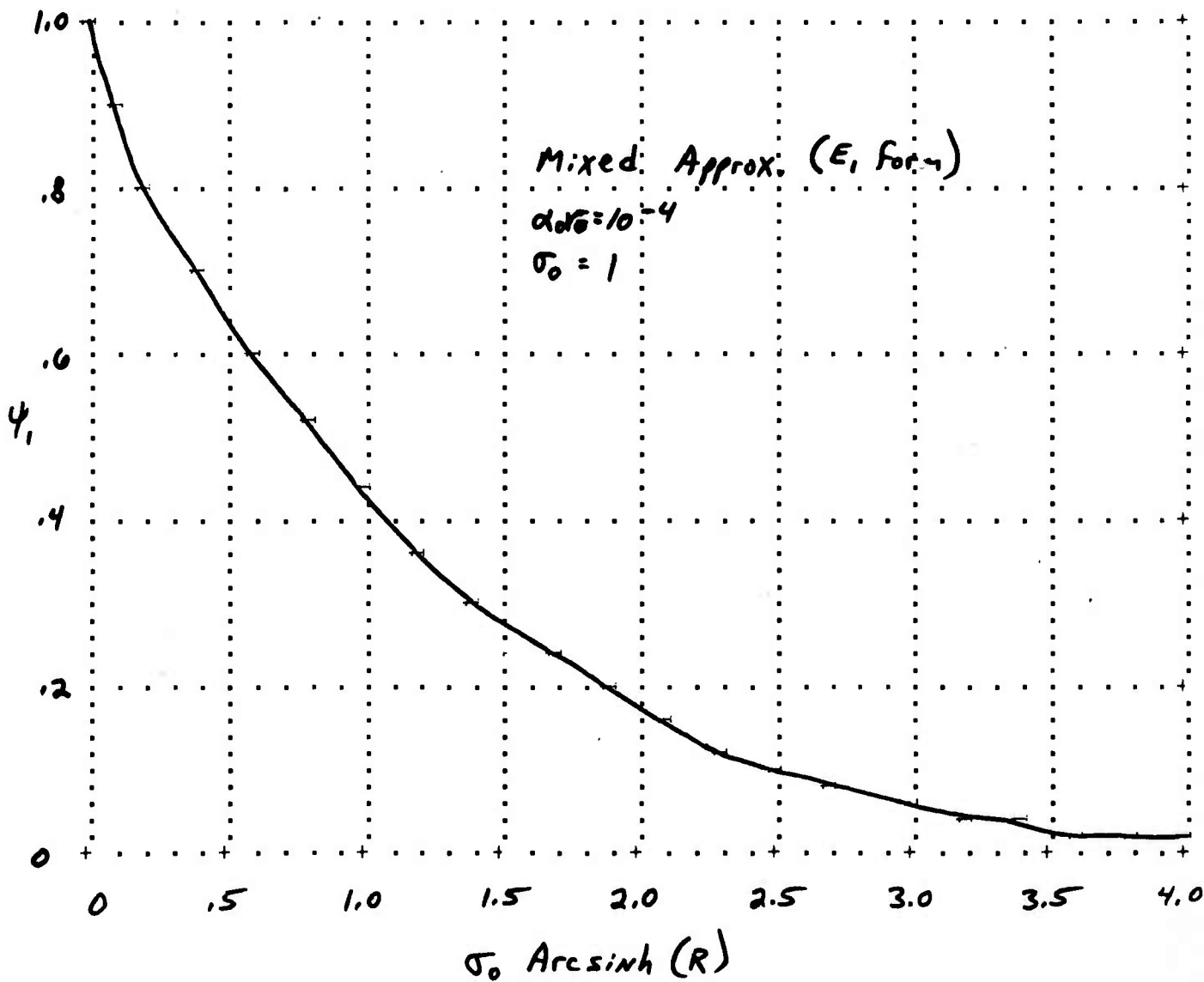
ψ_1 - mixed approximate E1 form

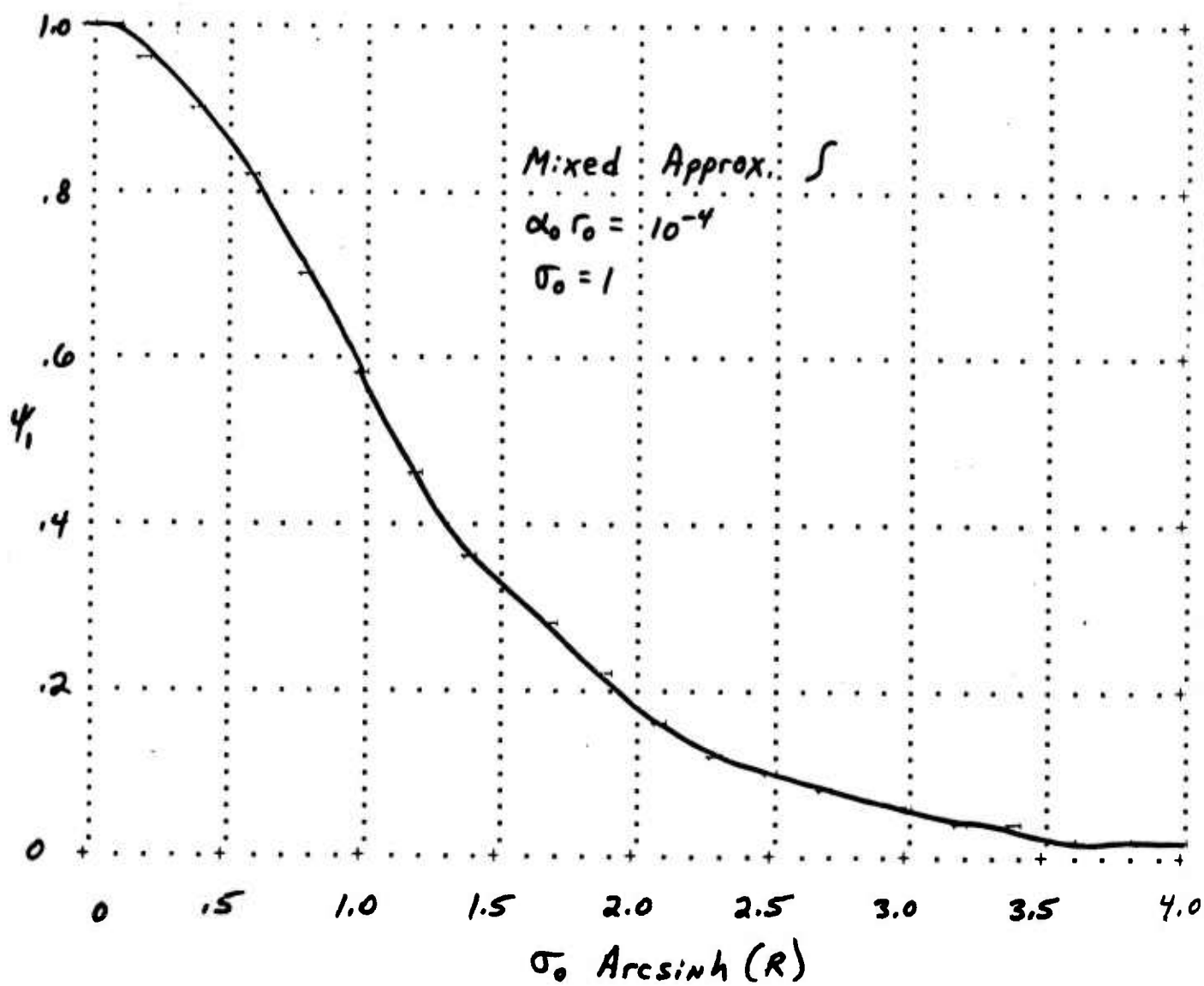
ψ_1 - mixed approximate integral form

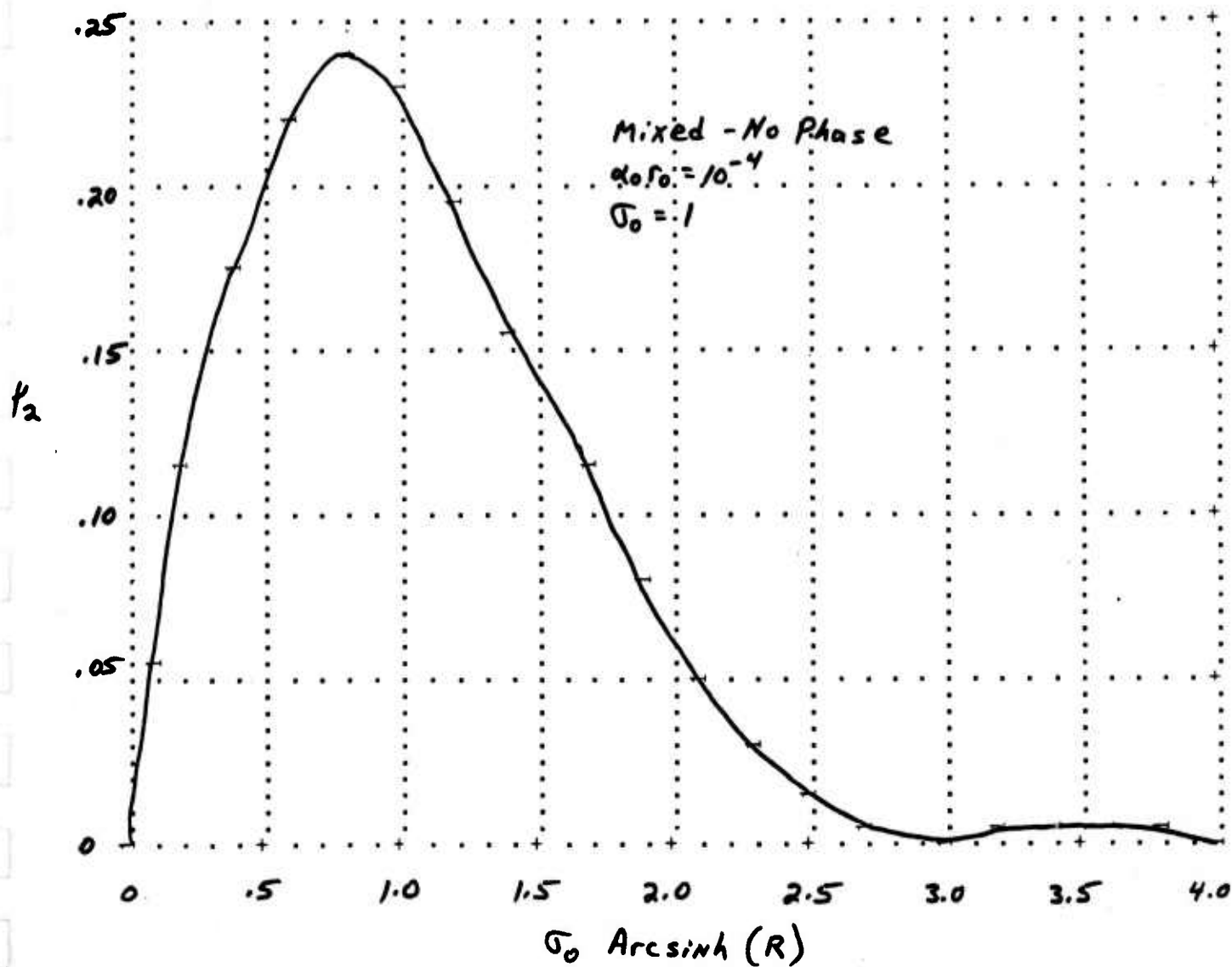
ψ_2 - mixed no phase

EXDB - mixed approximate integral form









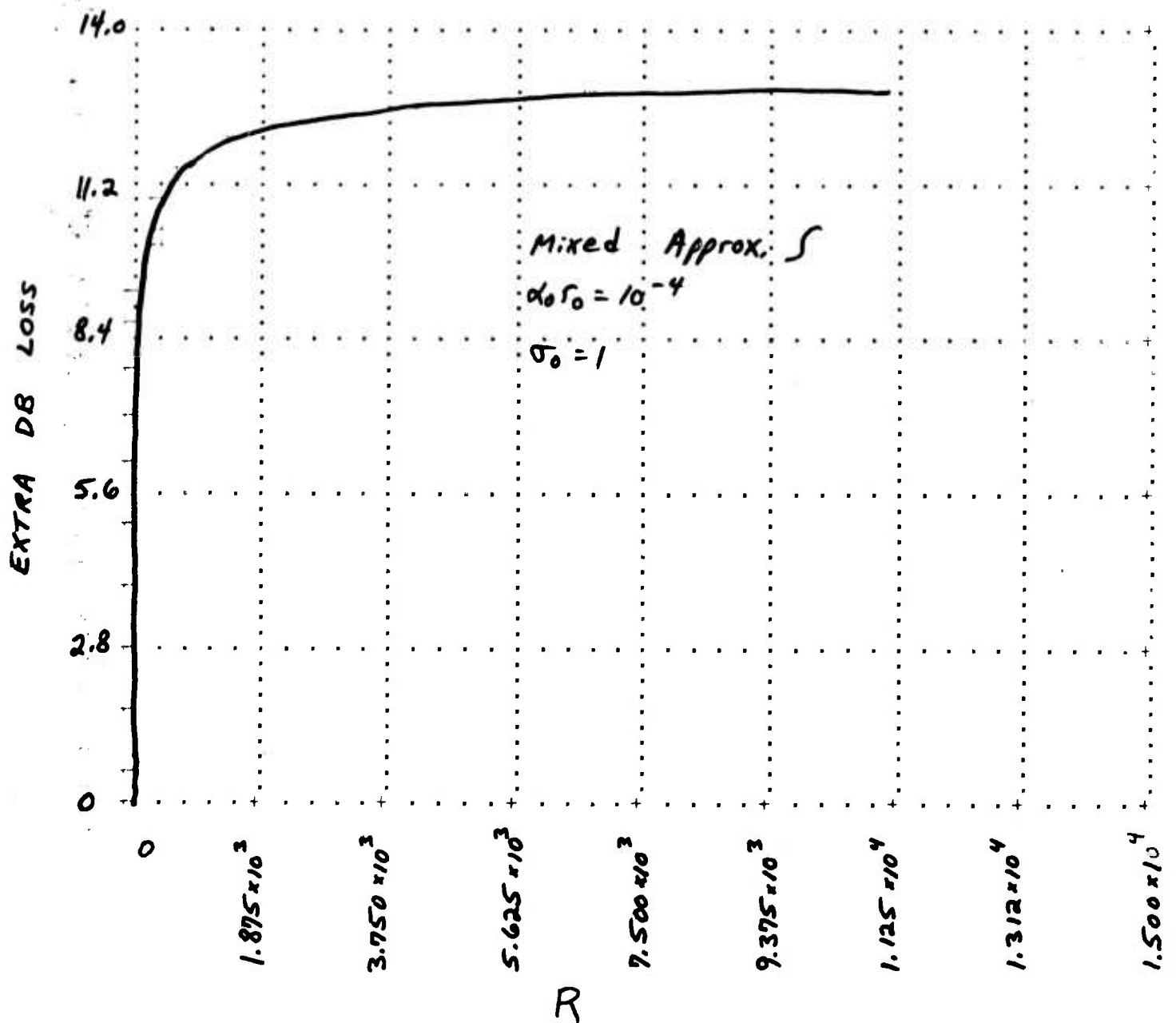


Figure 6
(pages F24to F29)

Mixed case

$$\alpha_o r_o = 10^{-2}$$

$$\sigma_o = 1$$

ψ_1 - mixed no phase

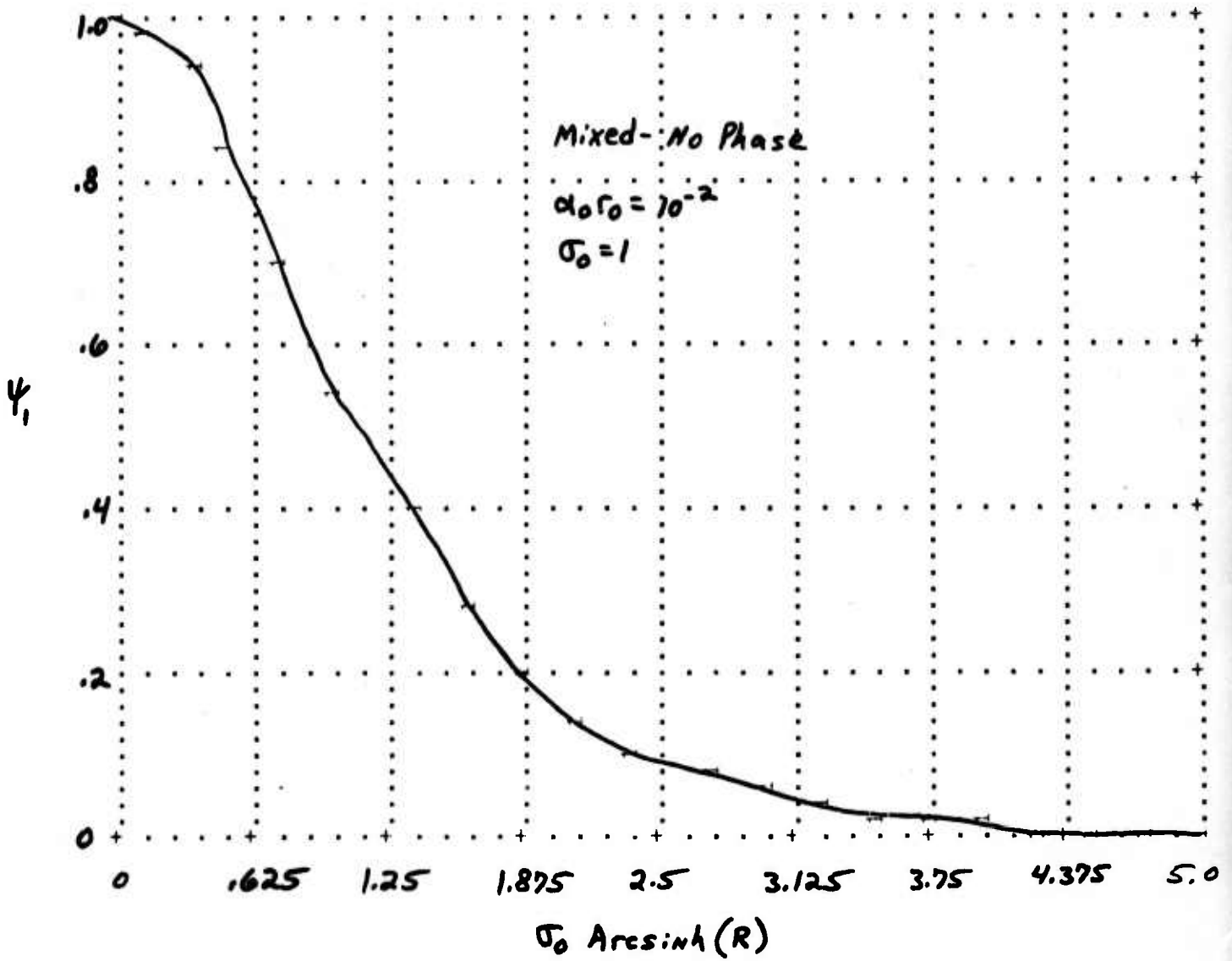
ψ_1 - mixed approximate E1 form

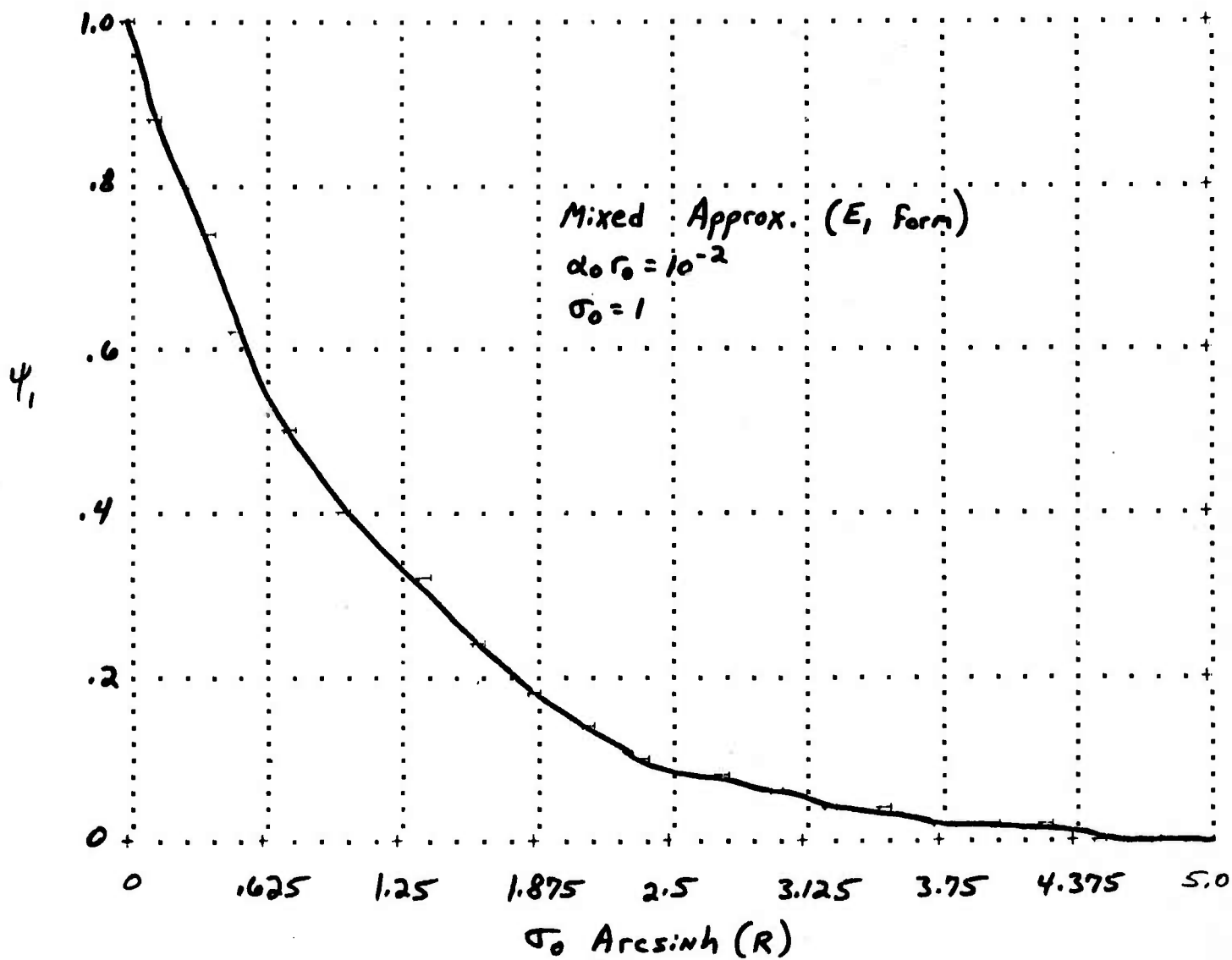
ψ_1 - mixed approximate integral form

ψ_2 - mixed no phase

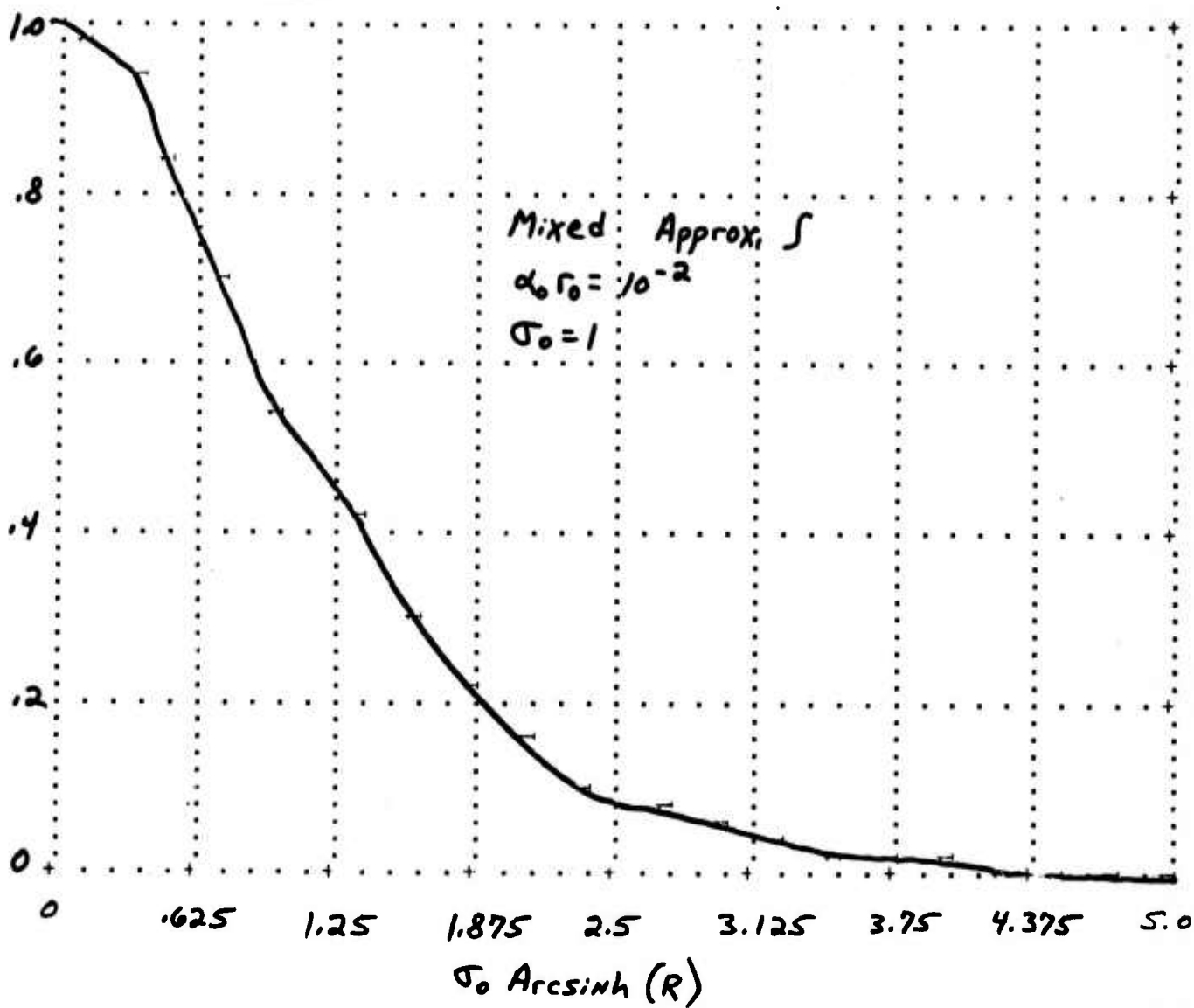
EXDB - mixed no phase

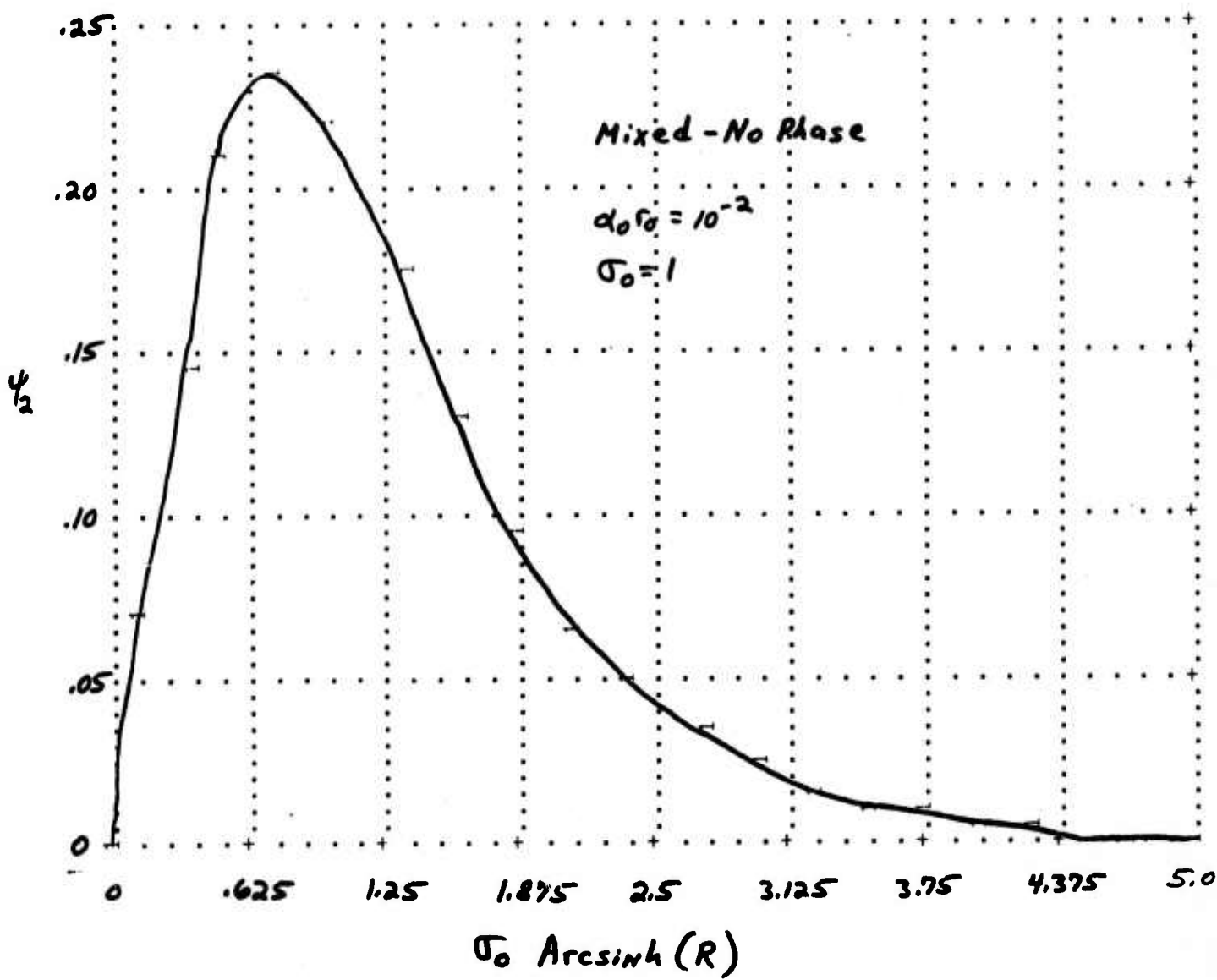
EXDB - mixed approximate integral form



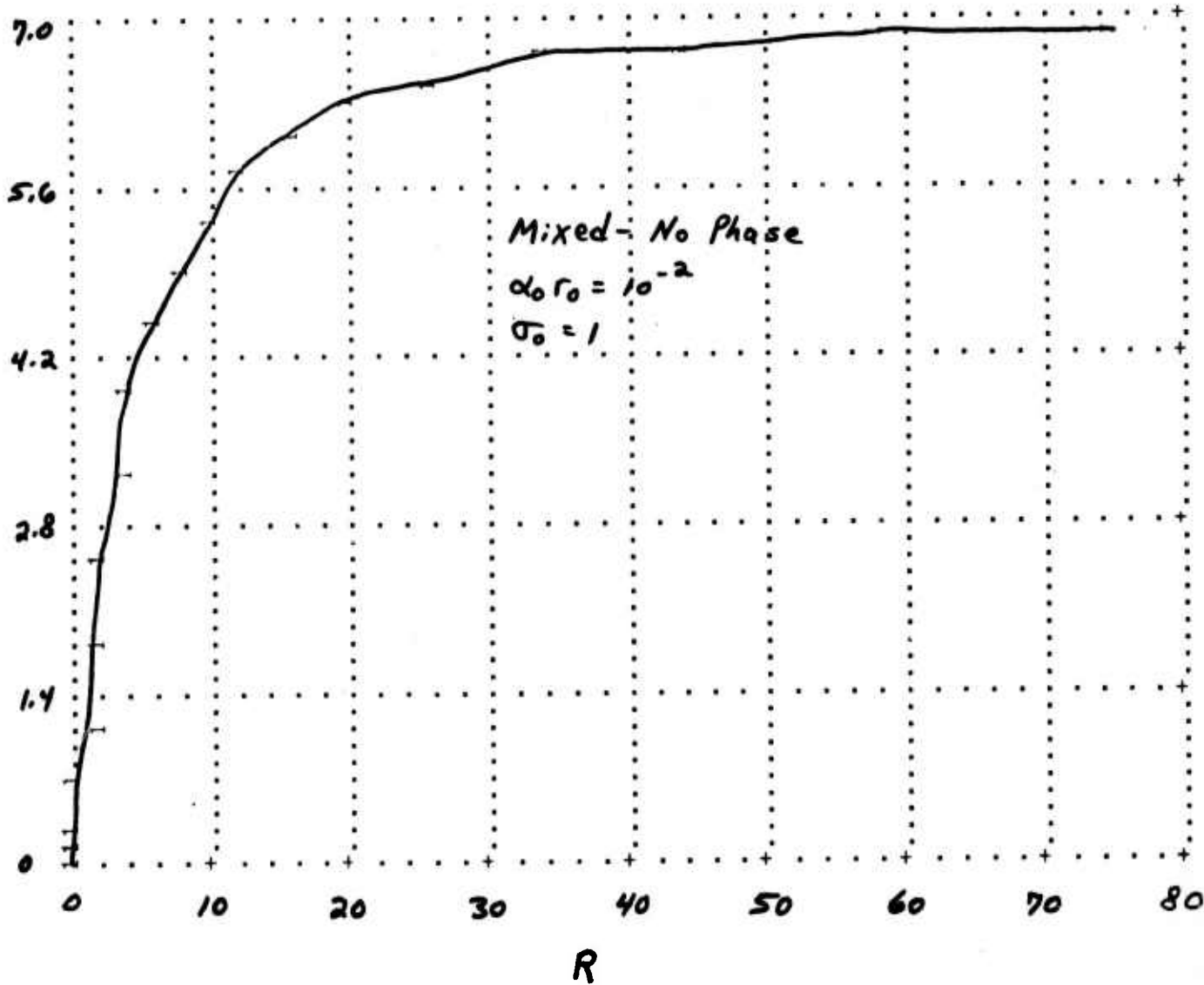


ψ_1





EXTRA DB LOSS



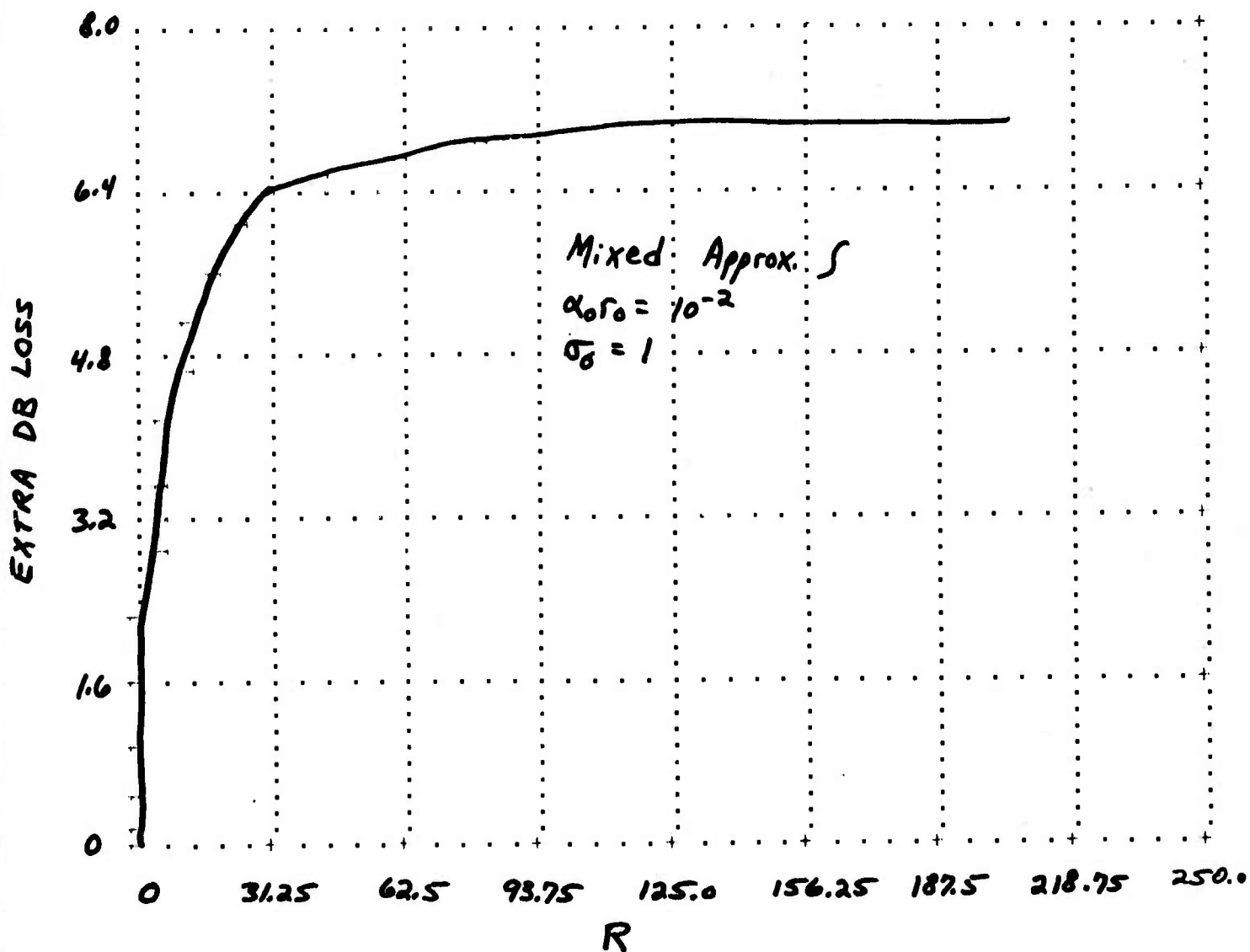


Figure 7
(pages F31to F38)

Mixed case

$$\alpha_o r_o = 7.35 \times 10^{-3}$$

$$\sigma_o = 0.1707, 0.3035, 0.5398, \\ 0.9600, 1.7071$$

ψ_1 - mixed no phase

ψ_1 - mixed phase

ψ_1 - mixed approximate integral form

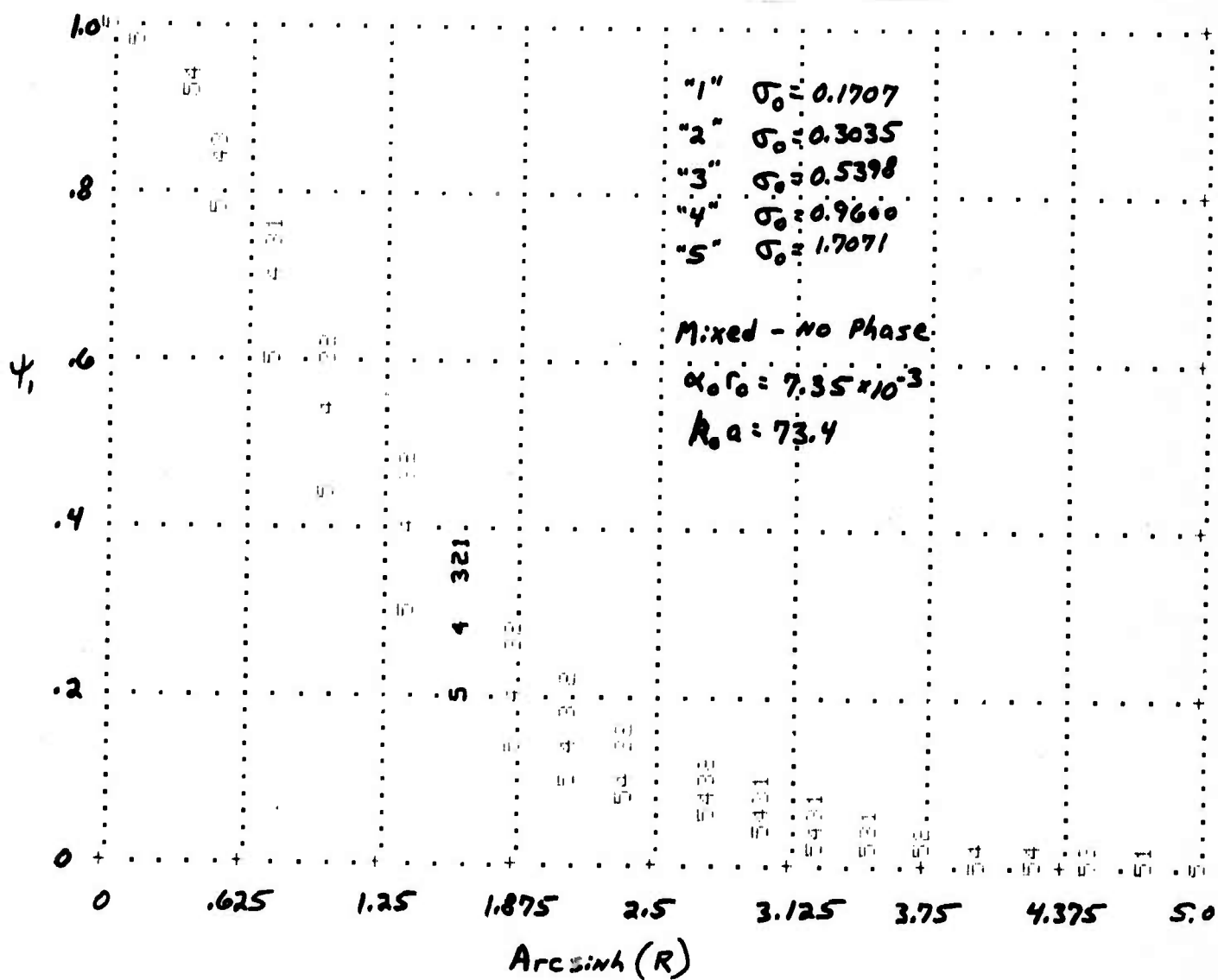
ψ_2 - mixed no phase

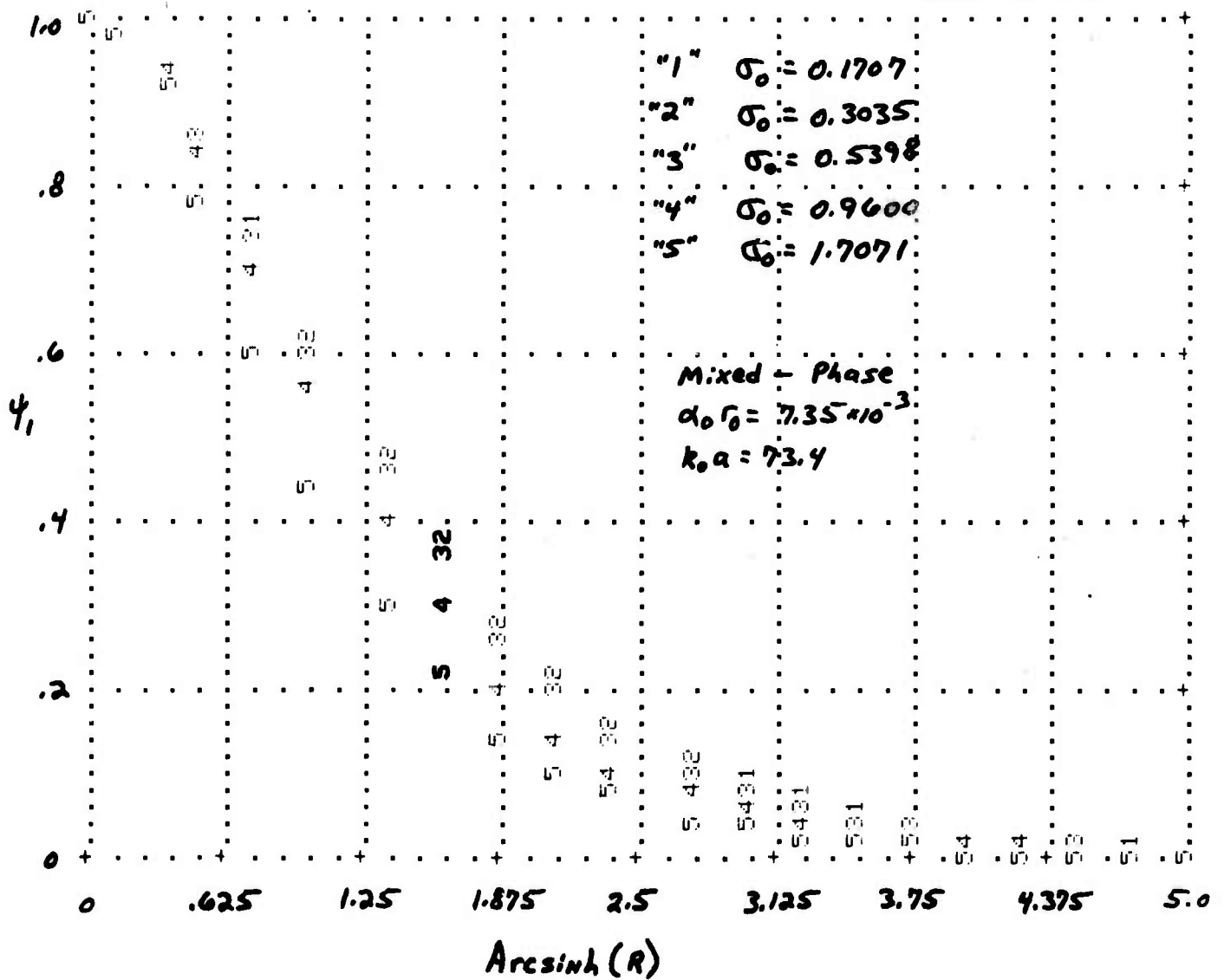
ψ_2 - mixed phase

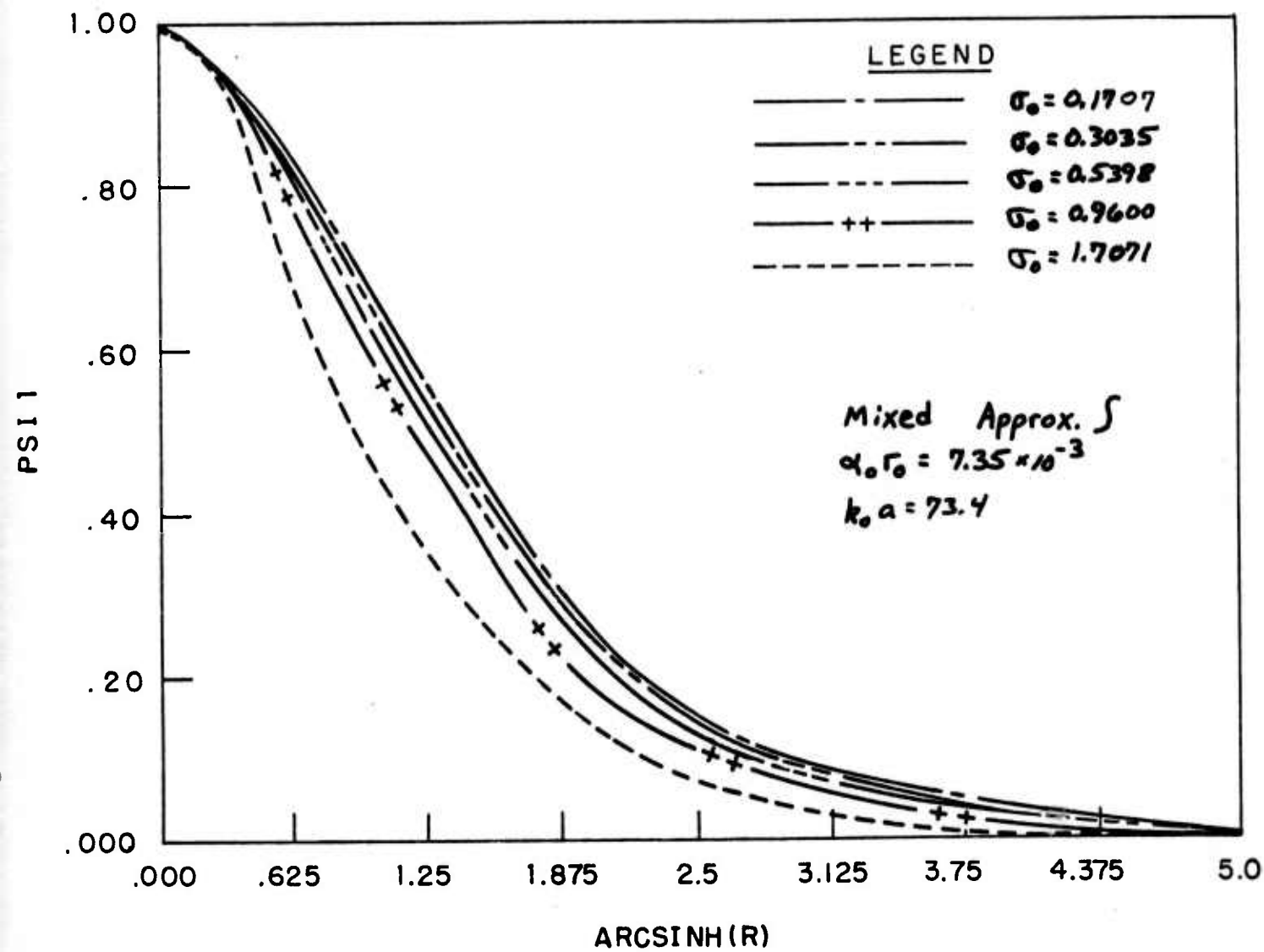
EXDB - mixed no phase

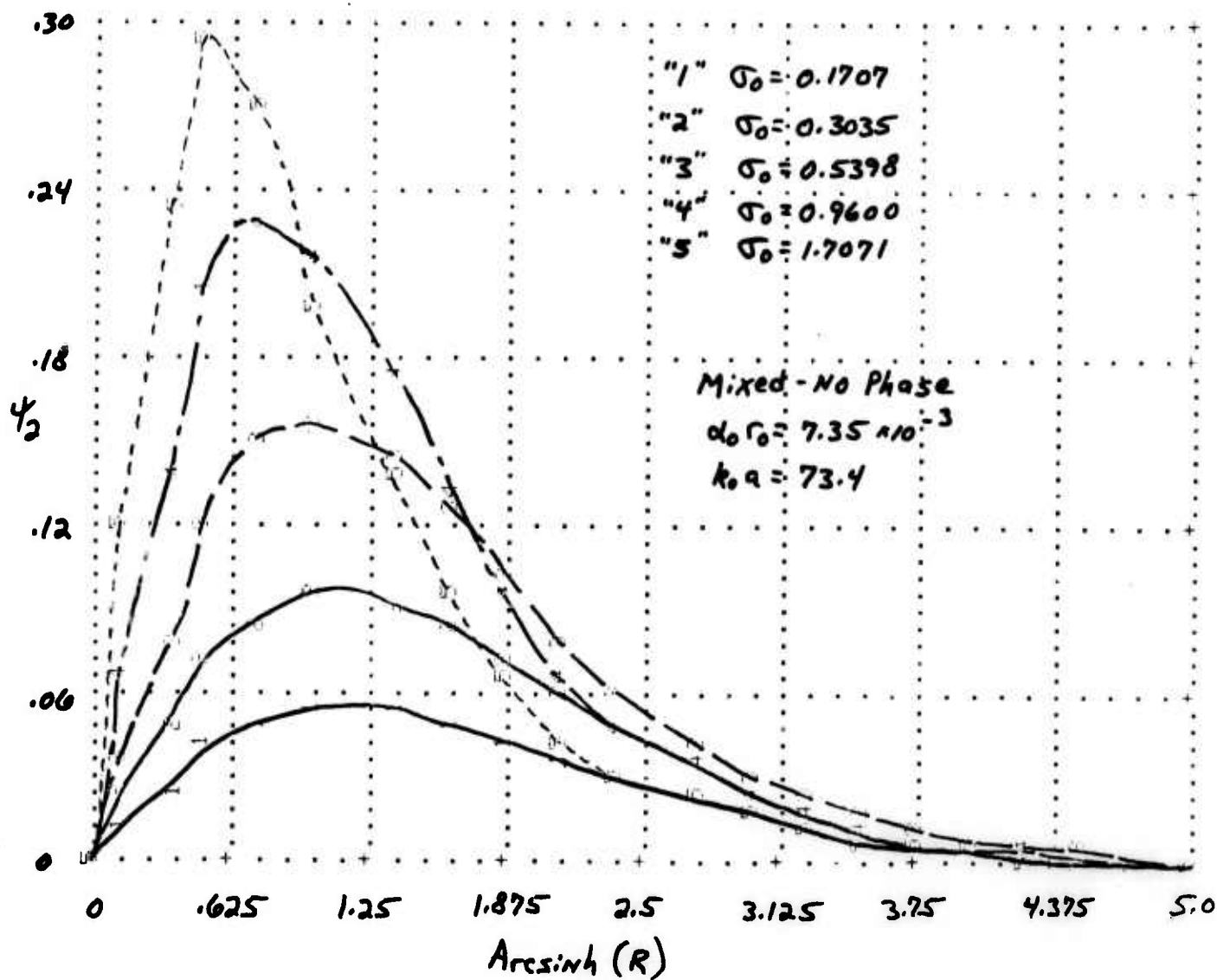
EXDB - mixed phase

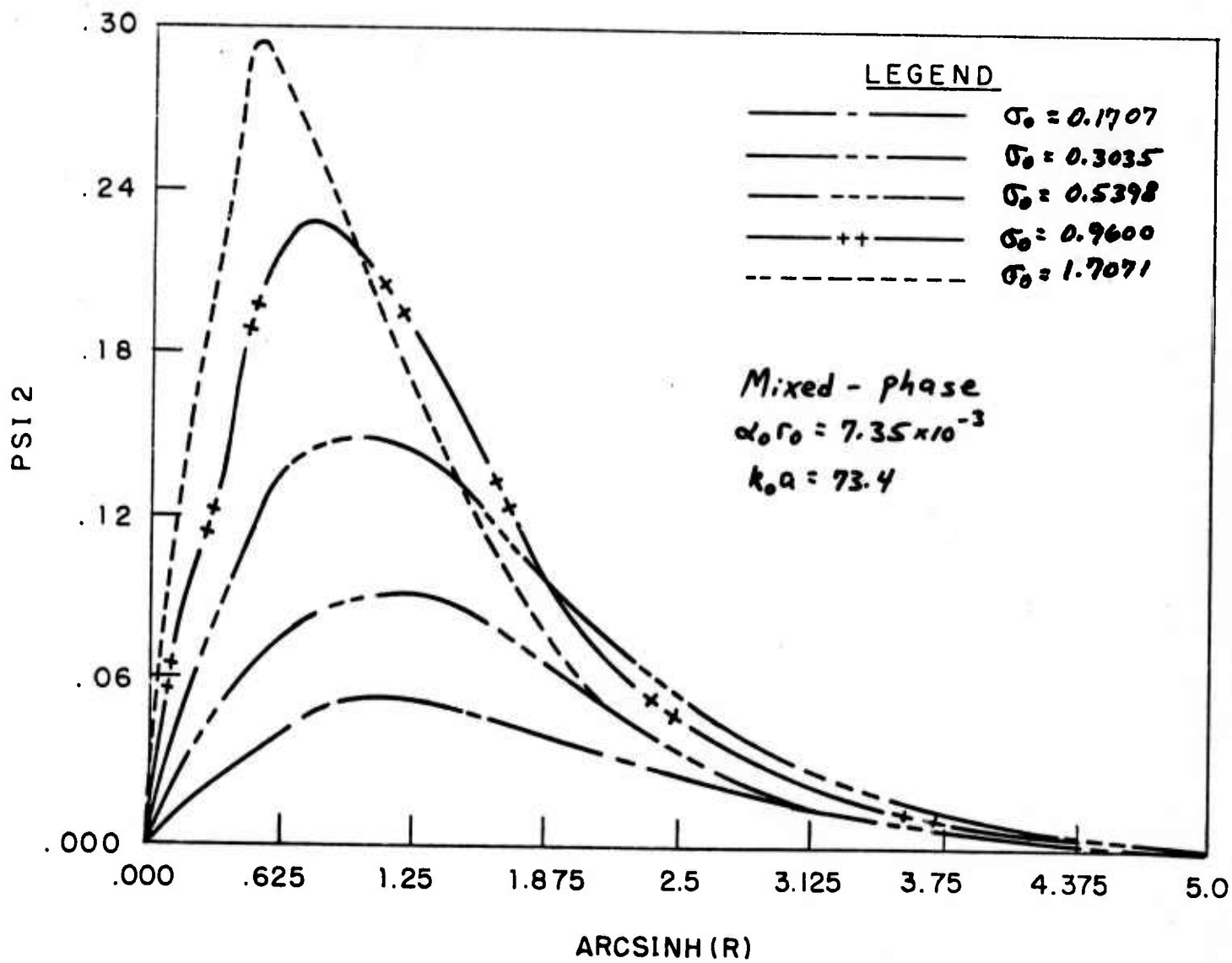
EXDB - mixed approximate integral form

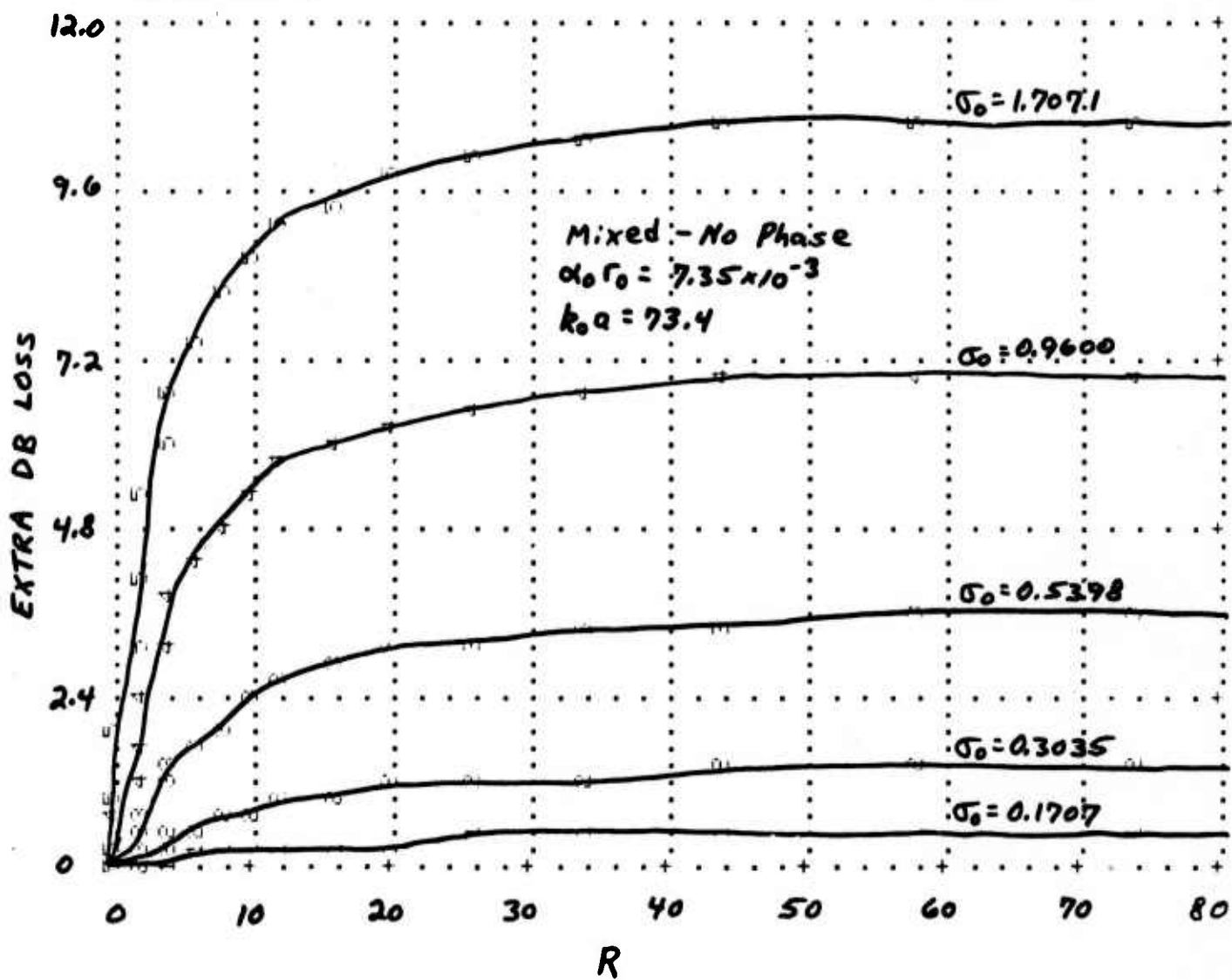


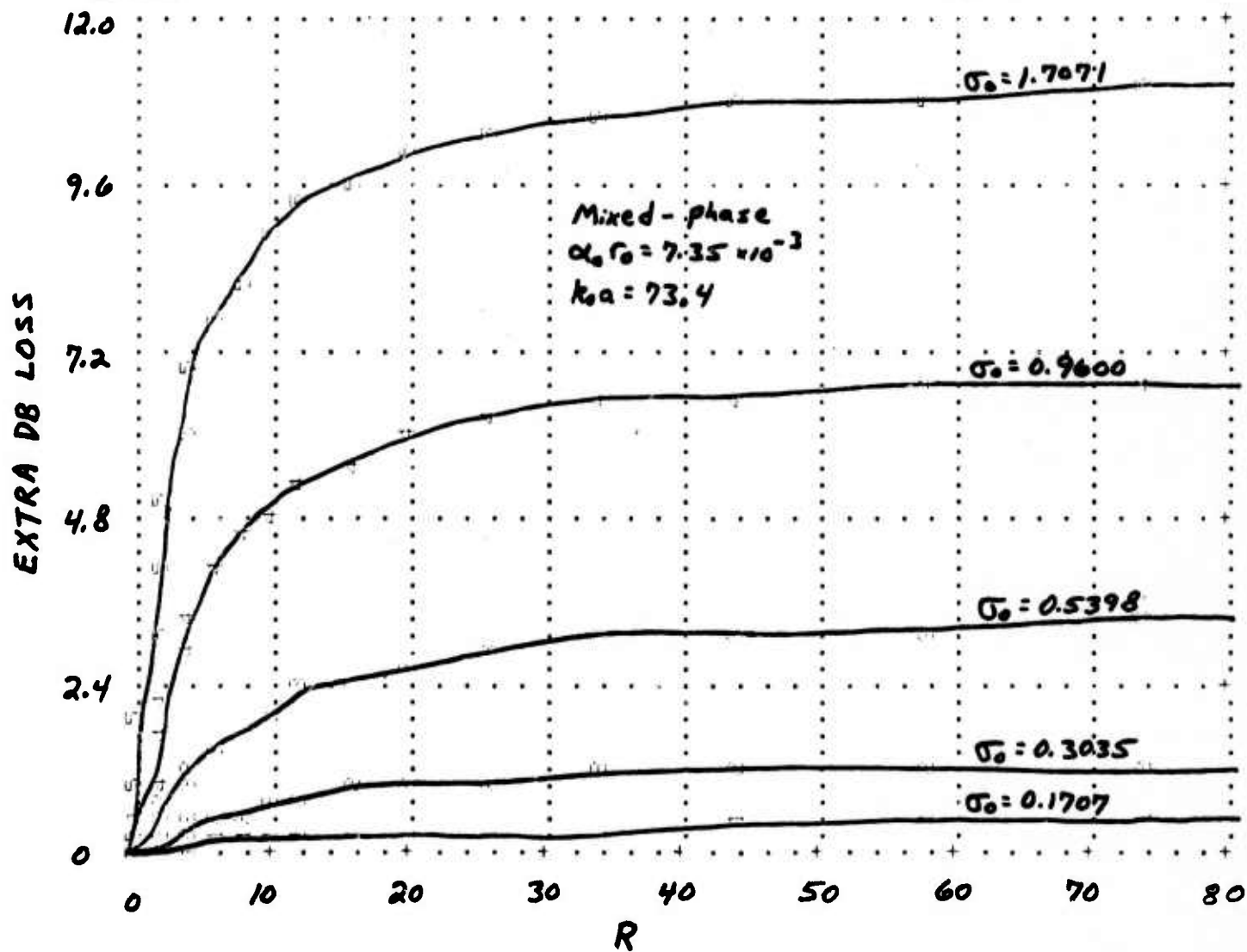












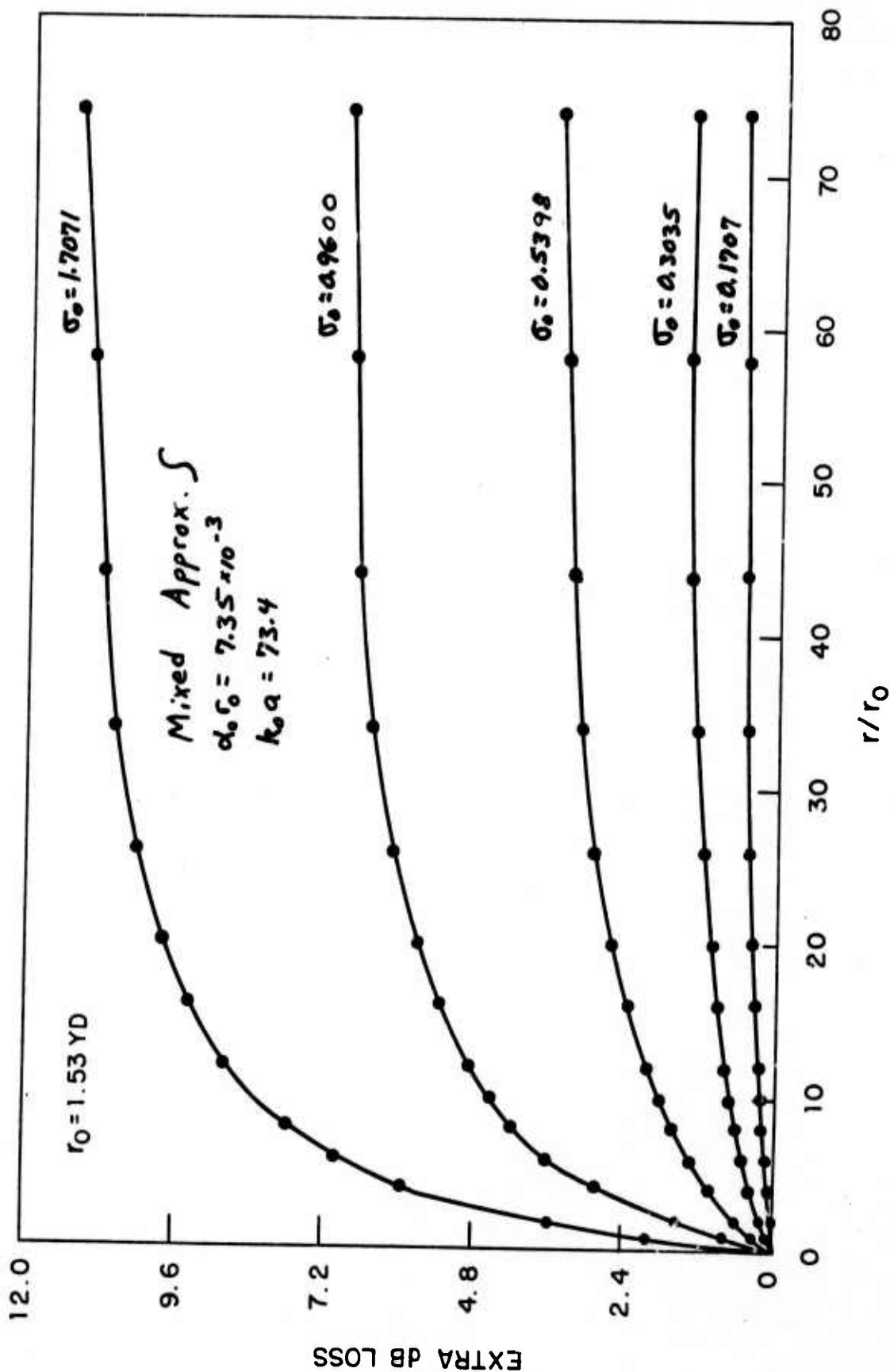


Figure 8

(page F40)

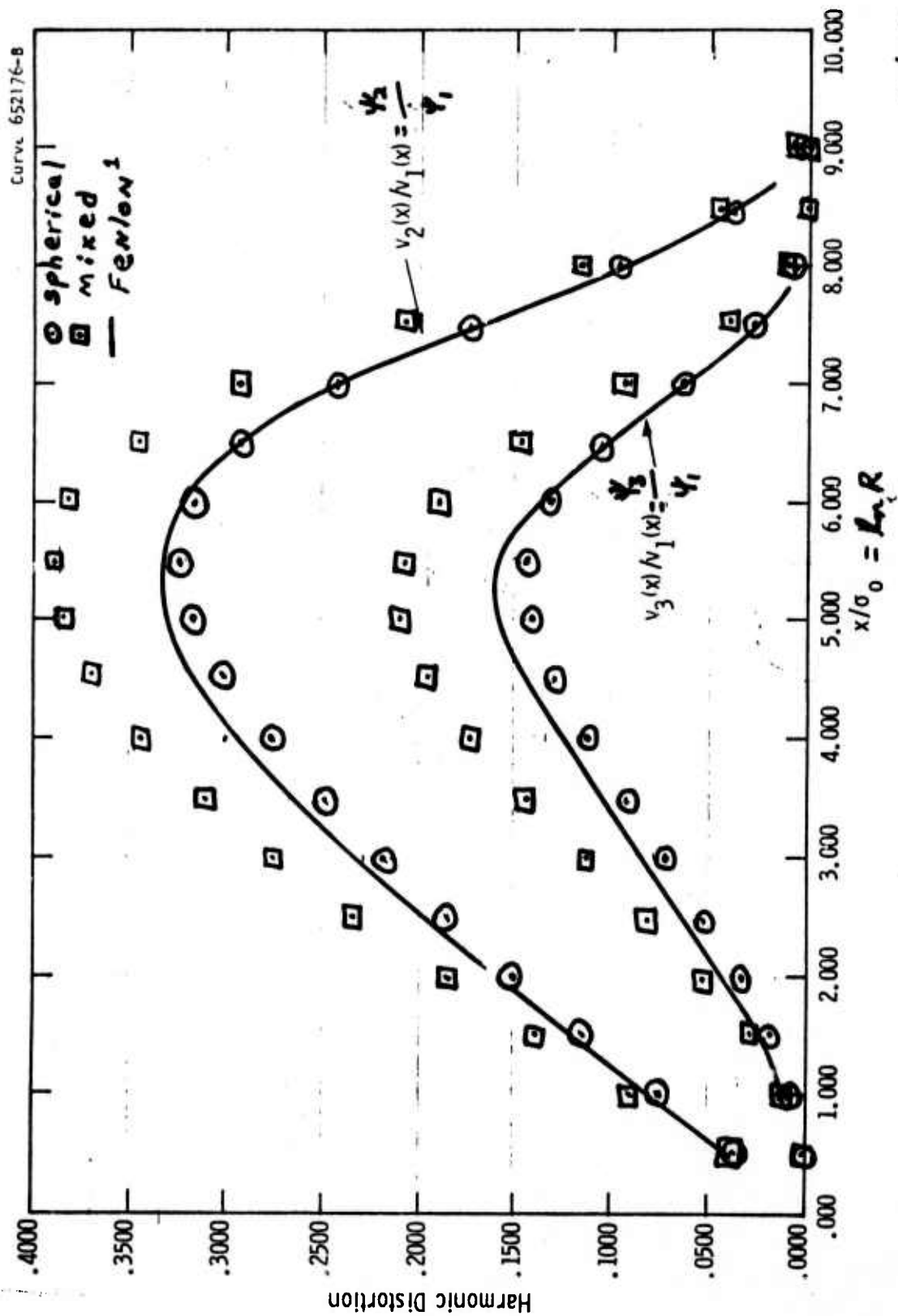
Spherical and mixed cases

$$\sigma_0 / \alpha_0 r_0 = 727$$

$$\sigma_0 = 0.1538$$

$$\psi_2 / \psi_1$$

$$\psi_3 / \psi_1$$



$$\Gamma_0 = 727 = \sigma_0/\sigma_0$$

$$\sigma_0 = 1.538 \times 10^{-1}$$

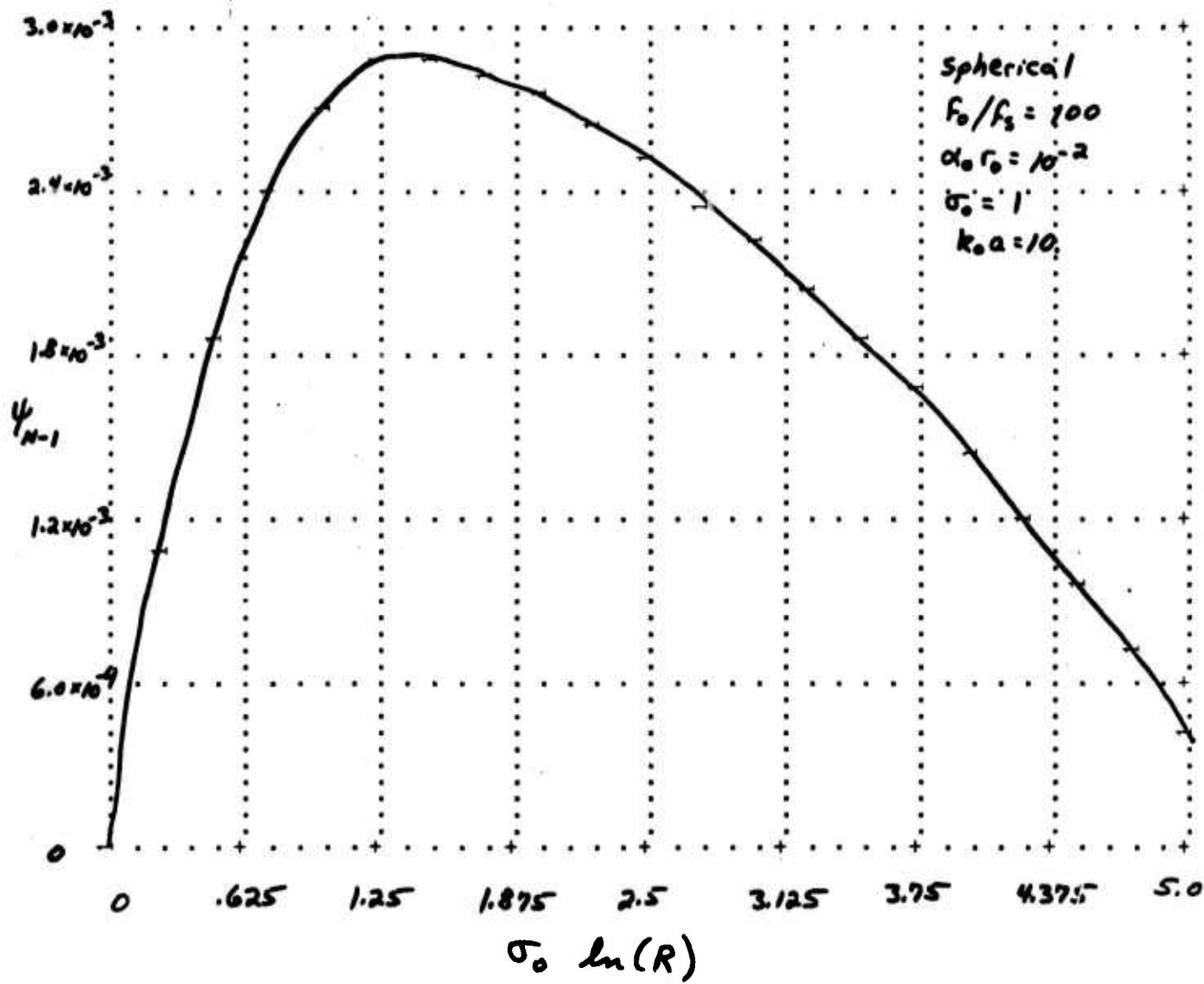
Figure 9
(pages F42to F43)

Spherical wave case

$$\alpha_o r_o = 10^{-2}$$

$$\sigma_o = 1$$

$$\psi_{NFR-1}, \psi_{NFR+1}$$



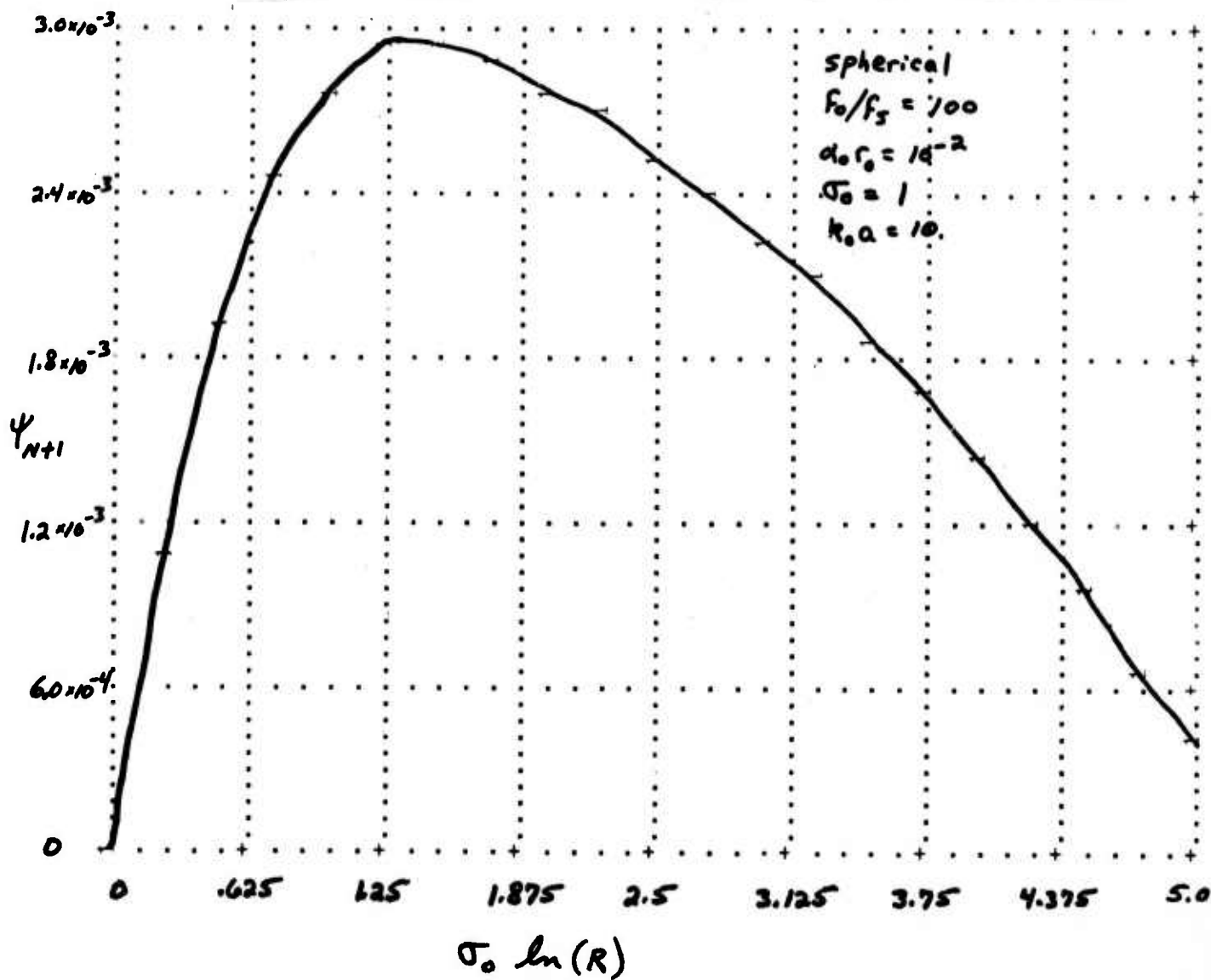


Figure 10
(pages F45 to F47)

Mixed case

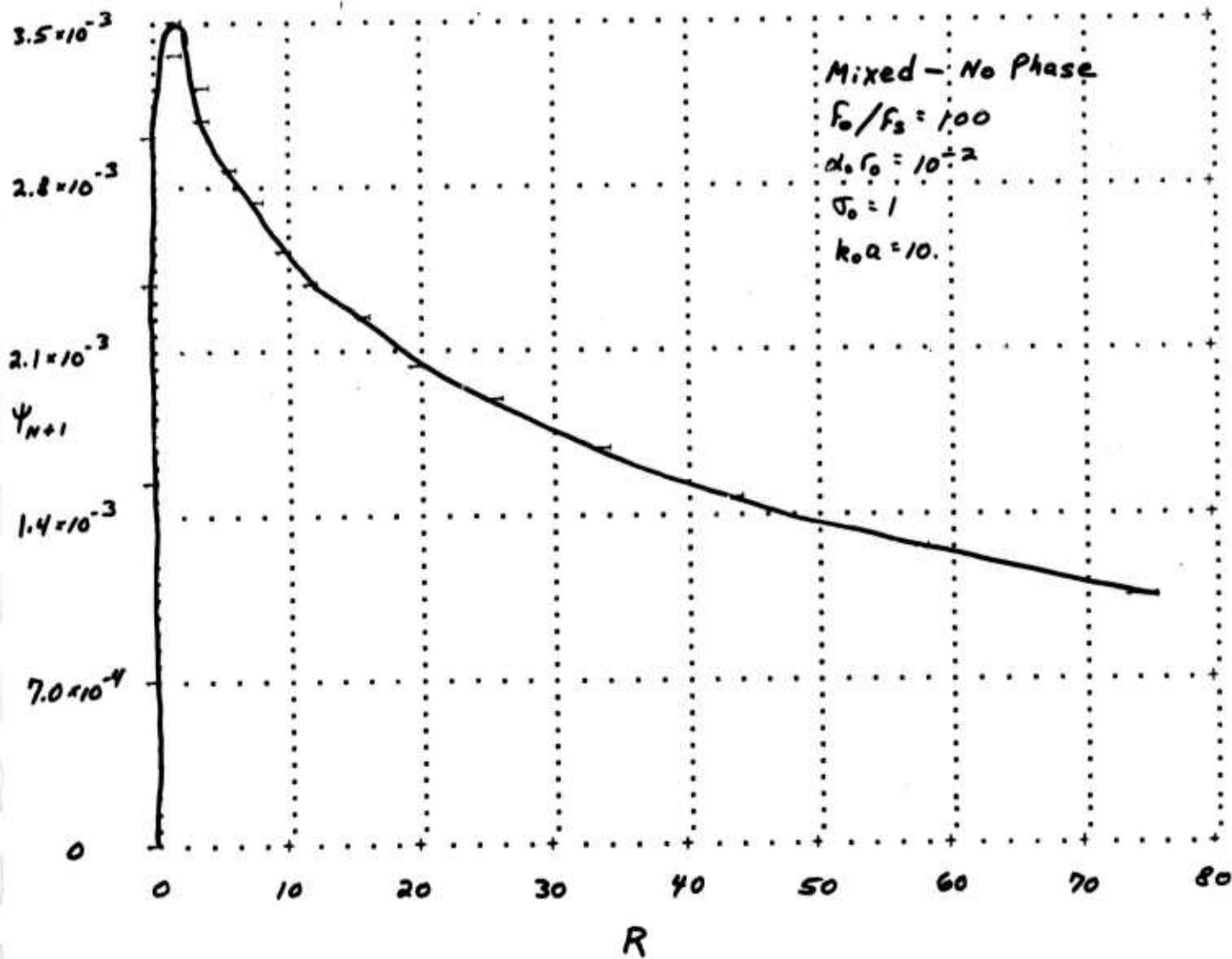
$$\alpha_o r_o = 10^{-2}$$

$$\sigma_o = 1$$

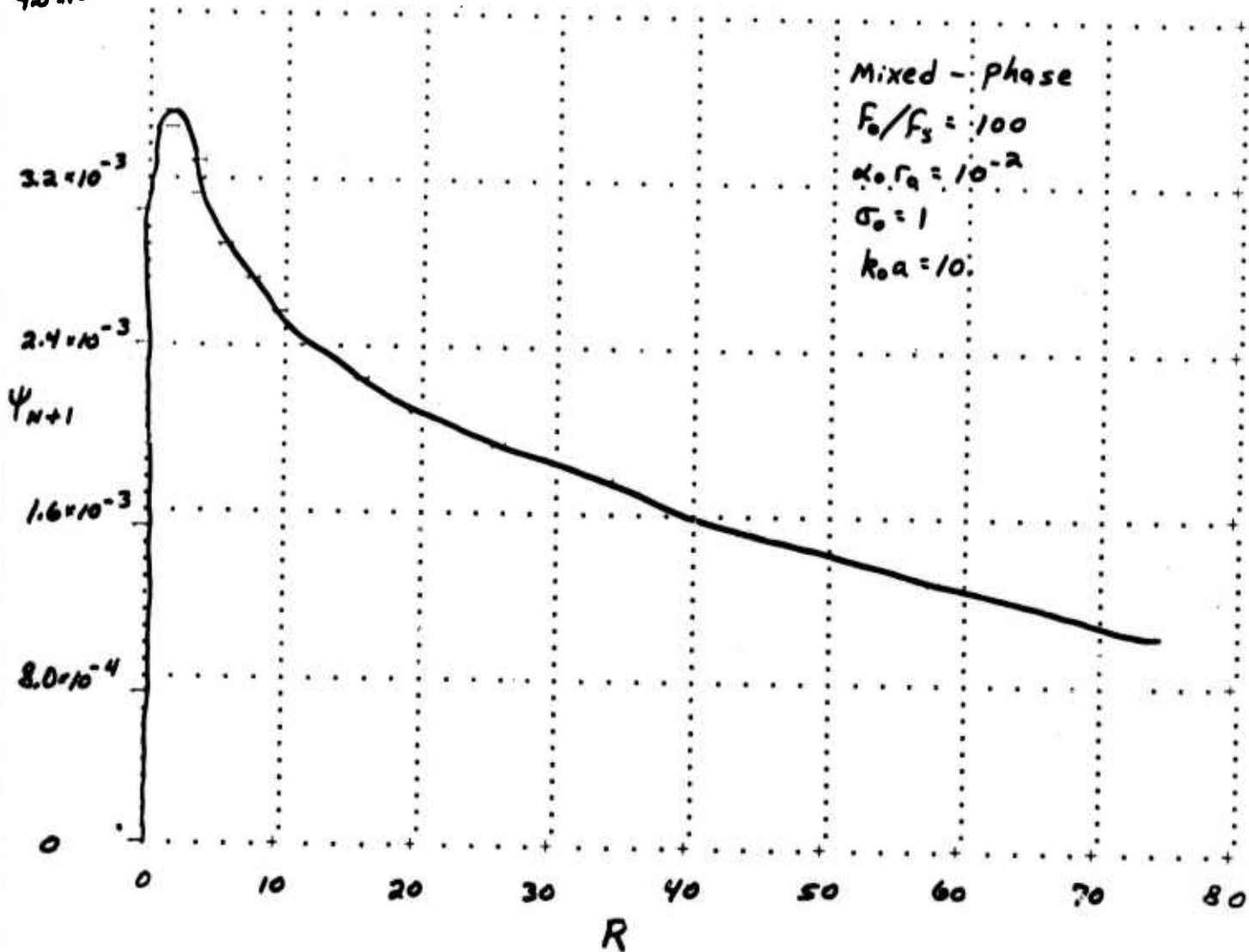
ψ_{NFR+1} - mixed no phase

ψ_{NFR+1} - mixed phase

ψ_{NFR+1} - mixed approximate integral form



4.0×10^{-3}



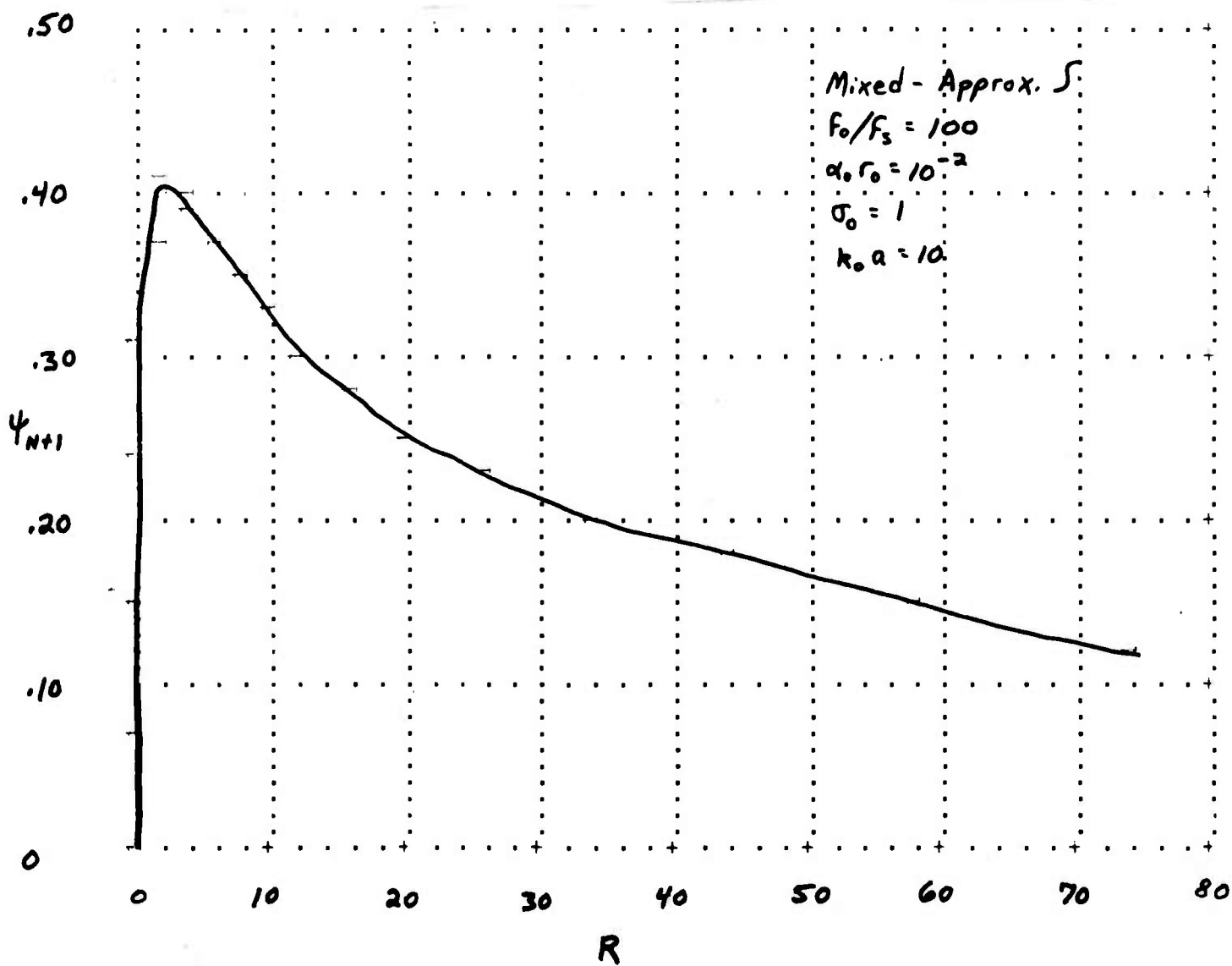


Figure 11
(pages F49 to F51)

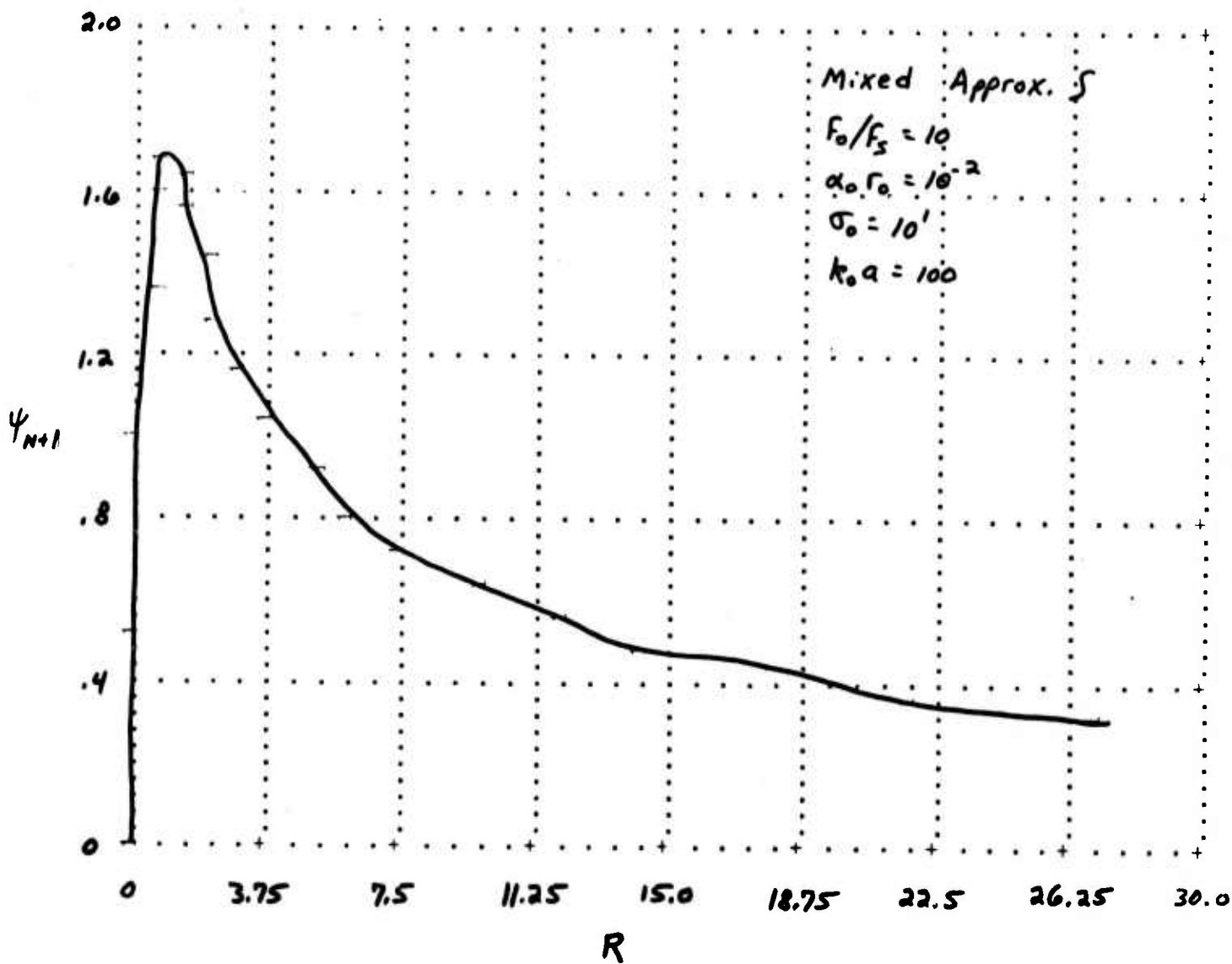
Mixed approximate integral case

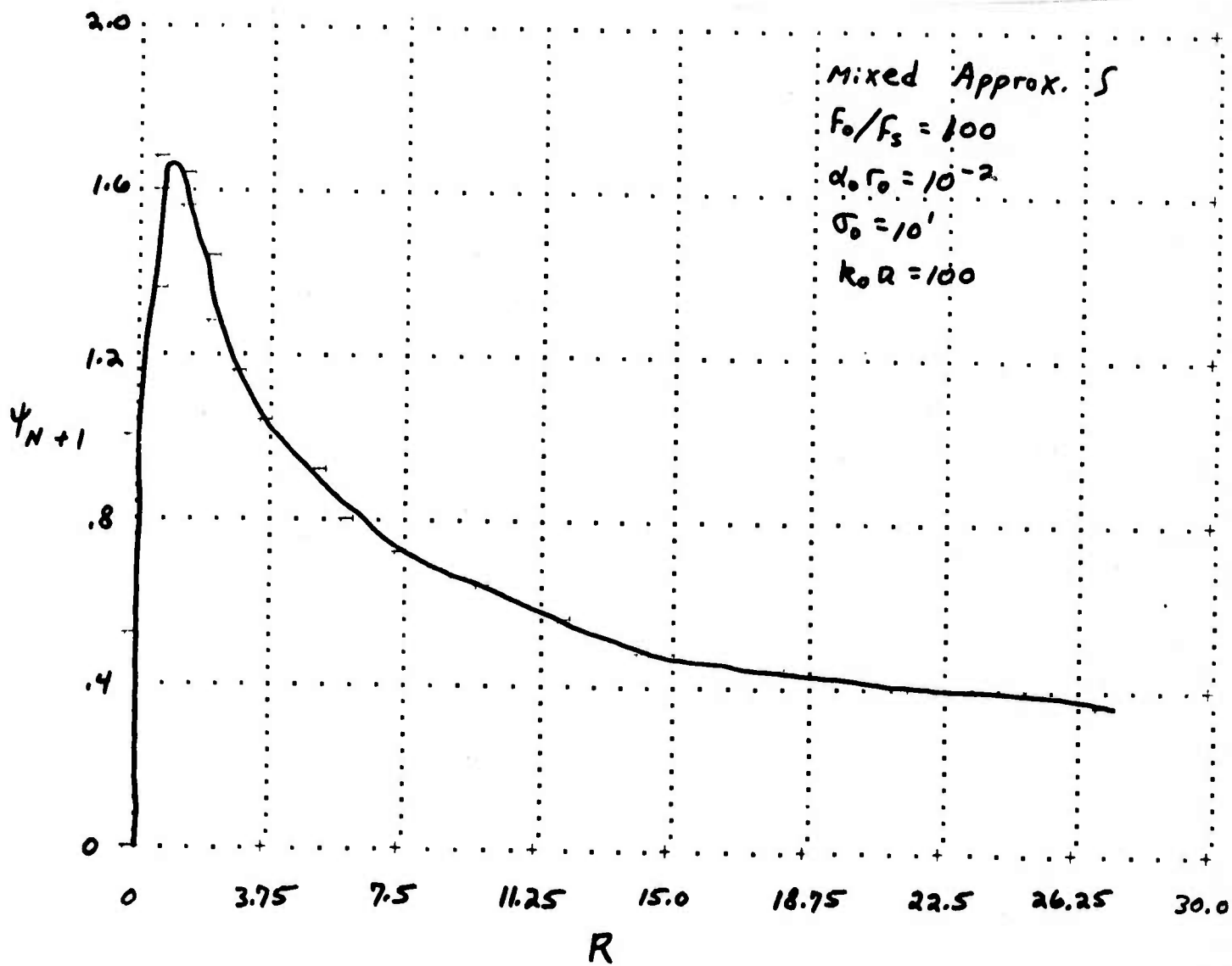
$$\alpha_o r_o = 10^{-2}$$

$$\sigma_o = 10$$

$$f_o/f_s = 10, 100, 1000$$

$$\psi_{NFR+1}$$





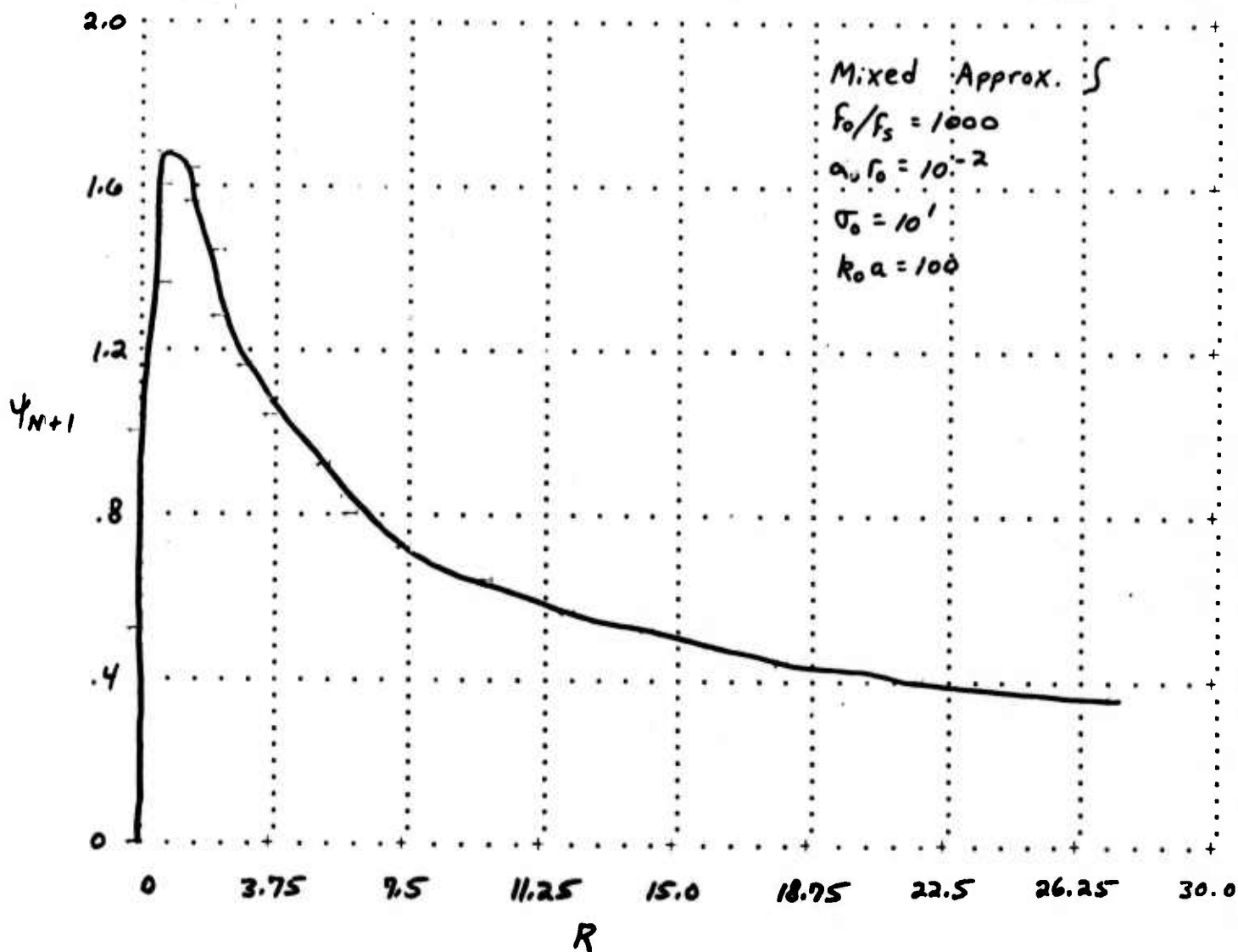


Figure 12
(pages F53 to F56)

Mixed approximate integral case

$$\alpha_o r_o = 10^{-4}, 10^1$$

$$\sigma_o = 10^{-3}, 10^5$$

$$\psi_{NFR+1}$$

6.0×10^{-4}

4.8×10^{-4}

3.6×10^{-4}

ψ_{N+1}

2.4×10^{-4}

1.2×10^{-4}

0

0

1.875×10^3

3.75×10^3

5.625×10^3

R

7.5×10^3

9.375×10^3

1.125×10^4

1.312×10^4

1.5×10^4

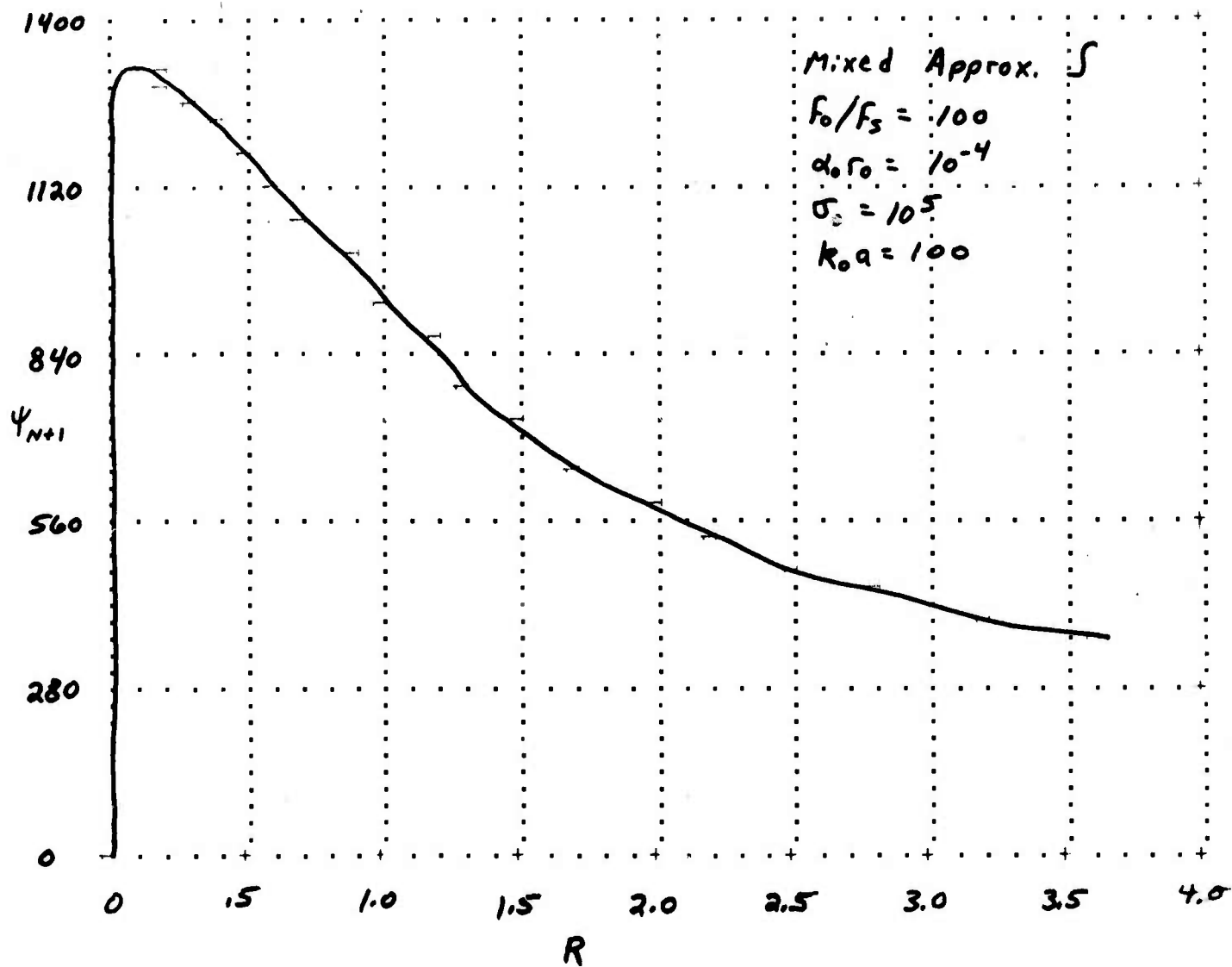
Mixed Approx. S

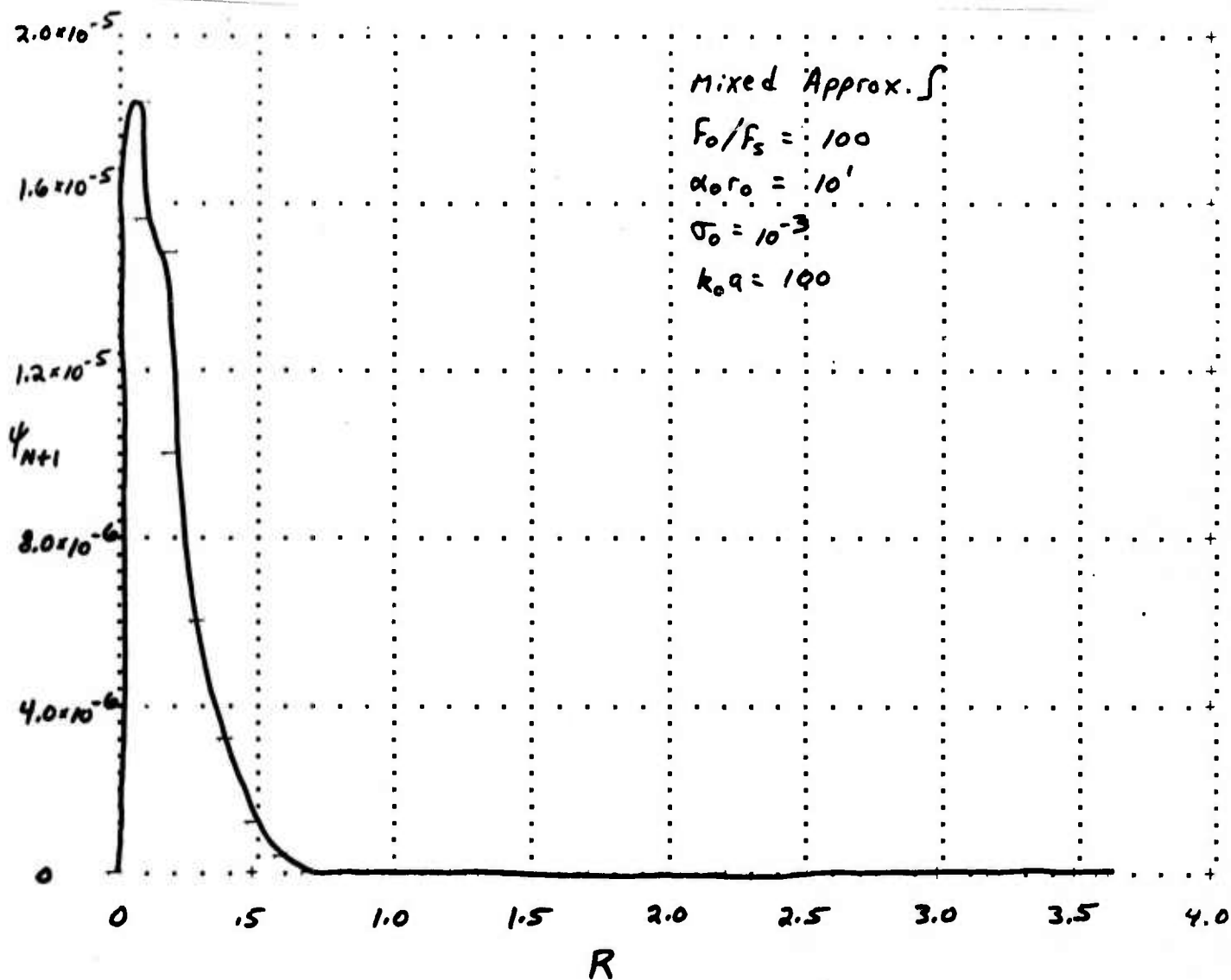
$f_0/f_s = 100$

$\alpha_0 f_0 = 10^{-4}$

$\sigma_0 = 10^{-3}$

$k_0 a = 100$





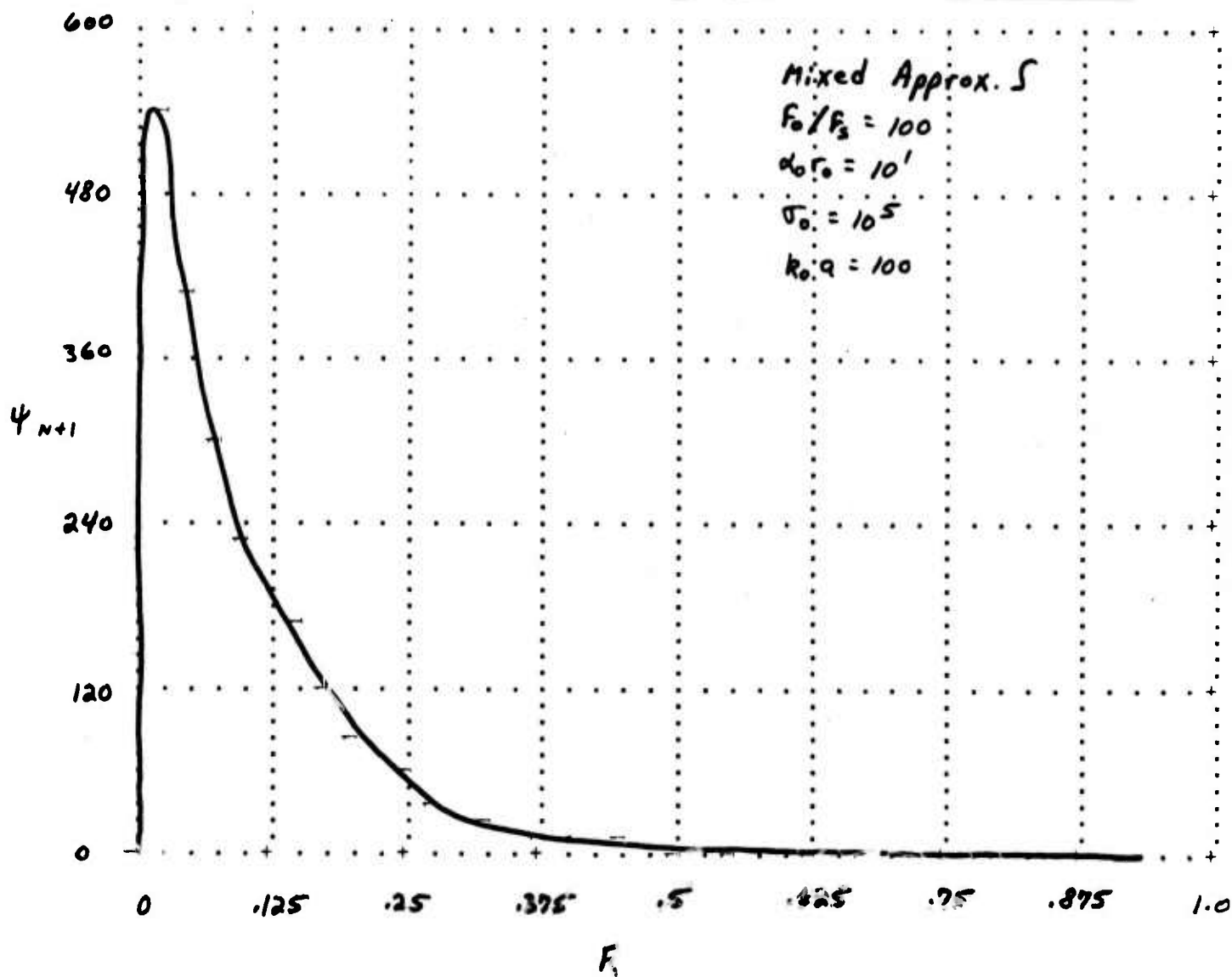


Figure 13
(pages F58to F63)

Beam patterns

Mixed approximate integral case

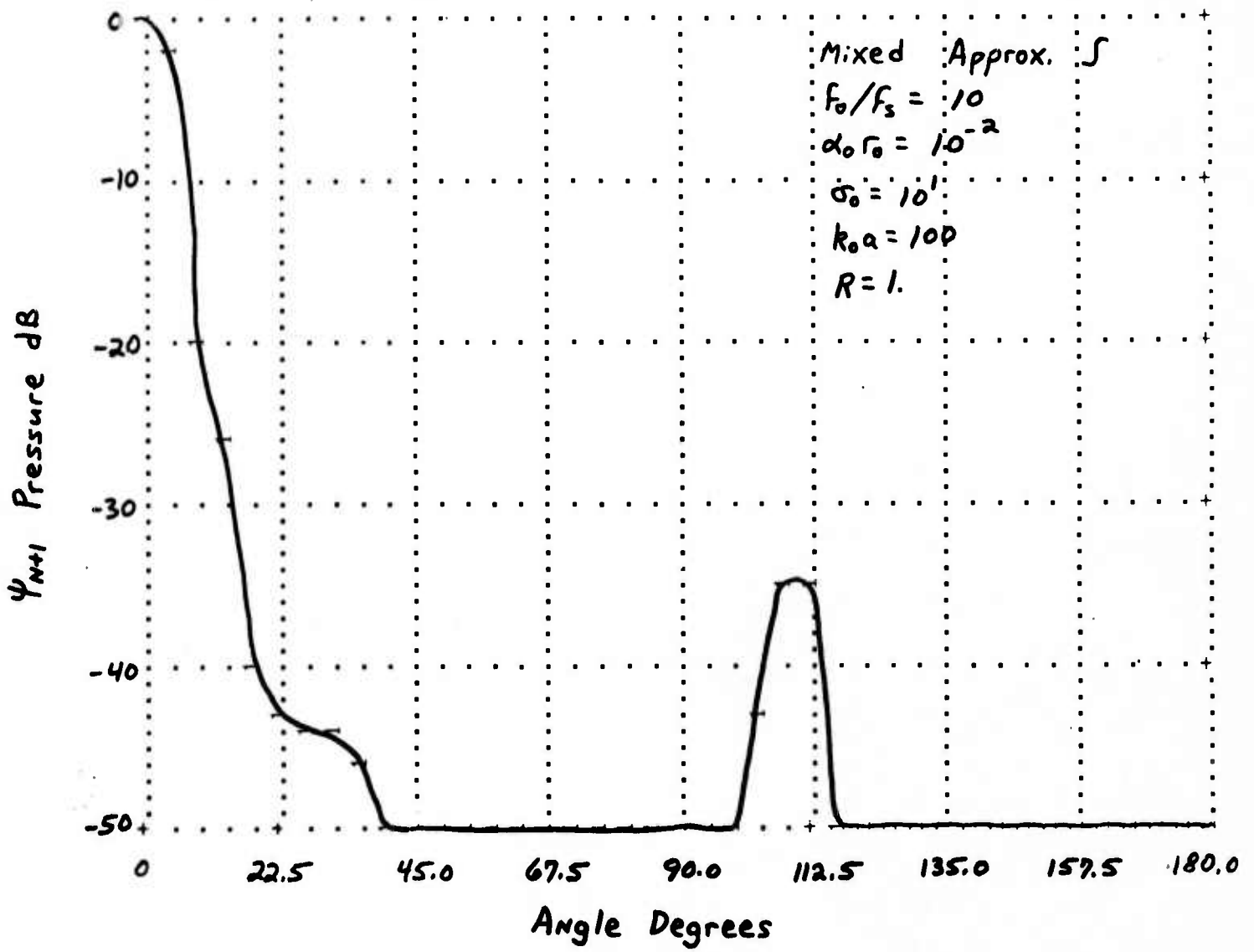
$$\alpha_o r_o = 10^{-2}$$

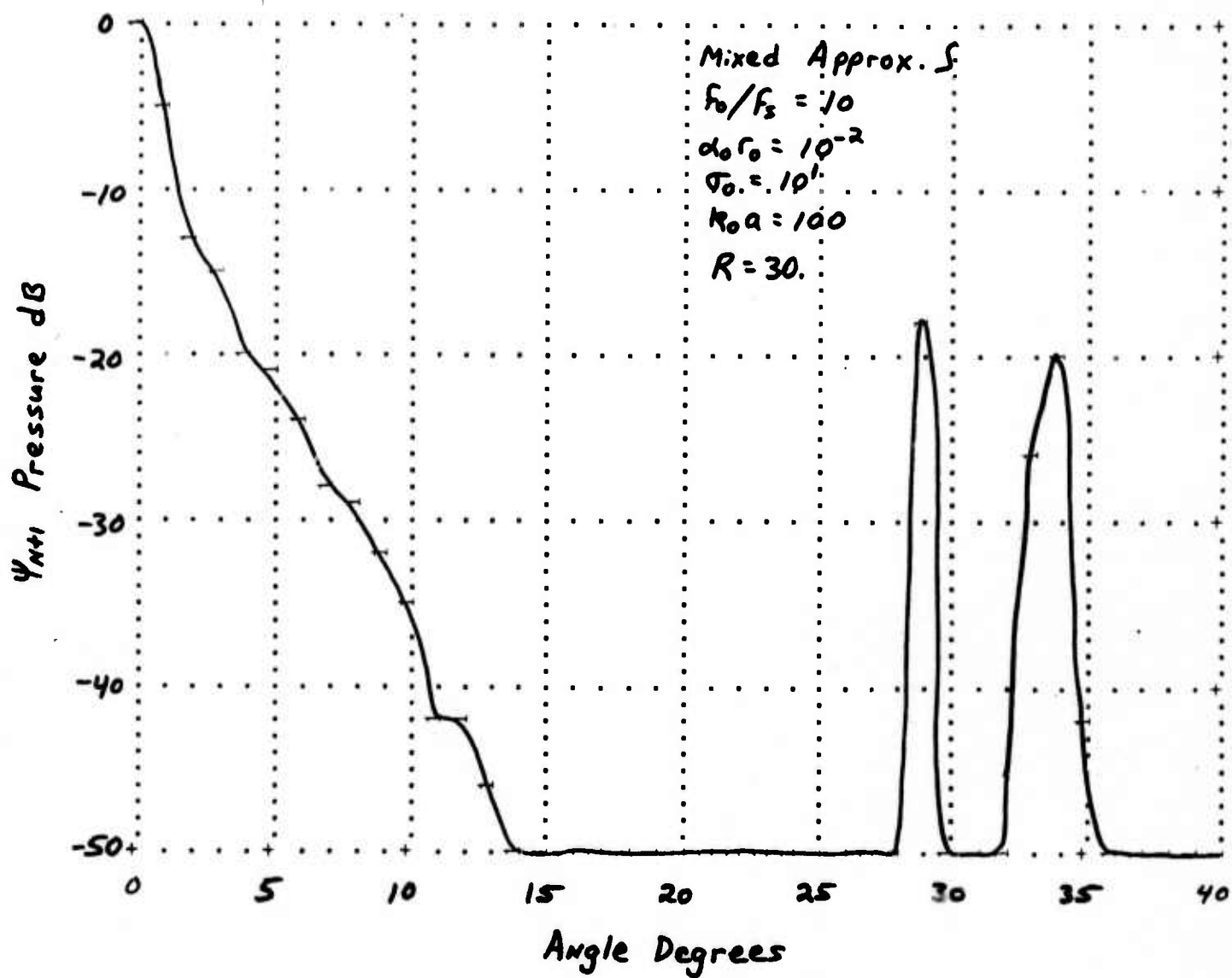
$$\sigma_o = 10$$

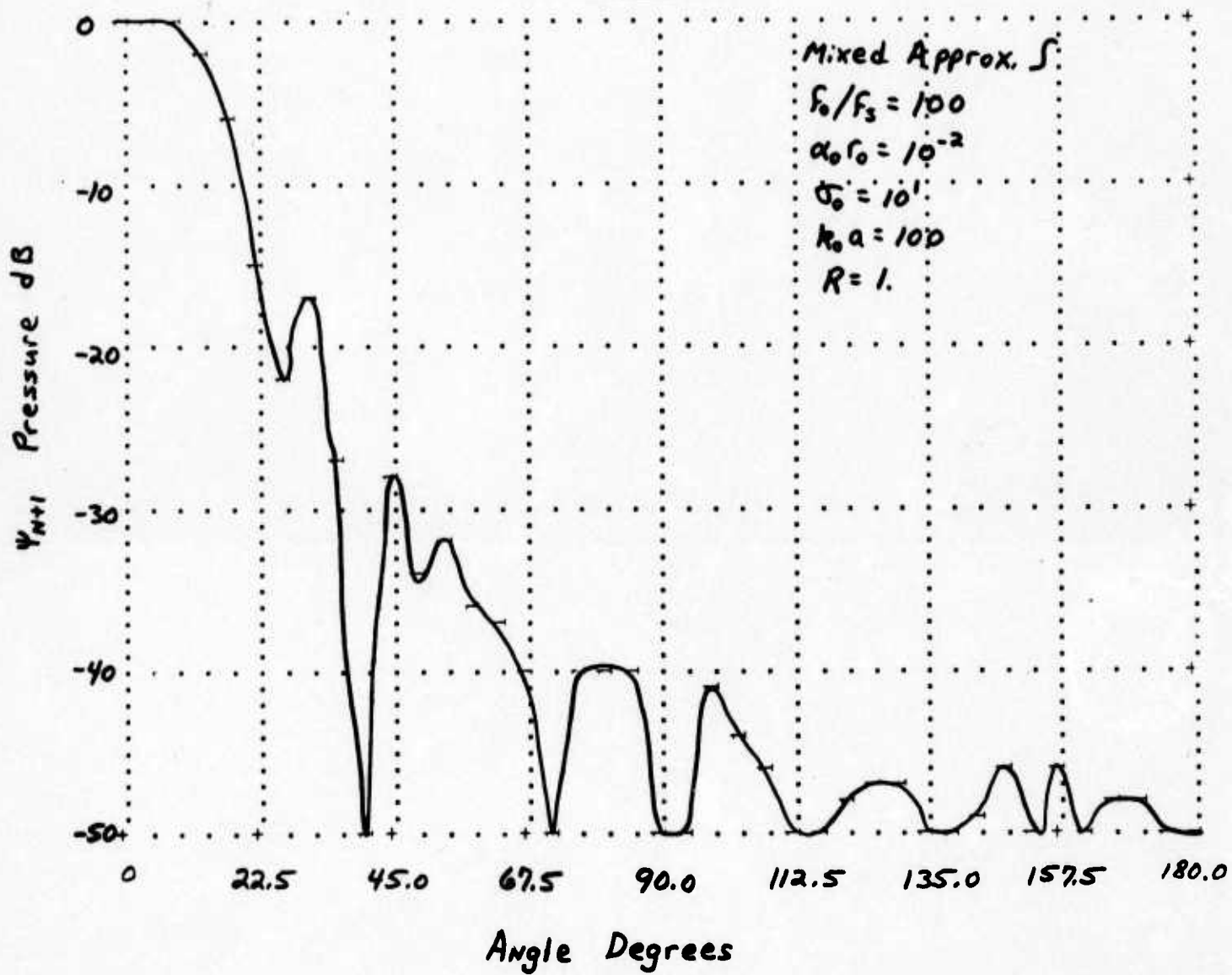
$$f_o/f_s = 10, 100, 1000$$

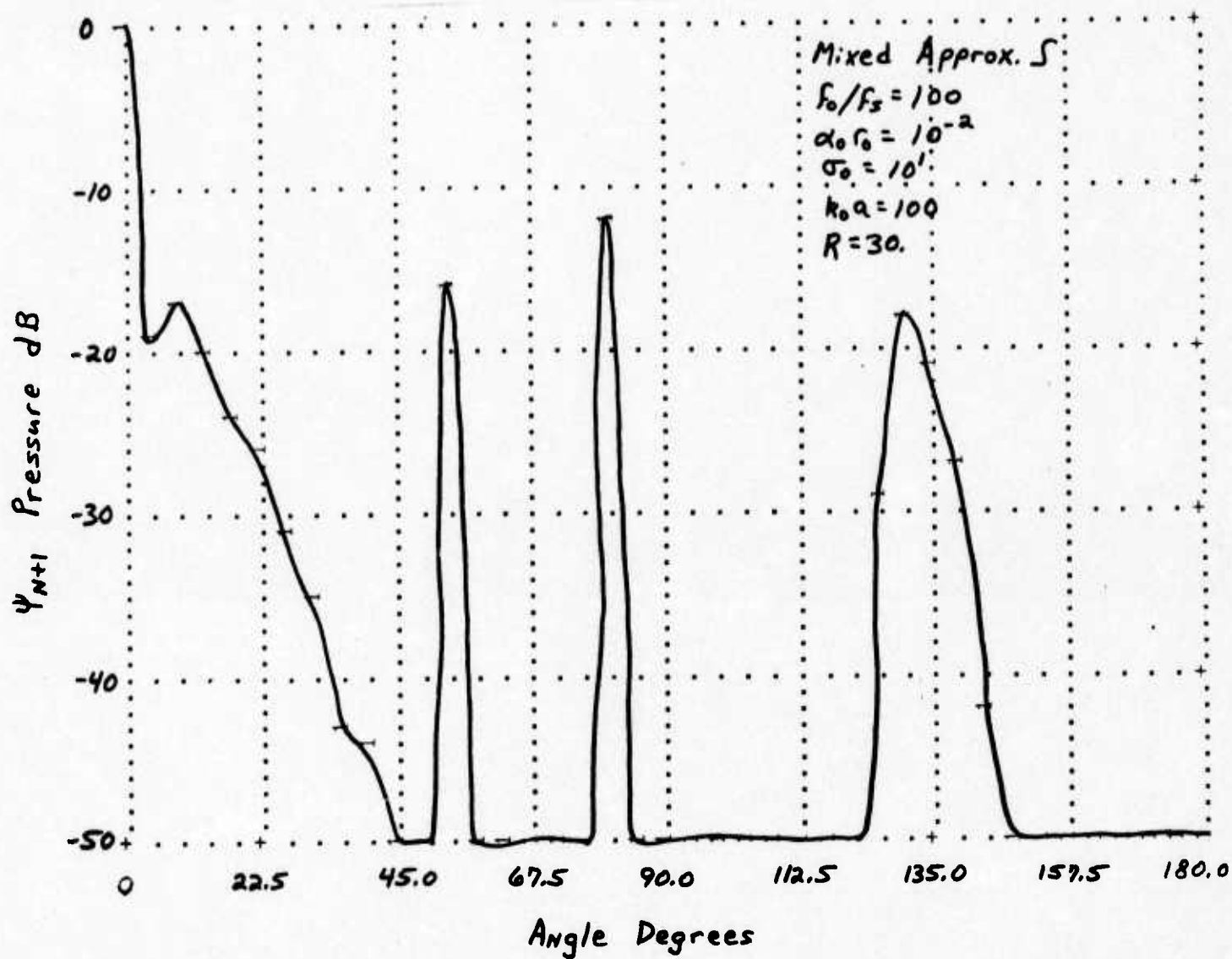
$$R = 1, 30$$

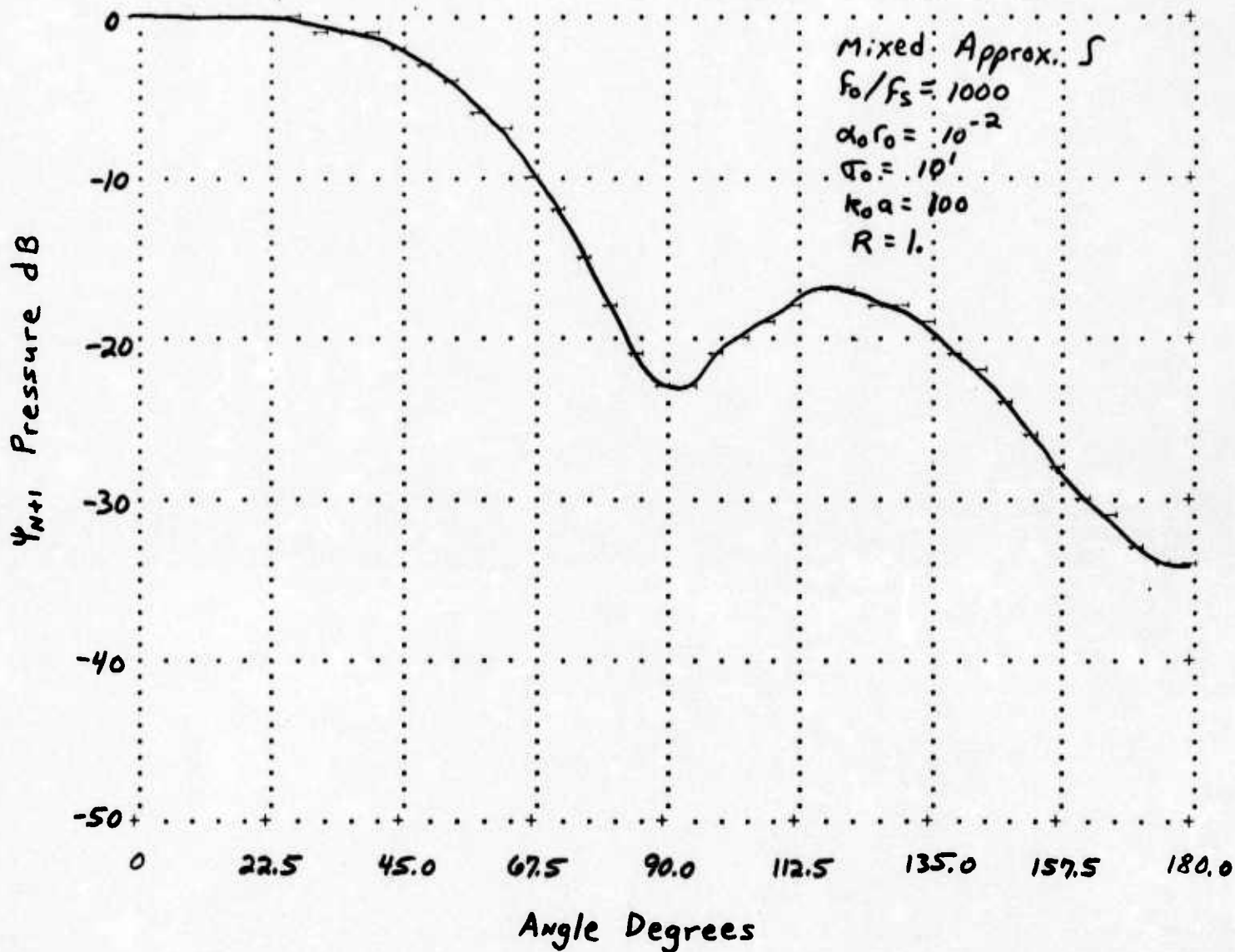
$$\psi_{NFR+1}$$











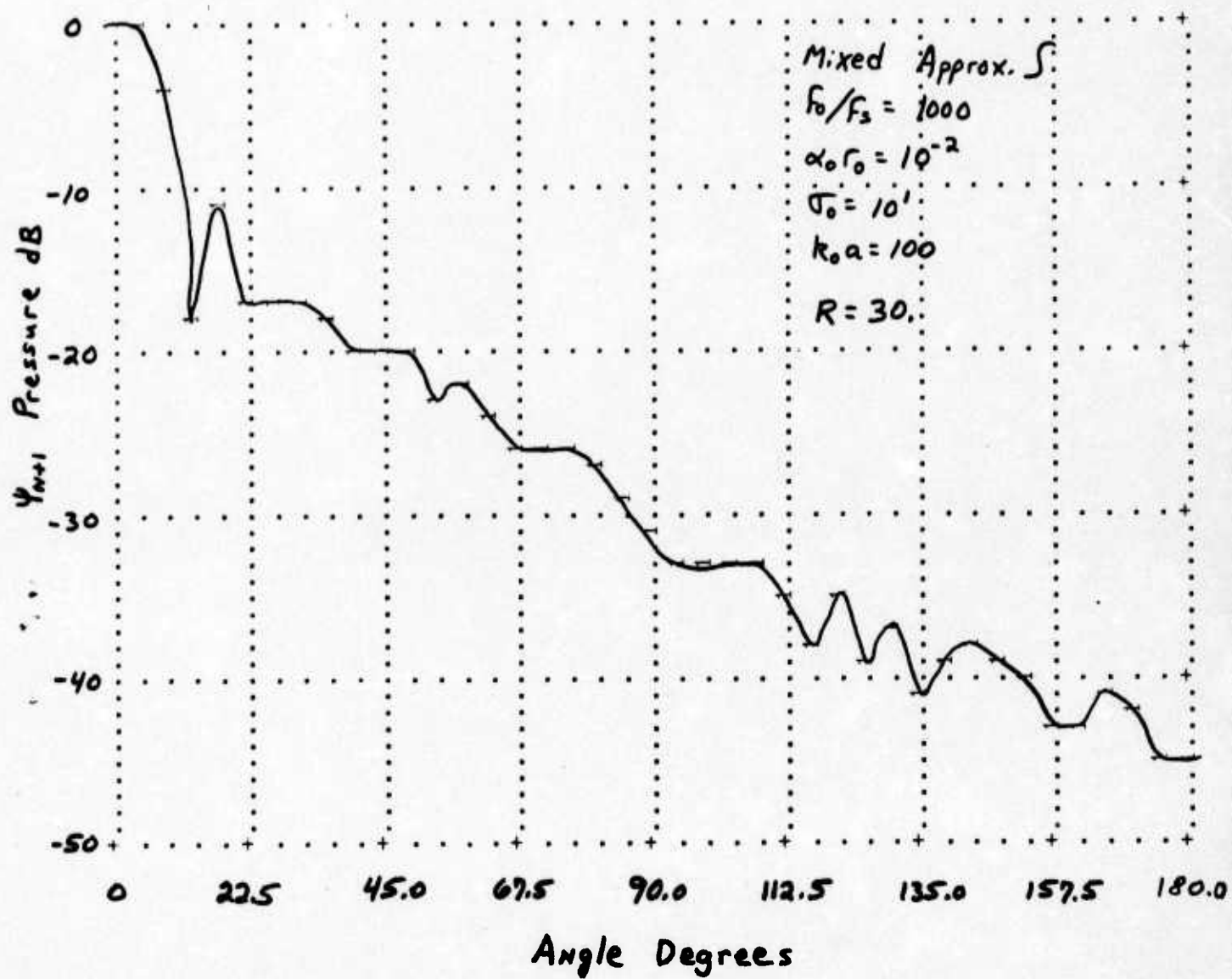


Figure 14
(pages F65to F66)

Beam patterns

Mixed approximate integral case

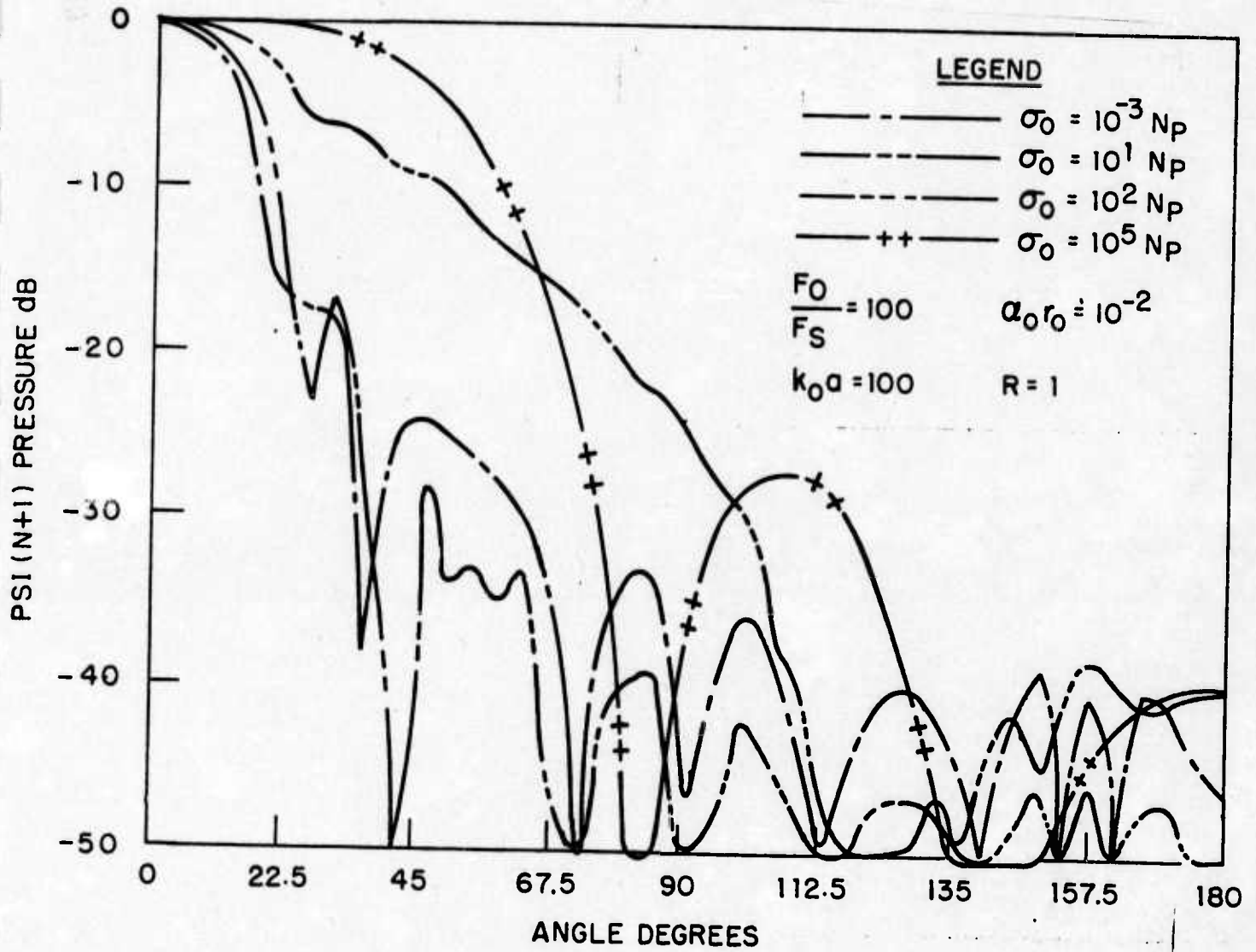
$$\alpha_o r_o = 10^{-2}$$

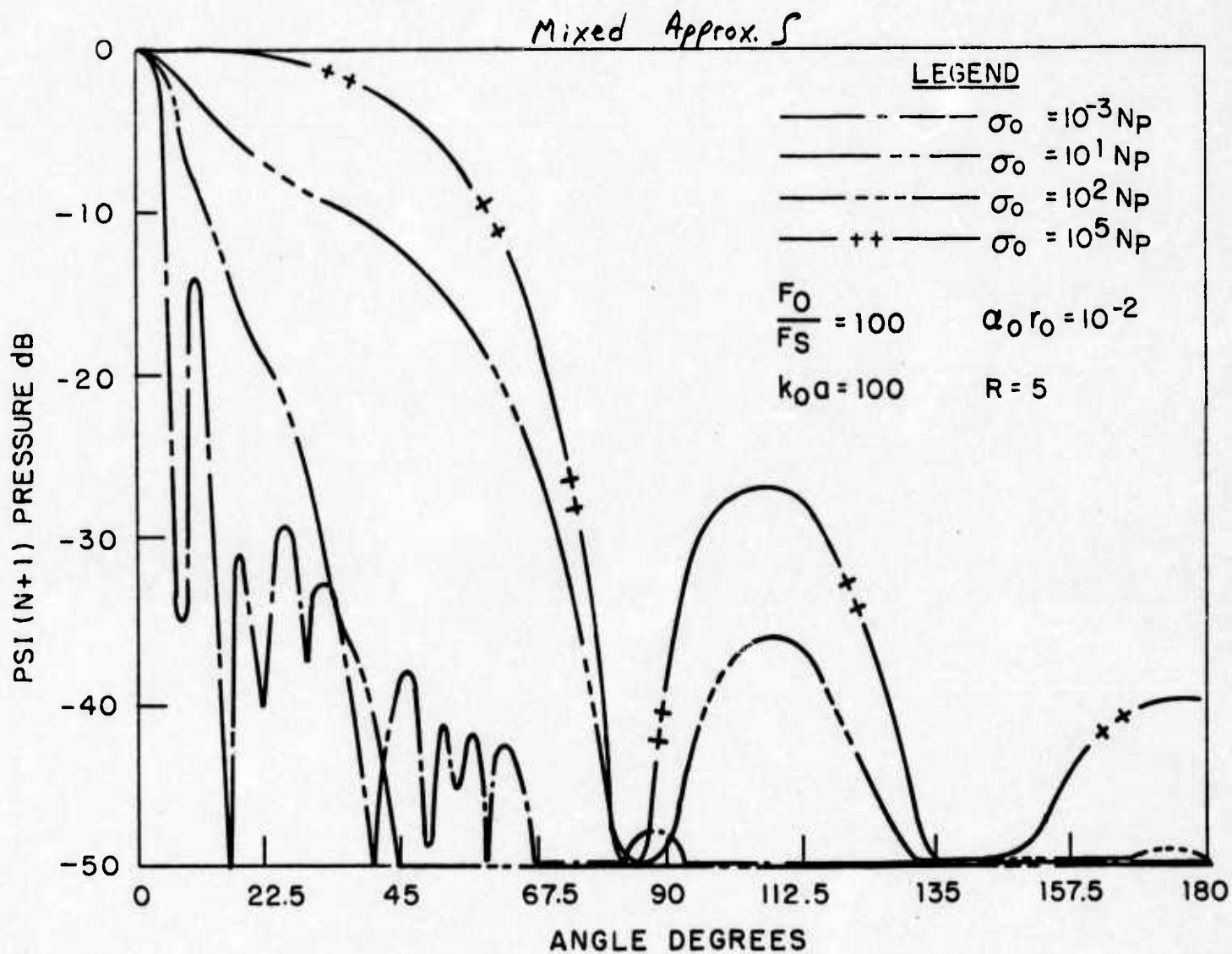
$$\sigma_o = 10^{-3}, 10^1, 10^2, 10^5$$

$$R = 1, 5$$

$$\psi_{NFR+1}$$

Mixed Approx. S





PROGRAM LISTING

```

1000    DOUBLE PRECISION R,Y(120),H,DELT,TFINAL,YN(120),TFMD,XMSG,
1010    1 PLAST
1020    DIMENSION V(60),W(60),KD(1),EP(1),KQ(120),DT(10,120),
1030    1 F(120),KS(60),AR(5),SG(5),RG00D(60),YG00D(60),YLAST(60),
1040    2 IYN(60)
1050    COMMON /BLK1/XP, XD,PS1,PS2,XDB1,IX,IFLAG,M,PSN1,PSN2,
1060    1 PLO,XMAX,NSC,XT,PAT,TMAX
1070    DIMENSION XP(22,5),XD(22,5),PS1(22,5),PS2(22,5),XDB1(22,5),
1080    1 PSN1(22,5),PSN2(22,5),XT(41,5),PAT(41,5)
1090    COMMON /BLK2/IDPD,DDT,TR,PLAST,RG00D,YG00D,YLAST,
1100    1 IYN,SWK
1110    DIMENSION PLX(22),PLY(22),PLXT(41),PLYT(41)
1120    DIMENSION ICHR(5),LBXD(2),LBXP(4),LBXD(4),LBX1(2),LBX2(2),
1130    1 LBXM(3),LBX3(3),LBX4(3),LBXT(4),LBXT(5)
1140    COMMON N,FRAT,AUR0,SIG0,F,LPS,KS,NHL,NHL21,NFR,NHL2,LUP,
1150    1 XMSG,IAP,KS1,KS2,AK0A,IBM,RBM
1160    1000 FORMAT (I)
1170    1010 FORMAT (7F5H LUP=,I3,2X,5H LPS=,I3,2X,5H NFR=,I3,2X,5H NHL=,
1180    1 I3,2X,5H NHL21=,I3,2X,5H NHL2=,I3,2X,5H NSG=,I3,7F5H LAP=,I3,
1190    2 2X,5H NSC=,I3,2X,5H IBM=,I3)
1200    1020 FORMAT (5H NDC=,I3)
1210    1030 FORMAT (5H PS0=,E10.4,2X,5H LPP=,I3)
1220    1040 FORMAT (6H XMAX=,E10.4)
1230    1050 FORMAT (6H AUR0=,5(E10.4,3X))
1240    1060 FORMAT (6H SIG0=,5(E10.4,3X))
1250    1070 FORMAT (6H IDPD=,E10.4,3X,6H PRES=,E10.4,3X,6H TEMD=,F6.1,3X,
1260    1 5H ITR=,I3,3X,7F5H SWK=,E10.4,3X,6H KQ0A=,E10.4)
1270    1080 FORMAT (5H IAP=,I3)
1280    1090 FORMAT (5H RBM=,E10.4,3X,5H TMAX=,F7.2)
1290    2010 FORMAT (4H EP=,E10.4,3X,7H HMINA=,E10.4)
1300    2020 FORMAT (25H ERROR IN CONTROL PARAMETERS)
1310    2030 FORMAT (40H APPROX. EQNS. NOT VALID FOR DIFF. FREQ.)
1320    2040 FORMAT (41H APPROX. EQNS. ONLY VALID FOR MIXED CASES)
1330    READ 1000,NDC
1340    PRINT 1020,NDC
1350    DO 400 I=1,NDC
1360    READ 1000,LUP,LPS,NFR,NHL,NHL21,NFR,NHL2,LUP,NSC,IBM
1370    PRINT 1010,LUP,LPS,NFR,NHL,NHL21,NFR,NHL2,LUP,NSC,IBM
1380 C    LUP=1 FOR HARMONIC STUDIES,=2 FOR UP-CONVERSION,=3 FOR DIFF.
1390 C    FREQ.
1400 C    LPS=0 FOR PLANE WAVE, =1 FOR SPHERICAL, =2 FOR COMBINED PHASE
1410 C    =3 FOR COMBINED NO PHASE
1420 C    LAP=1 IF WANT APPROXIMATE EQUATIONS,=0 OTHERWISE
1430 C    NSC=0 FOR NO SIG0 SCALE,=1 FOR SIG0 SCALE
1440 C    IBM=1 FOR BEAM PATTERN,=0 OTHERWISE
1450 C    IBM=1 ONLY FOR LUP=2 AND LAP=1
1460 C    INPUT CAN BE ORGANIZED BETTER
1470    IF (NSG .GT. 5) NSG=5
1480    READ 1000,XMAX
1490    PRINT 1040,XMAX
1500    READ 1000,(AR(L),L=1,NAR)
1510    PRINT 1050,(AR(L),L=1,NAR)
1520    READ 1000,(SG(L),L=1,NSG)

```

**COPY AVAILABLE TO CDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION**

```

1530      PRINT 1060, (SG(L), L=1, NSG)
1540      READ 1000, SALN, PRES, TEMC, ITA, SMK, AK0A
1550      PRINT 1070, SALN, PRES, TEMC, ITA, SMK, AK0A
1560 C    SALN=SALINITY IN PARTS PER 1000, =0. FOR FRESH WATER
1570 C    PRES=PRESSURE IN ATMOSPHERES
1580 C    TEMC=TEMPERATURE IN DEGREES CENTIGRADE
1590 C    ITA=1 FOR MARSH+SCHULKIN FORM, =2 FOR RUSSIAN FORM
1600 C    SMK=SEA WATER K, AK0A=K0+A
1610      TDPD=0.69145*SALN
1620      DOT=0.538E-3*(1.0-6.54E-4*PRES)
1630      IF (ITA .EQ. 1) TA=7.267E-12*10.0**((1520.0/(TEMC+273.0))
1640      IF (ITA .EQ. 2) TA=(1.0E-3/755.0)/(TEMC+273.0)*10.0**((934.0/
1650      1 (TEMC+273.0))
1660      IF (LAP .NE. 1) GO TO 118
1670      READ 1000, IAP
1680      PRINT 1080, IAP
1690 C    IAP=1 FOR 1ST APPROX. FORM, =2 FOR 2ND APPROX. FORM
1700      IF (IBM .NE. 1) GO TO 205
1710      READ 1000, RBM, TMAX
1720      PRINT 1090, RBM, TMAX
1730      205 CONTINUE
1740      IF (LPS .GE. 2) GO TO 204
1750      PRINT 2040
1760      GO TO 999
1770      204 IF (LUP .NE. 3) GO TO 118
1780      PRINT 2030
1790      GO TO 999
1800      118 CONTINUE
1810      IF (LUP .NE. 2) GO TO 101
1820      READ 1000, PSO, LPP
1830      PRINT 1030, PSO, LPP
1840 C    LPP=1 FOR ONLY N-1 CURVE, =2 FOR ONLY N+1 CURVE, =3 FOR BOTH
1850      101 CONTINUE
1860      ICHR(1)='1'
1870      ICHR(2)='2'
1880      ICHR(3)='3'
1890      ICHR(4)='4'
1900      ICHR(5)='5'
1910      LBXD(1)=1
1920      LBXD(2)='9'
1930      IF (LPS .NE. 0) GO TO 114
1940      IF (NSC .EQ. 0) GO TO 201
1950      LBXP(1)=6
1960      LBXP(2)='SIG0+R'
1970      GO TO 115
1980      201 LBXP(1)=1
1990      LBXP(2)='R'
2000      GO TO 115
2010      114 IF (LPS .NE. 1) GO TO 116
2020      IF (NSC .EQ. 0) GO TO 202
2030      LBXP(1)=10
2040      LBXP(2)='SIG0+L'
2050      LBXP(3)='N(R)'

```

COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION

```

2060      GO TO 115
2070 202 LBXP(1)=5
2080      LBXP(2)='LN(R)'
2090      GO TO 115
2100 116 IF (NSC .EQ. 0) GO TO 203
2110      LBXP(1)=15
2120      LBXP(2)='SIGO♦A'
2130      LBXP(3)='ARCSINH'
2140      LBXP(4)='(R)'
2150      GO TO 115
2160 203 LBXP(1)=10
2170      LBXP(2)='ARCSIN'
2180      LBXP(3)='H(R)'
2190 115 CONTINUE
2200      LBYD(1)=13
2210      LBYD(2)='EXTRA'
2220      LBYD(3)='DB LOS'
2230      LBYD(4)='S'
2240      LBY1(1)=4
2250      LBY1(2)='PSI1'
2260      LBY2(1)=4
2270      LBY2(2)='PSI2'
2280      LBYM(1)=11
2290      LBYM(2)='DIFF.'
2300      LBYM(3)='FREQ.'
2310      LBY3(1)=8
2320      LBY3(2)='PSI(N-'
2330      LBY3(3)='1)'
2340      LBY4(1)=8
2350      LBY4(2)='PSI(N+'
2360      LBY4(3)='1)'
2370      IF (IBM .NE. 1) GO TO 120
2380      LBXT(1)=13
2390      LBXT(2)='ANGLE'
2400      LBXT(3)='DEGREE'
2410      LBXT(4)='S'
2420      LBYT(1)=20
2430      LBYT(2)='PSI(N+'
2440      LBYT(3)='1) PRE'
2450      LBYT(4)='SSURE'
2460      LBYT(5)='DB'
2470 120 CONTINUE
2480      DO 350 L=1,NAP
2490      AOR0=AR(L)
2500      DO 351 M=1,NSG
2510      SIG0=SG(M)
2520      XMSG=XMAX/SIG0
2530      IF (NSC .EQ. 0) XMSG=XMAX
2540      FRAT=1.0/NFR
2550      F2AP=FRAT♦♦2 ♦AOR0
2560      IF (LUP .NE. 2) GO TO 102
2570      AKOR0=(AKOR♦♦2)/2.0
2580      FKR2=2.0♦FRAT♦AKOR0

```

```

2590      THET=0.0
2600      ST22=(SIN(THET/2.0))**2
2610 102 CONTINUE
2620      IF (LAP .NE. 1) GO TO 119
2630      CALL APPROX
2640      GO TO 351
2650 119 CONTINUE
2660      NHL2=2*NHL
2670      NHL21=NHL2+1
2680      IF (NFR .LE. NHL2) GO TO 102
2690      N=NHL+NHL*NHL21
2700      KSN=NHL2+1
2710      DO 109 J=1,N
2720      NLJ=NHL+1+J
2730      JMN=MOD(NLJ,NHL21)
2740      IF (JMN .EQ. 0) JMN=NHL21
2750      QJN=(NLJ-0.5)/NHL21
2760      JDN=INT(QJN)
2770      KR=JDN*NFR+(JMN-NHL-1)
2780      KS(J)=KR
2790 109 CONTINUE
2800      GO TO 110
2810 103 N=NHL*NFR+NHL
2820      KSN=NFR
2830      DO 111 J=1,N
2840 111 KS(J)=J
2850 110 CONTINUE
2860      KSN1=KSN-1
2870      KSN2=KSN+1
2880      NEQ=2*N
2890      DO 105 K=1,N
2900      V(K)=0.
2910      W(K)=0.
2920 105 CONTINUE
2930      IF (LUP .NE. 2) GO TO 103
2940      V(KSN)=0.0
2950      W(KSN)=1.0
2960      GO TO 112
2970 103 IF (LUP .EQ. 1) W(1)=1.0
2980      IF (LUP .NE. 3) GO TO 112
2990      W(KSN)=0.5
3000      W(KSN1)=0.5
3010 C      ASSUME PUMP FREDS. IN NFR-1,NFR AND DIFF. FREQ. IN 1
3020 112 CONTINUE
3030      KD(1)=1
3040      NHM=0
3050      IF (LPS .NE. 0) GO TO 106
3060      P=0.000
3070      RLD=0.0
3080      IF (LUP .EQ. 2) W(1)=PS0
3090      TFINAL=XMSG
3100      EP(1)=1.0E-3
3110      H=TFINAL*1.00-2

```

```

3120 HMINA=TFINAL*1.0E-4
3130 HMAXA=TFINAL/2.0
3140 DELT=TFINAL/20.0
3150 MXSTEP=20
3160 TFMD=TFINAL-HMINA
3170 GO TO 107
3180 106 IF (LPS .GE. 2) GO TO 117
3190 R=1.000
3200 RLD=1.0
3210 IF (LUP .EQ. 2) W(1)=PS0*EXP(-FRAT**2 *AOR0)
3220 TFINAL=DEXP(XMSG)
3230 EP(1)=1.0E-3
3240 H=TFINAL*1.0D-2
3250 HMINA=TFINAL*1.0E-4
3260 HMAXA=TFINAL/2.0
3270 DELT=DEXP(XMSG/20.000)-1.000
3280 MXSTEP=20
3290 TFMD=TFINAL-HMINA
3300 GO TO 107
3310 117 R=1.0D-3
3320 RLD=0.0
3330 IF (LUP .EQ. 2) W(1)=PS0*EXP(-FRAT**2 *AOR0)
3340 TFINAL=DEXP(XMSG)/2.000
3350 EP(1)=1.0E-3
3360 H=TFINAL*1.0D-2
3370 HMINA=TFINAL*1.0E-4
3380 HMAXA=TFINAL/2.0
3390 DELT=0.5D0*(DEXP(XMSG/20.000)-1.000)
3400 MXSTEP=20
3410 TFMD=TFINAL-HMINA
3420 107 CONTINUE
3430 DO 104 KI=1,N
3440 Y(2*KI-1)=V(KI)
3450 Y(2*KI)=W(KI)
3460 IYN(KI)=0
3470 104 CONTINUE
3480 IX=0
3490 CALL VODO(NEO,R,Y,F,KD,EP,IFLAG,H,HMINA,HMAXA,DELT,TFINAL,
3500 1 MXSTEP,KSTEP,KEMAX,EMAX,KD,YN,DT)
3510 GO TO 6
3520 4 CALL VODO1
3530 6 GO TO (10,10,30,40,50,60,70,80),IFLAG
3540 10 CONTINUE
3550 IF (LUP .NE. 2) GO TO 113
3560 PEFR=PS0*EXP(-F2AR*R)
3570 FRST=FKR2*R*ST22
3580 Y(1)=PEFR*SIN(FRST)
3590 Y(2)=PEFR*COS(FRST)
3600 113 CONTINUE
3610 CALL DERIV(R,Y)
3620 GO TO 4
3630 30 IF (R .GE. TFMD) GO TO 200
3640 CALL OUTPUT(R,Y)

```

**COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION**


```

3650      IF (IX .GE. 22) GO TO 300
3660      IF (LPS .EQ. 0) GO TO 4
3670      IF (LPS .EQ. 1) DELT=R*(DEXP(XMSG/20.000)-1.000)
3680      IF (LPS .GE. 2) DELT=(R+0.500)*(DEXP(XMSG/20.000)-1.000)
3690      GO TO 4
3700      40 GO TO 200
3710      50 CALL OUTPUT(R,Y)
3720      GO TO 4
3730      60 EP(1)=32.*EMAX*EP(1)
3740      GO TO 4
3750      70 HMINA=HMINA/10.0
3760      PRINT 2010,EP(1),HMINA
3770      TEND=TFINAL-HMINA
3780      NHM=NHM+1
3790      IF (NHM .LE. 3) GO TO 4
3800      GO TO 300
3810      80 PRINT 2020
3820      GO TO 999
3830      200 CALL OUTPUT(R,Y)
3840      300 CONTINUE
3850      351 CONTINUE
3860      IF (IBM .NE. 1) GO TO 360
3880      CALL TSETUP(0.0,TMAX,-50.0,0.0,1HE,1HG,LBXT,LBYT)
3890      DO 361 M=1,NSG
3900      DO 362 K=1,41
3910      PLXT(K)=XT(K,M)
3920      362 PLYT(K)=PAT(K,M)
3930      IF (M .EQ. NSG) GO TO 363
3940      CALL TPLOT(.FALSE.,ICHR(M),41,PLXT,PLYT)
3950      GO TO 361
3960      363 CALL TPLOT(.TRUE.,ICHR(M),41,PLXT,PLYT)
3970      361 CONTINUE
3980      GO TO 350
3990      360 IF (LUP .NE. 1) GO TO 315
4000      CALL TSETUP(0.0,XMAX,0.0,1.0,1HE,1HG,LBXP,LBY1)
4010      DO 301 M=1,NSG
4020      DO 302 K=1,IX
4030      PLX(K)=XP(K,M)
4040      302 PLY(K)=PS1(K,M)
4050      IF (M .EQ. NSG) GO TO 303
4060      CALL TPLOT(.FALSE.,ICHR(M),IX,PLX,PLY)
4070      GO TO 301
4080      303 CALL TPLOT(.TRUE.,ICHR(M),IX,PLX,PLY)
4090      301 CONTINUE
4100      IF (LAP .EQ. 1) GO TO 317
4110      DO 304 M=1,NSG
4120      DO 305 K=1,IX
4130      PLX(K)=XP(K,M)
4140      305 PLY(K)=PS2(K,M)
4150      IF (M .GT. 1) GO TO 306
4160      CALL TSCALE(PLY,IX,VLOW,VHIGH,.TRUE.,1)
4170      GO TO 304

```

COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION

```

4180 306 CALL TSCALE(PLY,IX,VLOW,VHIGH,.FALSE.,1)
4190 304 CONTINUE
4200 CALL TSETUP(0.0,XMAX,0.0,VHIGH,1HE,1HG,LBXP,LBY2)
4210 DO 307 M=1,NSG
4220 DO 314 K=1,IX
4230 PLX(K)=XP(K,M)
4240 314 PLY(K)=PS2(K,M)
4250 IF (M.EQ. NSG) GO TO 308
4260 CALL TPLOT(.FALSE.,ICHR(M),IX,PLX,PLY)
4270 GO TO 307
4280 308 CALL TPLOT(.TRUE.,ICHR(M),IX,PLX,PLY)
4290 307 CONTINUE
4300 317 DO 312 M=1,NSG
4310 DO 310 K=1,IX
4320 PLX(K)=XD(K,M)
4330 310 PLY(K)=XDB1(K,M)
4340 PLY(1)=PLY(2)
4350 IF (M.GT. 1) GO TO 311
4360 CALL TSCALE(PLY,IX,VLOW,VHIGH,.TRUE.,1)
4370 CALL TSCALE(PLX,IX,RLOW,RHIGH,.TRUE.,0)
4380 GO TO 312
4390 311 CALL TSCALE(PLY,IX,VLOW,VHIGH,.FALSE.,1)
4400 CALL TSCALE(PLX,IX,RLOW,RHIGH,.FALSE.,0)
4410 312 CONTINUE
4420 CALL TSETUP(RLO,RHIGH,VLOW,VHIGH,1HE,1HG,LBXD,LBYD)
4430 DO 309 M=1,NSG
4440 DO 316 K=1,IX
4450 PLX(K)=XD(K,M)
4460 316 PLY(K)=XDB1(K,M)
4470 PLY(1)=PLY(2)
4480 IF (M.EQ. NSG) GO TO 313
4490 CALL TPLOT(.FALSE.,ICHR(M),IX,PLX,PLY)
4500 GO TO 309
4510 313 CALL TPLOT(.TRUE.,ICHR(M),IX,PLX,PLY)
4520 309 CONTINUE
4530 GO TO 350
4540 315 IF (LUP.NE. 2) GO TO 320
4550 IF (LPP.EQ. 2) GO TO 337
4560 DO 331 M=1,NSG
4570 DO 332 K=1,IX
4580 PLX(K)=XD(K,M)
4590 332 PLY(K)=PSN1(K,M)
4600 IF (M.GT. 1) GO TO 333
4610 CALL TSCALE(PLY,IX,VLOW,VHIGH,.TRUE.,1)
4620 CALL TSCALE(PLX,IX,RLOW,RHIGH,.TRUE.,0)
4630 GO TO 331
4640 333 CALL TSCALE(PLY,IX,VLOW,VHIGH,.FALSE.,1)
4650 CALL TSCALE(PLX,IX,RLOW,RHIGH,.FALSE.,0)
4660 331 CONTINUE
4670 CALL TSETUP(PLD,RHIGH,0.0,VHIGH,1HE,1HG,LBXD,LBY3)
4680 DO 334 M=1,NSG
4690 DO 335 K=1,IX
4700 PLX(K)=XD(K,M)

```



```

4710 335 PLY(K)=PSN1(K,M)
4720 IF (M.EQ. NSG) GO TO 336
4730 CALL TPLOT(.FALSE., ICHR(M), IX, PLX, PLY)
4740 GO TO 334
4750 336 CALL TPLOT(.TRUE., ICHR(M), IX, PLX, PLY)
4760 334 CONTINUE
4770 337 IF (LPP.EQ. 1) GO TO 350
4780 DO 341 M=1, NSG
4790 DO 342 K=1, IX
4800 PLX(K)=XD(K,M)
4810 342 PLY(K)=PSN2(K,M)
4820 IF (M.GT. 1) GO TO 343
4830 CALL TSCALE(PLY, IX, VLOW, VHIGH, .TRUE., 1)
4840 CALL TSCALE(PLX, IX, PLOW, RHIGH, .TRUE., 0)
4850 GO TO 341
4860 343 CALL TSCALE(PLY, IX, VLOW, VHIGH, .FALSE., 1)
4870 CALL TSCALE(PLX, IX, PLOW, RHIGH, .FALSE., 0)
4880 341 CONTINUE
4890 CALL TSETUP(RLO, RHIGH, 0.0, VHIGH, 1HE, 1HG, LBXD, LBY4)
4900 DO 344 M=1, NSG
4910 DO 345 K=1, IX
4920 PLX(K)=XD(K,M)
4930 345 PLY(K)=PSN2(K,M)
4940 IF (M.EQ. NSG) GO TO 346
4950 CALL TPLOT(.FALSE., ICHR(M), IX, PLX, PLY)
4960 GO TO 344
4970 346 CALL TPLOT(.TRUE., ICHR(M), IX, PLX, PLY)
4980 344 CONTINUE
4990 GO TO 350
5000 320 IF (LUP.NE. 3) GO TO 350
5010 DO 324 M=1, NSG
5020 DO 322 K=1, IX
5030 PLX(K)=XP(K,M)
5040 322 PLY(K)=PS1(K,M)
5050 IF (M.GT. 1) GO TO 323
5060 CALL TSCALE(PLY, IX, VLOW, VHIGH, .TRUE., 1)
5070 GO TO 324
5080 323 CALL TSCALE(PLY, IX, VLOW, VHIGH, .FALSE., 1)
5090 324 CONTINUE
5100 CALL TSETUP(0.0, XMAX, 0.0, VHIGH, 1HE, 1HG, LBXP, LBYM)
5110 DO 321 M=1, NSG
5120 DO 326 K=1, IX
5130 PLX(K)=XP(K,M)
5140 326 PLY(K)=PS1(K,M)
5150 IF (M.EQ. NSG) GO TO 325
5160 CALL TPLOT(.FALSE., ICHR(M), IX, PLX, PLY)
5170 GO TO 321
5180 325 CALL TPLOT(.TRUE., ICHR(M), IX, PLX, PLY)
5190 321 CONTINUE
5200 350 CONTINUE
5210 400 CONTINUE
5220 999 END
END

```

```

1000      SUBROUTINE APPROX
1010      DIMENSION F(120),KS(60),XP(22,5),XD(22,5),PS1(22,5),
1020      1 PS2(22,5),XDB1(22,5),PSN1(22,5),PSN2(22,5),PSILK(42),
1030      2 PLK(42),XT(41,5),PAT(41,5)
1040      DOUBLE PRECISION DELT0,DELT,R,TFINAL,XMSG,XR,DELT1,
1050      1 DELT2,DELT3,DELT4,DELT1
1060      COMMON /BLK1/XP,XD,PS1,PS2,XDB1,IX,IFLAG,M,PSN1,PSN2,
1070      1 RLO,XMAX,HCC,XT,PAT,IMAX
1080      COMMON /R/PAT,AOR0,SIG0,F,LPS,KS,NHL,NHL21,NFR,NHL2,LUP,
1090      1 XMSG,IAP,KSH,KSH1,KSH2,AKOR,IBM,RBM
1100      IAP=1 FOR 1ST APPROX. FORM, =2 FOR 2ND APPROX. FORM
1110      IX=0
1120      IF (LPS .GE. 2) GO TO 41
1130      IF (LPS .EQ. 1) GO TO 42
1140      R=0.00
1150      RLO=0.0
1160      TFINAL=XMSG
1170      DELT0=TFINAL/20.000
1180      DELT1=TFINAL/1.004
1190      DELT2=TFINAL/1.003
1200      DELT3=TFINAL/1.002
1210      DELT4=TFINAL/1.001
1220      GO TO 43
1230      42 R=1.000
1240      RLO=1.0
1250      TFINAL=DEXP(XMSG)
1260      DELT0=DEXP(XMSG/20.000)-1.000
1270      DELT1=DEXP(XMSG/1.004)-1.000
1280      DELT2=DEXP(XMSG/1.003)-1.000
1290      DELT3=DEXP(XMSG/1.002)-1.000
1300      DELT4=DEXP(XMSG/1.001)-1.000
1310      GO TO 43
1320      41 R=0.00
1330      RLO=0.0
1340      TFINAL=DEXP(XMSG)/2.000
1350      DELT0=DEXP(XMSG/20.000)-1.000
1360      DELT1=DEXP(XMSG/1.004)-1.000
1370      DELT2=DEXP(XMSG/1.003)-1.000
1380      DELT3=DEXP(XMSG/1.002)-1.000
1390      DELT4=DEXP(XMSG/1.001)-1.000
1400      43 IF (LUP .NE. 2) GO TO 30
1410      PRESTORE PSI1 CURVE
1420      THERE ARE OTHER TECHNIQUES FOR EVALUATING THE NESTED INTEGRAL
1430      XX=0.0
1440      IF (LPS .EQ. 1) XX=1.0
1450      DO 100 IR=1,42
1460      XR=XX
1470      PLK(IR)=XX
1480      PSILK(IR)=FPSI(XR,AOR0,SIG0,LPS,IAP)
1490      IF (PSILK(IR) .LE. 1.0E-20) PSILK(IR)=1.0E-20
1500      IF (IR .LE. 10) DELT1=DELT1

```

COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION

```

1510      IF (IR .GT. 10 .AND. IR .LE. 20) DELTI=DELT2
1520      IF (IR .GT. 20 .AND. IR .LE. 30) DELTI=DELT3
1530      IF (IR .GT. 30) DELTI=DELT4
1540      IF (LPS .EQ. 0) DELX=DELT1
1550      IF (LPS .EQ. 1) DELX=XX*DELT1
1560      IF (LPS .GE. 2) DELX=(XX+0.5)*DELT1
1570      XX=XX+DELX
1580 100 CONTINUE
1590      30 IF (IBM .NE. 1) GO TO 20
1600      R=RBM
1610      TMX=TMAX*3.14159/180.0
1620      NTH=41
1630      GO TO 21
1640      20 NTH=1
1650      TMX=0.0
1660      21 DO 200 ITH=1,NTH
1670          THET=(ITH-1)*TMX/40.0
1680          XT(ITH,M)=THET*180.0/3.14159
1690      40 IX=IX+1
1700          IF (IBM .NE. 1) XD(IX,M)=R
1710          IF (LUP .NE. 1) GO TO 50
1720          PSI1=FPSI(R,AOR0,SIG0,LPS,IAP)
1730          IF (PSI1 .LE. 1.0E-20) PSI1=1.0E-20
1740          PS1(IX,M)=PSI1
1750          IF (LPS .GE. 2) GO TO 52
1760          IF (LPS .EQ. 1) GO TO 51
1770          IF (NSC .EQ. 0) XP(IX,M)=R
1780          IF (NSC .EQ. 1) XP(IX,M)=SIG0*R
1790          XDB1(IX,M)=-20.0*ALOG10(PSI1)-8.686*AOR0*R
1800          GO TO 60
1810      51 IF (NSC .EQ. 0) XP(IX,M)=DLOG(R)
1820          IF (NSC .EQ. 1) XP(IX,M)=SIG0*DLOG(R)
1830          XDB1(IX,M)=-20.0*ALOG10(PSI1)-8.686*(DLOG(R)+AOR0*
1840      1 (R-1.0D0))
1850          GO TO 60
1860      52 IF (NSC .EQ. 0) XP(IX,M)=DLOG(R+DSORT(R**2+1.0D0))
1870          IF (NSC .EQ. 1) XP(IX,M)=SIG0*DLOG(R+DSORT(R**2+1.0D0))
1880          XDB1(IX,M)=-20.0*ALOG10(PSI1)-8.686*(AOR0*R+
1890      1 DLOG(DSORT(1.0D0+R**2)))
1900          GO TO 60
1910      50 IF (LUP .NE. 2) GO TO 60
1920          APRR=-R*AOR0*(1.0+FRAT)**2
1930          IF (LPS .EQ. 0) DPSM=1.0
1940          IF (LPS .EQ. 1) DPSM=R
1950          IF (LPS .GE. 2) DPSM=SQRT(1.0+R**2)
1960          COEF=(SIG0/2.0)*EXP(APRR)/DPSM
1970          THET2=THET/2.0
1980          SCK=2.0*FRAT*(AK0R**2)/2.0 *SIN(THET2)**2
1990          IF (R .LE. 0.0D0) GO TO 61
2000          A=0.0
2010          IF (LPS .EQ. 1) A=1.0

```

```

2020      B=P
2030      HSTAR=(B-A)*1.0E-2
2040      HMIN=(B-A)*1.0E-4
2050      HMAX=(B-A)/2.0
2060      ERMAX=1.0E-3
2070      KEY=0
2080      CALL ROMBS(A,B,X,FOFX,HSTAR,HMIN,HMAX,ERMAX,ANS,KSTOP,KEY)
2090      62 IF (LPS .EQ. 0) DNI=1.0
2100      IF (LPS .EQ. 1) DNI=X
2110      IF (LPS .GE. 2) DNI=SQRT(1.0+X**2)
2120      IF (X .GT. RLK(1)) GO TO 104
2130      PSIC=PSILK(1)
2140      GO TO 103
2150      104 PM=PSILK(1)
2160      RM=RLK(1)
2170      PSIC=1.0E-20
2180      DO 101 IR=2,42
2190      IF (X .GT. RLK(IR)) GO TO 102
2200      RP=RLK(IR)
2210      PP=PSILK(IR)
2220      PSIC=((X-RM)/(RP-RM))*(PP-PM)+PM
2230      GO TO 103
2240      102 RM=RLK(IR)
2250      PM=PSILK(IR)
2260      IF (IR .LT. 42) GO TO 101
2270      PSIC=PSILK(42)
2280      GO TO 103
2290      101 CONTINUE
2300      103 CONTINUE
2310      FOFX=DNI*PSIC*COS(SCK*X)*EXP(AOR0*X)
2320      CALL ROM2
2330      IF (KSTOP .EQ. 1) GO TO 62
2340      GO TO 63
2350      61 ANS=0.0
2360      63 PPR=COEF*ANS
2370      IF (IBM .NE. 1) PSN2(IX,M)=ABS(PPR)
2380      IF (IBM .NE. 1) GO TO 60
2390      IF (ITH .EQ. 1) PAT0=ABS(PPR)
2400      PAT(ITH,M)=20.0*ALOG10(ABS(PPR)/PAT0)
2410      IF (PAT(ITH,M) .LT. -50.0) PAT(ITH,M)=-50.0
2420      GO TO 200
2430      60 CONTINUE
2440      IF (LPS .EQ. 0) DELT=DELTO
2450      IF (LPS .EQ. 1) DELT=R*DELTO
2460      IF (LPS .GE. 2) DELT=(R+0.500)*DELTO
2470      R=R+DELT
2480      IF (IX .GE. 22) GO TO 200
2490      IF (R .LE. TFINAL) GO TO 40
2500      200 CONTINUE
2510      RETURN
2520      END
99END

```

COPY AVAILABLE TO ONE DOES NOT
 PERMIT FULLY LEGIBLE PRODUCTION

```

1000      SUBROUTINE DEPRIV4 Y0
1010      DOUBLE PRECISION R, X(120), R21, RR21, R211, RI, RLAST, XMSG
1020      COMMON N, FRAT, AOR0, SIG0, F, LPS, KS, NHL, NHL21, NFR, NHL2, LUP,
1030      1 XMSG, IAP, KIN, KSN1, KSN2, AK0A, IBM, PEM
1040      COMMON /BLK2/TDPD, DDT, TA, RLAST, RGOOD, YGOOD, YLAST, IYN,
1050      1 SWK
1060      DIMENSION F(120), KS(60), RGOOD(60), YGOOD(60), YLAST(60),
1070      1 IYN(60)
1080      1010 FORMAT (1X,4H KI=,I3,2X,3H R=,E10.4)
1090 C      LPS=0 FOR PLANE WAVE,=1 FOR SPHERICAL,=2 FOR COMBINED PHASE,
1100 C      =3 FOR COMBINED NO PHASE
1110      IF (LPS .GE. 2 .OR. LUP .EQ. 2) GO TO 14
1120      DO 15 KI=2,N
1130      IF (Y(2*KI) .GE. 0.00) GO TO 16
1140      IF (IYN(KI) .NE. 1) PRINT 1010,KI,R
1150      IYN(KI)=1
1160      RGOOD(KI)=RLAST
1170      YGOOD(KI)=YLAST(KI)
1180      GO TO 17
1190      16 YLAST(KI)=Y(2*KI)
1200      17 IF (IYN(KI) .NE. 1) GO TO 15
1210      IF (Y(2*KI) .GE. 0.00) GO TO 15
1220 C      NNN=KS(KI)
1230 C      EARG=- (NNN**2) *AOR0 * (R-RGOOD(KI))
1240 C      EXA=EXP(EARG)
1250 C      IF (LPS .EQ. 0) DENR=1.0
1260 C      IF (LPS .EQ. 1) DENR=R/RGOOD(KI)
1270 C      Y(2*KI)=YGOOD(KI)*EXA/DENR
1280      Y(2*KI)=0.00
1290      15 CONTINUE
1300      14 CONTINUE
1310      INF2=2*N+1
1320      SMT=TDPD*DDT*1500.0*AK0A**2 /2.0
1330      APF2=AOR0*(FRAT)**2
1340      F32=0.25*FRAT*SIG0
1350      R21=R**2+1.000
1360      RR21=R/R21
1370      R211=1.000/R21
1380      IF (LPS .EQ. 1) RI=1.000/R
1390      DO 100 LI=1,N
1400      SUM1=0.
1410      SUM2=0.
1420      KI=KS(LI)
1430      ATTF=1.0+TDPD/(1.0+KI**2*SWK**2)
1440      DO 101 MM=1,INF2
1450      M=MM-N-1
1460      IF (M .EQ. 0) GO TO 101
1470      IMA=IABS(M)
1480      MA=KS(IMA)
1490      IF (M .LT. 0) MA=-MA
1500      VM=Y(2+IMA-1)
1510      WM=Y(2+IMA)
1520      IF (M .LT. 0) WM=-WM
1530      NM=KI-MA
1540      IF (NM .EQ. 0) GO TO 101
1550      LNMA=IABS(NM)

```

COPY AVAILABLE TO DDC DOES NOT
 PERMIT FULLY LEGIBLE PRODUCTION

```

1560 IF (LNMA .GT. KS(N)) GO TO 101
1570 IF (NFR .LE. NHL2) GO TO 11
1580 MFNM=MOD(LNMA,NFR)
1590 IF (MFNM .EQ. 0) MFNM=NFR
1600 MH=IABS(NFR-MFNM)
1610 IF (MFNM .LE. NHL .OR. MH .LE. NHL) GO TO 10
1620 GO TO 101
1630 10 QNMN=(LNMA-0.5)/NFR
1640 NMDN=INT(QNMN)
1650 MD=MFNM
1660 IF (MH .LE. NHL) MD=NHL21-MH
1670 INMA=NMDN*NHL21+MD
1680 GO TO 12
1690 11 INMA=LNMA
1700 12 CONTINUE
1710 VNM=Y(2*INMA-1)
1720 WNM=Y(2*INMA)
1730 IF (NM .LT. 0) WNM=-WNM
1740 SUM1=SUM1+ (QNMN*VM+VNM*WM)
1750 SUM2=SUM2+ (QNMN*WM-VNM*VM)
1760 101 CONTINUE
1770 IF (LUP .NE. 3 .OR. KI .GT. NHL) GO TO 13
1780 IF (LPS .GE. 2) GO TO 13
1790 FRN=FRAT*KI
1800 FRFN=FRN*R
1810 DFR=(FRN+FRFN**2)/(1.0+FRFN**2)
1820 DFB=FRFN*(1.0-FRN)/(1.0+FRFN**2)
1830 TNLN1=KI*FS2*(DFR*SUM1+DFB*SUM2)
1840 TNLN2=KI*FS2*(DFR*SUM2-DFB*SUM1)
1850 GO TO 18
1860 13 TNLN1=KI*FS2*SUM1
1870 TNLN2=KI*FS2*SUM2
1880 18 CONTINUE
1890 TLN1=-Y(2*LI-1)*ARF2*KI**2 *ATTF
1900 TLN1=TLN1-Y(2*LI) *KI*SWT/(1.0+KI**2 *SMK**2)
1910 TLN2=-Y(2*LI) *ARF2*KI**2 *ATTF
1920 TLN2=TLN2+Y(2*LI-1)*KI*SWT/(1.0+KI**2 *SMK**2)
1930 IF (LPS .EQ. 0) GO TO 104
1940 IF (LPS .EQ. 1) GO TO 105
1950 IF (LPS .EQ. 2) GO TO 106
1960 TLN1=TLN1-PR21*Y(2*LI-1)
1970 TLN2=TLN2-PR21*Y(2*LI)
1980 GO TO 104
1990 106 TLN1=TLN1-PR21*Y(2*LI-1)+R21I*Y(2*LI)
2000 TLN2=TLN2-PR21*Y(2*LI) -R21I*Y(2*LI-1)
2010 GO TO 104
2020 105 TLN1=TLN1-RI*Y(2*LI-1)
2030 TLN2=TLN2-RI*Y(2*LI)
2040 104 CONTINUE
2050 F(2*LI-1)=TLN1+TNLN1
2060 F(2*LI) =TLN2+TNLN2
2070 100 CONTINUE
2080 PLAST=R
2090 RETURN
2100 END
99END

```

**COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION**


```

100      SUBROUTINE OUTPUT(R,Y)
110      DOUBLE PRECISION R,Y(120),RLN,YMAG,XMSG
120      COMMON N,FRAT,AOR0,SIG0,F,LPS,K3,NHL,NHL21,NFR,NHL2,LUP,
130      1 XMSG,IAP,KSN,KSN1,KSN2,AOR0,IBM,RBM
140      DIMENSION F(120),KS(60)
150      COMMON /BLK1/XP,XD,PS1,PS2,XDB1,IX,IFLAG,M,PSN1,PSN2,
160      1 RLD,XMAX,NSC,XT,PAT,TMAX
170      DIMENSION XP(22,5),XD(22,5),PS1(22,5),PS2(22,5),XDB1(22,5),
180      1 PSN1(22,5),PSN2(22,5),XT(41,5),PAT(41,5)
190      2030 FORMAT (7H IFLAG=,I3)
200      3010 FORMAT (3H R=,E10.4,3X,8H SIG0+R=,E10.4,3X,7H LN(R)=,E10.4,3X,
210      1 /12H SIG0+LN(R)=,E10.4,3X,17H SIG0+ARCSINH(R)=,E10.4)
220      3020 FORMAT (3X,2H N,5X,2H V,11X,2H W,11X,4H MAG,9X,9H EXTRA DB,
230      1 5H LOSS)
240      3030 FORMAT (2X,I3,4(3X,E10.4))
250      3040 FORMAT (/6H SIG0=,E10.4)
260      IF (IX.EQ. 0) PRINT 3040,SIG0
270      PRINT 2030,IFLAG
280      SR=R
290      IF (NSC.EQ. 1) SR=SIG0*SR
300 C    LPS=0 FOR PLANE WAVE, =1 FOR SPHERICAL, =2 FOR COMBINED PHASE,
310 C    =3 FOR COMBINED NO PHASE
320      IF (R.LT. 1.0D-20) GO TO 10
330      RLN=DLOG(R)
340      IF (LPS.EQ. 0) DBL=9.686+AOR0*P
350      IF (LPS.EQ. 1) DBL=8.686*(RLN+AOR0*(R-1.0D0))
360      IF (LPS.GE. 2) DBL=8.686*(AOR0*P+DLOG(DSORT(1.0D0+P**2)))
370      SOLP=RLN
380      IF (NSC.EQ. 1) SOLP=SIG0*SOLP
390      GO TO 11
400      10 RLN=-9.99D9
410      SOLP=-9.99E9
420      11 CONTINUE
430      SOGR=DLOG(R+DSORT(R**2+1.0D0))
440      IF (NSC.EQ. 1) SOGR=SIG0*SOGR
450      IF (IFLAG.EQ. 5) GO TO 16
460      IX=IX+1
470      IF (LPS.EQ. 0) XP(IX,MY)=SR
480      IF (LPS.EQ. 1) XP(IX,M)=SOLP
490      IF (LPS.GE. 2) XP(IX,M)=SOGR
500      XD(IX,M)=R
510      16 CONTINUE

```

```

520 C      IF (N3C .EQ. 1) PRINT 3010,R,SR,RLN,SOLR,SUGR
530 C      PRINT 3020
540      DO 301 KI=1,N
550      K=KS(KI)
560      K21=2*KI-1
570      K2=2*KI
580      YMAG=DSORT(Y(K21)**2+Y(K2)**2)
590      IF (YMAG .LT. 1.0D-20 .OR. R .LT. 1.0D-20) GO TO 12
600      EDBL=-20.0*DLOG10(YMAG)-DBL
610      GO TO 13
620 12 EDBL= 9.99E9
630 13 CONTINUE
640      IF (IFLAG .EQ. 5) GO TO 14
650      IF (KI .NE. 1) GO TO 15
660      PS1(IX,M)=YMAG
670      XDB1(IX,M)=EDBL
680      GO TO 14
690 15 IF (KI .NE. 2) GO TO 17
700      PS2(IX,M)=YMAG
710      GO TO 14
720 17 IF (KI .NE. KSN1) GO TO 18
730      PSN1(IX,M)=YMAG
740      GO TO 14
750 18 IF (KI .NE. KSN2) GO TO 14
760      PSN2(IX,M)=YMAG
770      GO TO 14
780 14 CONTINUE
790      IF (KI .GT. 2) GO TO 301
800 C      PRINT 3030,K,Y(K21),Y(K2),YMAG,EDBL
810 301 CONTINUE
820      RETURN
830      END
999END

```



```

100 FUNCTION FPSI(R,A0R0,SIG0,LPS,IAP)
110 DOUBLE PRECISION R,ARG1,ARG2,ARG,ARGN,DEI,EIT,SQR,DEN,DNR
120 IF (IAP.EQ. 2) GO TO 30
130 ARGN=-A0R0/R
140 ARG1=A0R0+2.000
150 ARG=ARG1+2.000
160 IF (LPS.GE. 2) GO TO 47
170 IF (LPS.EQ. 1) GO TO 46
180 PSI1=DEXP(ARGN)
190 GO TO 50
200 46 ARG2=ARG1/R
210 DEN=R
220 GO TO 45
230 47 ARG2=ARG1+(1.000+R)
240 DEN=1.000+R
250 45 EIT=(-DEI(-ARG1))-(-DEI(-ARG2))
260 SQR=1.000+(SIG0/2.000)**2+DEXP(ARG)*EIT**2
270 PSI1=DEXP(ARGN)/(DEN*DSORT(SQR))
280 GO TO 50
290 30 IF (R.LE. 0.00) GO TO 11
300 A=0.0
310 IF (LPS.EQ. 1) A=1.0
320 B=R
330 HSTAR=(B-A)*1.0E-2
340 HMIN=(B-A)*1.0E-4
350 HMAX=(B-A)/2.0
360 ERMAX=1.0E-3
370 KEY=0
380 CALL ROMBS(A,B,X,FOFX,HSTAR,HMIN,HMAX,ERMAX,ANS,KSTOP,KEY)
390 10 ARG=-2.0+A0R0/X
400 IF (LPS.EQ. 0) DNI=1.0
410 IF (LPS.EQ. 1) DNI=X
420 IF (LPS.GE. 2) DNI=DSORT(1.0+X**2)
430 FOFX=EXP(ARG)/DNI
440 CALL ROM2
450 IF (KSTOP.EQ. 1) GO TO 10
460 GO TO 20
470 11 ANS=0.0
480 20 SQR2=1.0+(SIG0/2.000)**2+ANS**2
490 ARGN=-A0R0/R
500 IF (LPS.EQ. 0) DNR=1.000
510 IF (LPS.EQ. 1) DNR=R
520 IF (LPS.GE. 2) DNR=DSORT(1.000+R**2)
530 PSI1=DEXP(ARGN)/(DNR*DSORT(SQR2))
540 50 CONTINUE
550 FPSI=PSI1
560 RETURN
570 END
580 END

```