Three-Dimensional Parabolic-Equation-Based Estimation of the Ocean Acoustic Field

JOHN S. PERKINS AND RALPH N. BAER

Large Aperture Acoustics Branch
Acoustics Division

and

LAURA F. ROCHE AND L. BRUCE PALMER

Planning Systems Incorporated
McLean, Virginia 22102

March 31, 1983

NAVAL RESEARCH LABORATORY
Washington, D.C.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Report 8685	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

THREE-DIMENSIONAL PARABOLIC-EQUATION-BASED ESTIMATION OF THE OCEAN ACOUSTIC FIELD

5. TYPE OF REPORT & PERIOD COVERED

Interim report on a continuing NRL problem.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

John S. Perkins, Ralph N. Baer, Laura F. Roche*, and L. Bruce Palmer*

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Research Laboratory
Washington, DC 20375

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

Prog. Elem. 61153N
Proj. RR0110843
Problem 51-1327-0-3

11. CONTROLLING OFFICE NAME AND ADDRESS

Office of Naval Research
Arlington, VA 22217

12. REPORT DATE

March 31, 1983

13. NUMBER OF PAGES

48

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

*Planning Systems, Inc., McLean, VA 22102

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Acoustic propagation	Three-dimensional propagation models
Transmission loss	Propagation models
Array performance	Parabolic equation
Signal gain	Split-step algorithm

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a three-dimensional parabolic-equation-based model for deterministic acoustic propagation in a deep-ocean environment. It constitutes one of the outputs of a research program in spatial properties of the low-frequency acoustic field initiated in 1974 to describe the effects of internal ocean inhomogeneities on the performance of large undersea acoustic antennas. In that program these inhomogeneities were divided into large-scale phenomena such as ocean eddies or major ocean fronts and smaller scale internal-wave-type features thought to be more evenly distributed throughout the ocean. This model deals with the former case and enables the

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued)

complete calculation of the three-dimensional acoustic field, that is, the azimuthal as well as the vertical distribution of energy. It can handle both a three-dimensionally varying sound-speed field and a triangularly faceted bathymetry. To facilitate calculations with moderate horizontal gradients and little or no crossangle scattering generated by the bottom, a modified version that in effect enables the propagation of a coherent phase-tracked fan is also described and developed. The programs of the model are operational on the Texas Instruments Advanced Scientific Computer (ASC) system at NRL and are available to the Navy scientific community through the Navy Laboratory Computer Network (NALCON). The programs may be operated remotely, or the entrire code may be transferred to a receiving site over this network. At present the complete three-dimensional calculation requires the capabilities of a large machine comparable to the ASC, such as the Cray-1, or conceivably a super-minicomputer augmented with a number of array processors. The complete model is best restricted to frequencies less than 50 Hz, and the modified version, which is faster and requires less central memory, can affordably deal with frequencies an order of magnitude greater. The report contains analytic developments, operating instructions, and test-case calculations.

CONTENTS

PREFACE

This report is a major output of a program in spatial properties of low-frequency acoustic fields in the deep ocean. This project was instituted at the Naval Research Laboratory in 1974 to provide a priori estimates of the capabilities or limits to large-array construction and performance due to coherence degradation from environmental causes. Although the project work has emphasized stochastic propagation measures of irregularities in the ocean, a significant effort was allotted to developing formulations applicable in three-dimensional deterministic ocean environments. The methods described in this report have been used in theoretical studies of propagation as well as for pre-experimental predictions and verifications.

The objective of the present report is to present the techniques, as well as the resultant computer implementation, for the first numerical algorithm for acoustic propagation able to incorporate arbitrary environmental information in three dimensions. Since in many instances there is little, if any, fine-scale structure that needs to be included in a statistical fashion, it is appropriate in such scenarios to treat the propagation as a deterministic process. Especially it is in regions with large (order of 100 km) three-dimensional features that the model described herein has its greatest utility.

THREE-DIMENSIONAL PARABOLIC-EQUATION-BASED ESTIMATION OF THE OCEAN ACOUSTIC FIELD

INTRODUCTION

The propagation of sound in the ocean is influenced by many factors. These include both processes that are deterministic and processes that can be best described statistically. Deterministic processes include both refraction and diffraction; processes best described statistically include scattering by both volume inhomogeneities and bathymetric irregularities. There have been many theoretical formulations and computer-based implementations of deterministic solutions of the governing equations of sound propagation. Reference 1 is an excellent survey of these models. Most of these models are two dimensional. Interest in three-dimensional acoustic propagation models has increased due to the need to understand and predict the performance of acoustic antennas in ocean environments which vary in all spatial dimensions.

We have by use of the parabolic-equation (PE) technique developed numerical algorithms and a sequence of FORTRAN programs to model underwater acoustic propagation in three dimensions through a deterministic ocean region. These programs reside on the Texas Instruments Advanced Scientific Computer (ASC) system at NRL and are available to the Navy scientific community through the Navy Laboratory Computer Network (NALCON). They represent the first implementation of a model predicting the received field in an ocean region capable of handling both an arbitrary three-dimensionally varying sound-speed field and arbitrary bathymetry. Thus, it is possible to investigate propagation in an environment where the presence of a mesoscale feature, such as a front or an eddy, introduces an azimuthal inhomogeneity. How such a feature affects the apparent source bearing, the array signal gain, and the spreading of the incoming energy can be studied numerically.

We have developed a technique which uses the PE approximation to the Helmholtz equation for acoustic propagation and solves the equation using a three-dimensional (3D) extension of the split-step algorithm. There are two versions of the 3D PE model: a complete version best restricted to low-frequency propagation and an approximate version that can accommodate higher frequency propagation. As currently implemented on the ASC, the complete (low-frequency) version requires the maximum water depth to be less than approximately 100 acoustic wavelengths, due to central memory limitations. This version allows for either a full 3D model, requiring the use of direct and inverse Fourier transforms in two dimensions (depth and azimuthal angle),[2] or a modified approach which treats each specified azimuthal angle as a separate (two-dimensional) case.[3] We will refer to the latter approach as the $N \times 2D$ method. This method, which does not perform transforms in angle, is typically faster by 40% to 50%. Although this technique does not allow energy to transfer between different vertical planes, the phase may vary with azimuthal angle, allowing wavefronts to bend. This has been shown to be an excellent approximation for waterborne propagation.[3]

The high-frequency version is based on the ability with the $N \times 2D$ method to efficiently maintain certain large matrices on disk (instead of in central memory). This allows finer grids and therefore higher frequencies. If the frequency is sufficiently high that the maximum water depth is greater than 100 wavelengths, then the $N \times 2D$ approach, as implemented in the high-frequency version, *must* be used. On a computer with greater central memory than NRL's ASC (which has roughly 172 user-available memory pages, where 1 page is 4096 32-bit words) the 100-wavelength limit of the complete (low-frequency) version could be raised. The high-frequency version can accommodate approximately 1500 wavelengths in the water column.

Manuscript approved January 4, 1983.

All of the programs in this package have been written to make full use of the vectorizing and pipelining capabilities of the ASC, which is a two-pipe machine with an optimizing compiler and is capable of computation speeds of 50 million floating-point operations per second for certain vector calculations. This enables rapid calculation of the fast Fourier transforms (FFTs) on which the model is heavily dependent. The FORTRAN codes are easily transportable to another computer, except for possible modifications required by disk input/output or graphic results.

In the next section we outline the theoretical formulations of the PE approximation and the split-step algorithm. A more detailed derivation of the 3D extension of the split-step algorithm is contained in Appendix A. The remaining sections are devoted to the programs comprising the computer model. We explain the purpose of each program, its implementation, and its operation on the ASC. Special attention is placed on the differences between the two versions discussed previously. Appendix B contains a sample run of the sequence of programs, including inputs and illustrative graphic outputs.

THREE-DIMENSIONAL PARABOLIC-EQUATION THEORY

In cylindrical coordinates the Helmholtz equation for 3D acoustic wave propagation in the ocean is

$$\frac{1}{r}\frac{\partial}{\partial r}\left[r\frac{\partial p}{\partial r}\right] + \frac{\partial^2 p}{\partial z^2} + \frac{1}{r^2}\frac{\partial^2 p}{\partial \theta^2} + k_0^2 n^2(r,z,\theta)p = 0, \tag{1}$$

where

$k_0 =$ wavenumber $= 2\pi f/c_0$,
$n(r,z,\theta) =$ refraction index $= c_0/c(r,z,\theta)$,
$p =$ acoustic pressure,
$f =$ source frequency,
$c_0 =$ reference sound speed,
$c(r,z,\theta) =$ sound speed,
$r =$ range,
$z =$ depth,
$\theta =$ azimuthal angle.

The solution of Eq. (1) can be written as

$$p(r,z,\theta) = \left\{\frac{e^{ik_0 r}}{\sqrt{r}}\right\}\Phi(r,z,\theta), \tag{2}$$

where Φ satisfies

$$\Phi_{rr} + \Phi_{zz} + 2ik_0\Phi_r + \frac{\Phi_{\theta\theta}}{r^2} + k_0^2\left\{n^2 - 1 + \frac{1}{4k_0^2 r^2}\right\}\Phi = 0. \tag{2a}$$

We may assume that Φ/r^2 is negligible in the far field and that[4]

$$|\Phi_{rr}| << |2k_0\Phi_r|. \tag{2b}$$

The physical assumptions have been well considered in the literature.[5–7] We now may say that Φ approximately satisfies

$$\Phi_{zz} + 2ik_0\Phi_r + \frac{\Phi_{\theta\theta}}{r^2} + k_0^2\ (n^2 - 1)\Phi = 0. \tag{3}$$

An approximate solution of Eq. (3) can be found using the split-step algorithm, which marches the solution Φ in range according to[2]

$$\Phi(r + \Delta r, z, \theta) = \exp\left\{\left(\frac{ik_0}{2}\right)(n^2 - 1)\Delta r\right\} \times F_\theta^{-1}\left\{\exp\left[\frac{-i\Delta r m^2}{2k_0 r(r + \Delta r)}\right]\left[F_\theta F_z^{-1}\left\{\exp\left\{\frac{-il^2\Delta r}{2k_0}\right\} F_z(\Phi(r,z,\theta))\right\}\right]\right\}, \tag{4}$$

where F_z and F_z^{-1} are the Fourier transform and inverse transform in depth and F_θ and F_θ^{-1} are the transform and inverse transform in angle. A derivation of this formula is outlined in Appendix A.

The corresponding formula for the $N \times 2D$ approach is[3]

$$\Phi(r + \Delta r, z; \theta) = \exp\left\{\left(\frac{ik_0}{2}\right)(n^2 - 1)\Delta r\right\} F_z^{-1}\left\{\exp\left[\frac{-il^2\Delta r}{2k_0}\right] F_z(\Phi(r,z;\theta))\right\}, \tag{5}$$

where θ is simply a parameter. The dependence of n on θ is retained. Thus, the $N \times 2D$ method does not allow energy to transfer between different vertical planes, but phase may vary with θ, allowing wavefronts to bend. This has been shown to be an excellent approximation for waterborne propagation.[3] The $N \times 2D$ method cannot be used when there is significant horizontal redirection of energy due to bottom reflection.

In practice, the angle transforms of Eq. (4) are accomplished with the fast Fourier transform (FFT), and the depth transforms of Eqs. (4) and (5) are accomplished with a fast sine transform.

GENERAL DESCRIPTION OF THE COMPUTER IMPLEMENTATION

We have written a package of FORTRAN programs to numerically solve Eqs. (4) and (5). These programs reside on the Texas Instruments Advanced Scientific Computer (ASC) at NRL. The structure of this set of programs, their relationships to one another, their titles, and their place in the whole calculation are schematically shown in Figs. 1 through 4.

The heart of the 3D PE calculation is the program SPLCYL, which is an implementation of the split-step algorithm. Before the algorithm can be executed, the bottom topography, the sound-speed field (and hence $n(r,z,\theta)$), and an initial pressure field must be specified. These tasks are separated from the main program SPLCYL and are performed in the preliminary programs BOT3D, PRO3D, and START3.

Program BOT3D reads in bottom-depth data and generates an output file containing the ocean bathymetry. This file (FT03F001) serves as input to programs PRO3D, START3, and SPLCYL. Program BOT3D can also produce a plot of the ocean bottom.

Similarly program PRO3D reads in sound-speed-versus-depth profiles and generates an output file of sound-speed profiles. This file (FT02F001) serves as input to programs START3, SPLCYL, and TRNISM. Program PRO3D also produces plots of the sound-speed field in the form of sound-speed contours for depth versus range (at a fixed azimuthal angle) and angle versus range (at a fixed depth).

Programs BOT3D and PRO3D both edit and interpolate the input data to reduce the computing and storage loads of succeeding programs. Also, both are common to the low- and high-frequency versions of the 3D PE model. However, the choice of some of their input parameters (such as interpolation tolerances) may depend on whether the full 3D or $N \times 2D$ approach is to be employed by SPLCYL.

Program START3 generates the initial (startup) field that will be subsequently propagated, or stepped out in range, by the split-step algorithm in program SPLCYL. This field can be achieved in

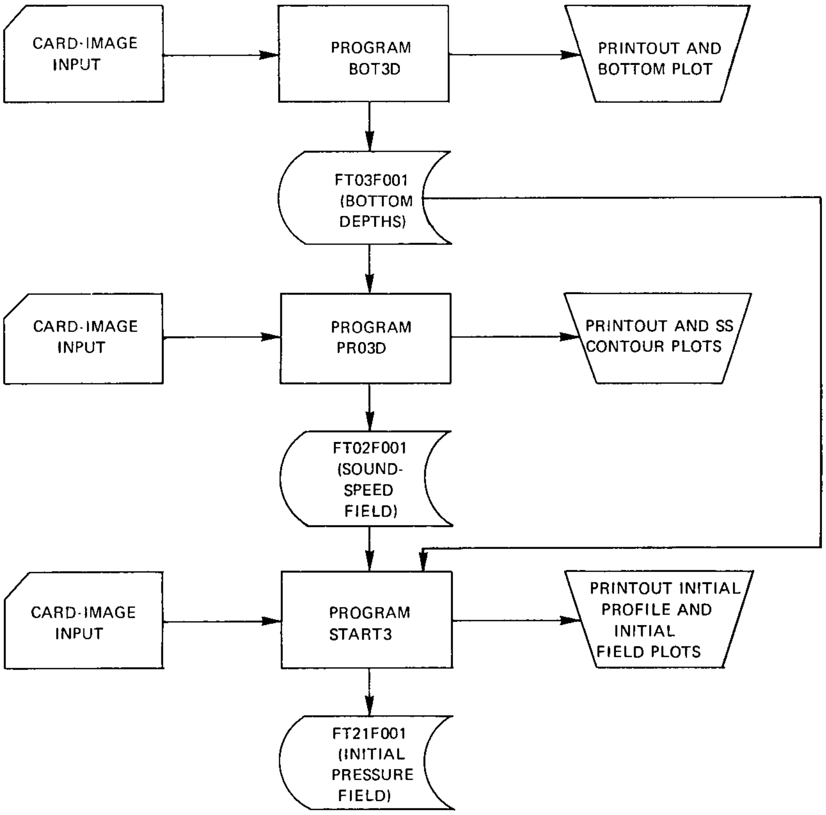

Fig. 1 — Input/output flowchart of programs BOT3D, PRO3D, and START3

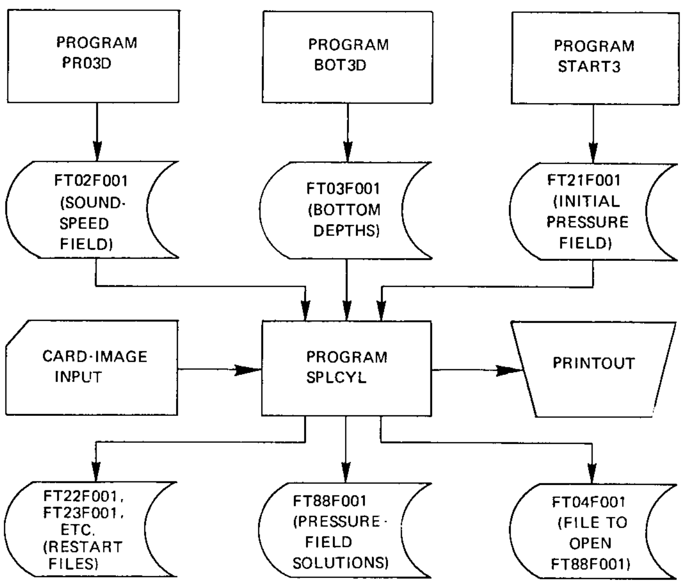

Fig. 2 — Input/output flowchart of program SPLCYL

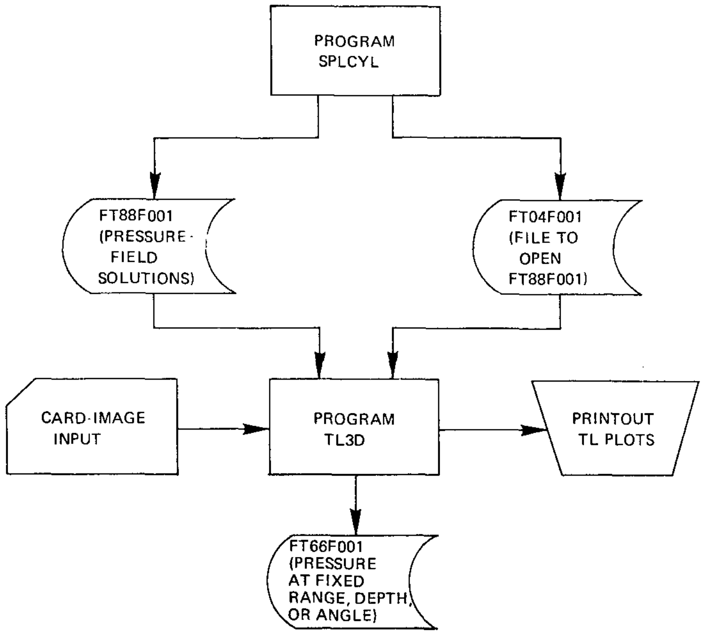

Fig. 3 — Input/output flowchart of program TL3D

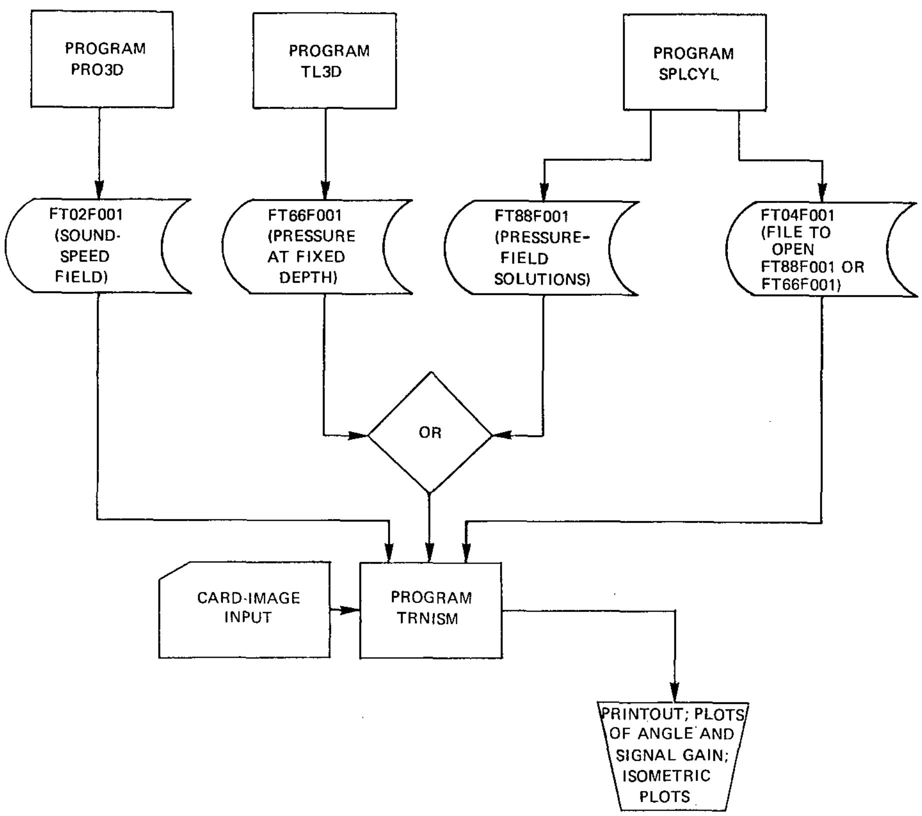

Fig. 4 — Input/output flowchart of program TRNISM

one of three ways: by a normal-mode calculation,[8,9] by a functional form which is Gaussian in depth, or by a user-supplied complex-FORTRAN-function subroutine to override the Gaussian. The normal-mode method is more accurate but considerably more expensive, especially at higher frequencies. In any case the initial field is written onto file FT21F001 for input to program SPLCYL. Also, program START3 can produce a plot of the initial pressure field. Program START3 differs somewhat between the low- and high-frequency versions. The high-frequency version allows some matrices (most notably those which hold the eigenvectors) to be dimensioned much larger than the corresponding matrices in the low-frequency version. The high- and low-frequency versions of program START3 have the same input parameters.

Program SPLCYL is the primary program of the 3D PE model. It performs the actual split-step algorithm to produce the PE solution at each specified depth, range, and azimuthal angle. The low-frequency version allows the user to choose whether or not to use a full 3D model, which requires the use of direct and inverse Fourier transforms in two dimensions (angle and depth), or a modified approach which treats each angle as a separate two-dimensional case, the $N \times 2D$ method. The high-frequency version of program SPLCYL uses the $N \times 2D$ method.

Input to program SPLCYL includes card-image data and the three previously generated files of bottom topography (FT03F001), sound-speed field (FT02F001), and initial startup field (FT21F001). Output includes printout and at least two output files. One file (FT88F001) contains the predicted pressure field as a function of depth, angle, and range. The other file (FT04F001) serves as a management file for reading the first file (FT88F001). Both files serve as input to the two succeeding postprediction processing programs TL3D and TRNISM. In addition to these two files, program SPLCYL optionally generates one or more restart files (FT22F001, FT23F001, ...). Each restart file contains the pressure field at a given range, which can be used to restart the split-step algorithm (replace the initial field on FT21F001).

There are two followup programs to program SPLCYL: programs TL3D and TRNISM. Program TL3D uses the pressure-field solution from program SPLCYL (FT88F001) to optionally generate the following plots: transmission loss versus range (at fixed angle and depth), transmission loss versus angle (at fixed range and depth), and transmission loss versus depth (at fixed range and angle). TL3D also optionally creates output files (FT66F001, FT67F001, ...) which are files of the pressure field at an input fixed range, angle, or depth. These files can later be used to generate gray-scale plots of intensity on the VAX 11/780 computer system of the Large Aperture Acoustics Branch at NRL.

Program TRNISM draws isometric plots of acoustic intensity-versus-arrival angle as a function of range and also predicts the performance of a (perhaps tilted) horizontal line array when the array is within the region of the calculated pressure field. The array performance statistics include the arrival angle of maximum intensity, the 3-dB beamwidth, and the array signal gain.

Finally, there are ASC library files associated with the low- and high-frequency versions. LIB3D is a library file containing the subprograms for all of the low-frequency PE programs. When running any of the low-frequency programs, the user must allow the ASC access to LIB3D. Then, whenever a subprogram is called, the ASC searches through LIB3D until it finds the subprogram in question. Similarly, HILIB contains the subprograms for the high-frequency version. Although many of the subprograms in LIB3D are identical to their HILIB counterparts, because of the differences between the low- and high-frequency versions several LIB3D subprograms differ significantly from the corresponding HILIB subprograms. It is, therefore, *essential* to use the correct library file.

PROGRAM BOT3D: GENERATION OF THE BATHYMETRY

Program BOT3D generates a file (FT03F001) describing the bottom topography, which is then used in subsequent programs. Input to BOT3D is in the form of bathymetric data on card-image files.

The output bathymetry is the result of editing and interpolating input information. This processing is done to minimize the amount of information (and subsequent processing) needed to adequately describe the ocean bottom.

Program BOT3D uses the range parameters (in kilometers) RANMIN, RANINC, and RANMAX and the angle parameters (in degrees) ANGMIN, ANGINC, and ANGMAX to define the initial range-azimuthal-angle grid shown in Fig. 5. This grid determines candidate locations in range-angle space where bottom depths may be calculated and recorded. Program BOT3D also reads in a sequence of bottom depths and their range-angle locations:

(RANGE(I), ANGLE(I), DEP(I), I=1, NPTS) (km, deg, m).

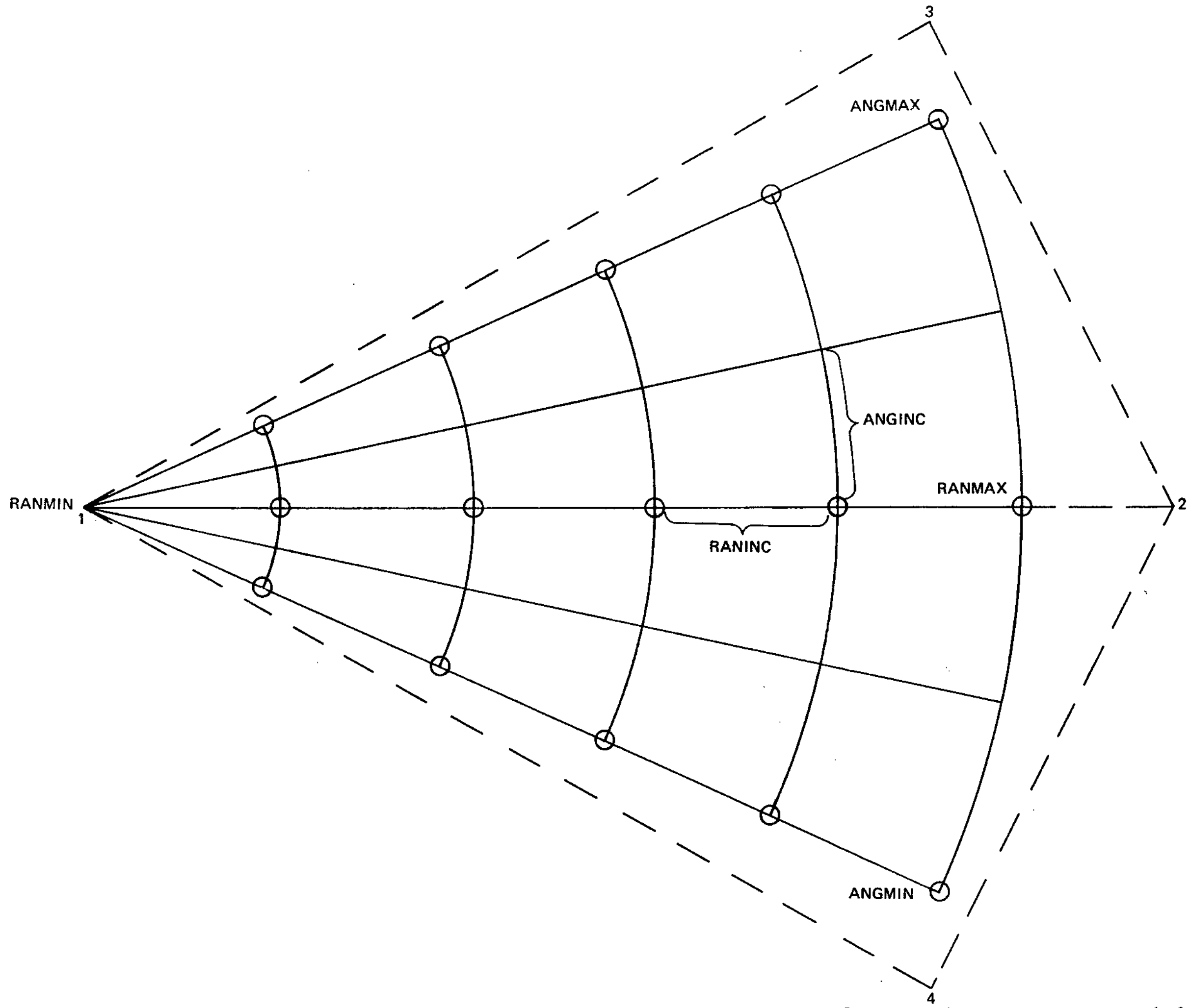

Fig. 5 — Sample range-angle grid for program BOT3D. Depth values have been specified at the four locations labeled 1, 2, 3, and 4. Two triangles have been specified, as indicated by the dashed lines. The circles indicate the three test points (M1 = 3) for each range.

Thus there are NPTS data points. These data points, which may be read without regard to order, are then indexed in the order in which they are read. From these points the user constructs triangles (Fig. 5) which in the aggregate must overlay the initial range-angle grid. The triangles are used in determining the bottom depth at any grid point. For a given grid point, program BOT3D finds the triangle which encompasses that grid point and then determines the bottom depth at the grid point by fitting a plane through the depths at the triangle vertices (the bottom is triangularly faceted).

The triangles are specified by the user through the input array LTRI. Array LTRI is two-dimensional with three columns and NTRI rows, where NTRI is the number of triangles to be constructed. For each row of LTRI (triangle) the column values are the indices of the input depth locations defining the vertices of the triangle. That is, ((LTRI(I,J), J=1, 3), I=1, NTRI) specifies, by vertex triples, a total of NTRI sets of three depth-location numbers defining NTRI triangles. For the case depicted in Fig. 5, NTRI=2 and two triples of numbers would be read in, such as 1 2 3 1 2 4.

Once the range-angle grid and the triangles are constructed, the parameters M1, ANGTOL (m), and RANTOL (m) are used to decide at which grid points the depth will be recorded. The program uses a two-level procedure to determine at which grid points bottom depths are to be recorded. First, it steps in range and performs a test to decide if depth values are needed at the current range. If no values are needed, the range is incremented and the test is repeated at the new range. When this test shows that depth values are needed at a particular range, the second level of the procedure is employed. The program now steps in angle from ANGMIN to ANGMAX (holding the range constant), performing tests to determine which angles require depth values. RANTOL and ANGTOL are depth tolerances used in the range and angle testing procedures respectively. When it is decided that the bottom depth should be recorded at a given range-angle combination, the depth is determined, as previously stated, by finding which input triangle the point is in and then fitting a plane through the depths at the triangle vertices.

The range incrementing and testing procedure is as follows. Depth values are always recorded at range RANMIN. Once the depths at a particular range have been recorded, the program decides if any depths at all are to be recorded at the next incremented range on the grid. This is accomplished by test points determined by ANGMIN, ANGMAX, and M1, with M1 test points being set up at equally spaced angles between ANGMIN and ANGMAX inclusive (Fig. 5). At each test angle the depths at the last recorded range and the current range are compared. If the depth differences at all test points are less than RANTOL, no depths are recorded for the current range, and program BOT3D goes on to consider succeeding ranges. The test at succeeding ranges is more complicated. At each test angle, it must be possible to regenerate (with error less than RANTOL) the depth values for all the intervening ranges between the last recorded range and the current range using linear interpolation. Otherwise, the test is failed, and depths must be recorded at the range step immediately preceding the current range. If the test is passed, the range is incremented, and the test is repeated. Regardless of the value of RANTOL, depths are always recorded at RANMAX.

The angle incrementing and testing procedure is as follows. If it has been decided to record bottom depths at a given range, the program finds the depth at angle ANGMIN and records it. Also, that depth is initially denoted the test depth. Then, stepping in the direction toward ANGMAX to the next angle on the grid at that range, the program determines the depth at the new angle. If the difference between this depth and the test depth is less than ANGTOL, the new depth is not recorded, and the program moves on to the succeeding angles. It must be possible to regenerate (within an error of ANGTOL) the depth values of all intervening angles between the last recorded angle and the current angle by linear interpolation or the test is failed. If the test is failed, a depth value must be recorded at the angle immediately preceding the current one. If the test is passed, the angle is incremented, and the test is repeated. A depth is always recorded at ANGMAX.

Program SPLCYL, in marching the parabolic field in range, may need depth values outside the region defined by RANMIN, RANMAX, ANGMIN, and ANGMAX. At ranges smaller than RANMIN the depth values for RANMIN will be in effect, and at ranges greater than RANMAX the values for RANMAX will be in effect. For any range, the depth at angle ANGMIN will be used for angles smaller than ANGMIN, and the depth at angle ANGMAX will be used for angles greater than ANGMAX.

Program BOT3D uses the data stored on FT03F001 to create a plot of the ocean bottom.

Inputs Required by Program BOT3D

The following is a list and brief descriptions of the card-image inputs of BOT3D:

M1	Number of test points in angle. This is used while deciding whether or not to record the depths at a certain range. M1=0 implies a flat bottom.
DMAX	Bottom depth (m). This is input only if M1=0.
IPLOT	Plotting flag. IPLOT=1 produces a bottom plot; IPLOT=0 produces no plot.
RANMIN	Minimum range (km).
RANMAX	Maximum range (km).
RANINC	Range increment, or step size (km).
ANGMIN	Minimum angle (deg).
ANGMIN	Maximum angle (deg).
ANGINC	Angle increment, or step size (deg).
	The area of the ocean under consideration is described by ANGMIN, ANGMAX, RANMIN, and RANMAX. Specifically, concern is with ranges from RANMIN to RANMAX km from the source and encompassed by ANGMIN to ANGMAX degrees. Further, a grid is set up between ANGMIN and ANGMAX, and RANMIN and RANMAX, where points on the grid are determined by ANGINC and RANINC.
RANTOL	Depth error tolerance (m), with respect to range.
ANGTOL	Depth error tolerance (m), with respect to angle.
	RANTOL is used in the range incrementing and testing procedure, and ANGTOL is used in the angle incrementing and testing procedure.
NPTS	Number of bottom points to be read in.
NTRI	Number of triangles to form from the input points.
RANGE(I)	Range (km) of the Ith input point; I=1, NPTS.
ANGLE(I)	Angle (deg) of the Ith input point; I=1, NPTS.
DEP(I)	Depth (m) of the Ith input point; I=1, NPTS.
LTRI(I,J)	Index (as read in) of the range-angle-depth point defining the Jth vertex (of three vertexes) of triangle I. For example, if LTRI(3,1)=4, LTRI(3,2)=6, and LTRI(3,3)=7, then the third triangle has vertices of the fourth, sixth, and seventh input points. The values LTRI can take on are from 1 to NPTS.

Formats for the Inputs to Program BOT3D

Input formats are as follows:

Card	Inputs	Format
1A	M1	(I5)
2A	DMAX	(F10.3)

Only cards 1A and 2A are required for a flat bottom. In that case, M1=0 and DMAX is the constant depth.

1	M1, IPLOT	(2I5)
2	RANMIN, RANMAX, ANGMIN, ANGMAX	(4F10.3)
3	RANINC, ANGINC	(2F10.3)
4	RANTOL, ANGTOL	(2F10.3)
5	NPTS, NTRI	(2I5)
6	RANGE(I), ANGLE(I), DEP(I), RANGE(I+1), ANGLE(I+1), DEPTH(I+1) Two range-angle-depth triples are entered per card. Card 6 is repeated until all NPTS points are entered.	(6F10.3)
7	((LTRI(I,J), J=1, 3), I=1, NTRI) There are vertices for five triangles per card. Card 7 is repeated until vertices for all NTRI triangles are read in.	(15I5)

Outputs From Program BOT3D

The outputs are as follows:

File FT03F001, the bathymetric data. This file serves as input to programs PRO3D, START3, and SPLCYL.

File FT06F001, the printout from BOT3D. This prints many of the input parameters, information about the triangles, and the points generated and kept on file FT03F001. On the printout the triangles may not be ordered or labeled as they were input.

File FT59F001, the bathymetric plot.

Use of Program BOT3D

The following is a sample deck for running BOT3D:

```
Card
 1    / JOB jobname,account,usercode,CAT=5,OPT=(T,R)
 2    / LIMIT BAND=50,SEC=500
 3    / LIMIT BAND=50,SEC=100
 4    / KEYBOARD
 5    / JSLOPTS OPT=(F,L,M)
 6    / PD PE,USERCAT/D81/L60/PE3D
 7    / PD YOU,USERCAT/D--/B--/usercode
 8    / DISSPLA VERS=8.2
 9    / ASG LIB3D,PE/LIB3D/OLIB,USE=SHR
10    / ASG BOTOBJ,PE/BOT3D/OBJ,USE=SHR
11    / LNK LSPACE=15000,LOAD=BOTGO
12    INCLUDE BOTOBJ/
13    LIBRARY LIB3D
14    LIBRARY DISSPOBJ
15    / REL SYS.PRT
16    / FXQT OPT=(I),GO=BOTGO,CPTIME=6000
        ···DATA···
17    / FOSYS FT06F001
18    / FOSYS FT59F001,TYPE=PLOT
19    / CATN YOU/nodename
20    / CAT YOU/nodename/FT03F001,ACNM=FT03F001
21    / EOJ
```

To run the high-frequency version, change line 9 to

```
9        / ASG LIB3D,PE/HILIB/OLIB,USE=SHR
```

Cards 1 through 5 and 21 need be used only if program BOT3D is not run directly from a terminal (if BOT3D is run from batch, or as an EXJOB). If BOT3D is run directly from a terminal, the user must sign on as a CATEGORY 5 job (or larger). Card 6 defines the path at which BOT3D and its subprograms are residing, and card 7 should be a path to the user's own files to catalog file FT03F001 for later use. Card 8 assigns library file DISSPLA, which is a package of plotting subroutines, and card 9 assigns library file LIB3D (or HILIB, if the high-frequency version is being run), which contains BOT3D's subprograms. Card 10 assigns the BOT3D main program. Card 11 is the ASC link command, and card 16 tells the ASC to execute BOT3D. Cards 17 and 18 have hard copies made of BOT3D's output, and cards 19 and 20 catalog FT03F001 in the user's own files. Many of these cards are used repeatedly in subsequent programs and will not be explained again. Instead the reader will be referred to this subsection.

PROGRAM PRO3D: GENERATION OF THE SOUND-SPEED FIELD

Program PRO3D operates much the way program BOT3D operates except that sound-speed profiles are involved rather than bottom depth points. That is, PRO3D reads card-image sound-speed profile data, interpolates and edits these data using triangles again, and generates an output file (FT02F001) of the resulting profiles which describes the sound-speed field.

There are two distinct versions of PRO3D, which we will refer to as the low- and high-frequency versions because, even though the choice of which version to use is independent of frequency, the subroutines for one version are in the low-frequency library and the subroutines for the other version are in the high-frequency library. The high-frequency version is recommended for use with typical ocean profiles. It is faster (and hence cheaper) and is sufficiently accurate for typical ocean profiles. The low-frequency version is normally used for analytic formulations of sound-speed profiles, where the predetermined or standard depths internal to the high-frequency version may be inadequate.

Program PRO3D uses a range-angle grid and triangles, whose vertices are the locations of input sound-speed profiles, similar to the grid and triangles used in BOT3D. The user should be aware, however, that the grid and triangles used by PRO3D are independent of those used by BOT3D.

Program PRO3D uses the range parameters (in kilometers) RANMIN, RANINC, and RANMAX and the angle parameters (in degrees) ANGMIN, ANGINC, and ANGMAX to define a two-dimensional range-angle grid. This grid is analogous to the grid in program BOT3D; the range-angle grid represents candidate locations for the recording of sound-speed profiles. Program PRO3D reads a sequence of NPRO data sound-speed profiles, along with their range and angle location, and the number of entries in the profile:

NPTS, RANGE, ANGLE	(., km, deg)
(D(I), C(I), I=1, NPTS)	(m, m/s).

These profiles, which may be read without regard to order, are consecutively indexed in the order in which they are read. From the locations of these input profiles, range-angle triangles are constructed, which in total must overlay the initial range-angle grid. These triangles are specified in the identical manner as in program BOT3D. That is, the indices of the profile locations defining the vertices of NTRI triangles are read in by vertex triples: ((LTRI(I,J), J=1, 3), I=1, NTRI).

The tolerance tests performed by program PRO3D are more complicated, due to the additional dimension of depth. Here, the parameters M1, N1, ANGTOL (m/s), and RANTOL (m/s) are used to decide which of the initial range-angle grid points are to have profiles recorded.

The profiles which are input to PRO3D are all extrapolated linearly to a depth of DEPMAX m (DEPMAX is the maximum water depth which was input to BOT3D and is on file FT03F001), and all profiles calculated by PRO3D extend to this depth. That is, PRO3D carries out the range and angle incrementing and testing procedures (described below) as if the bottom were flat and the depth was DEPMAX m.

It should be noted that PRO3D calculates the sound speed at a given range-angle-depth position by the following procedure:

1. The input triangle in which the range-angle point lies is determined.
2. Each of the three profiles at the triangle vertices are linearly interpolated in depth to get a sound speed at the given depth.
3. The sound speed at the given location is then found by planar interpolation using the three sound speeds.

We will refer to this procedure as triangular interpolation.

If the sound speeds at a particular range have been recorded (profiles are always recorded for ranges RANMIN and RANMAX), program PRO3D then decides whether any profiles will be recorded at the next range of the range-angle grid. It decides by comparing point by point the sound speeds on a two-dimensional depth-angle test grid at the new range with sound speeds on the same test grid at the last range. If at least one difference in sound speeds is greater than RANTOL, profiles are recorded at the new range, which also becomes the new test range. (This is a difference and not an interpolation. It is done this way since SPLCYL will use sound-speed data valid at one range until it encounters new sound-speed information.) The depth-angle test grid is determined by parameters M1 and N1. M1 test angles are equally spaced between ANGMIN and ANGMAX degrees inclusive (in the same way as shown in Fig. 5 for program BOT3D), and N1 depths are equally spaced between 0.0 and DEPMAX m inclusive. Otherwise, no profiles are recorded for that range, and succeeding ranges are considered until one is found where at least one difference is greater than RANTOL.

If it has been decided to record sound-speed profiles at a given range, the program puts a profile at ANGMIN. It then steps through the angles of the initial range-angle grid toward ANGMAX, while testing each new angle in a manner analogous to that of program BOT3D. However, here the test between sound speeds at the last recorded sound-speed profile and at the new angle is performed at each of Nl equally spaced depths between 0.0 and DEPMAX. If any one of the sound-speed differences at these depths is greater than ANGTOL, a profile is generated at the new angle which becomes the new test profile. Otherwise, no profile is generated, and the next angle is considered. This process is repeated until ANGMAX is reached and a profile is recorded there.

Once it has been decided to record a sound-speed profile at a particular range and angle, an additional process is performed. This is the process which differs between the low- and high-frequency versions of PRO3D. In the high-frequency version, sound speeds are recorded at a predetermined set of depths which are internal to the program. These depths are adequate to describe most ocean profiles. In the low-frequency version, an editing and interpolating procedure is carried out: The first point on the profile is always the surface point (the sound speed is determined by triangular interpolation), and this point becomes the first test point in the procedure. The program then steps down the depths of the initial range-angle-depth grid, testing to see if linear interpolation in depth between the test depth and the current depth is sufficiently accurate (with an error less than DEPTOL m/s) to regenerate the sound speeds (as determined by triangular interpolation) at all intervening points on the grid. If the test is passed, the current depth is incremented one grid point, and the test is repeated. If the test is failed, the current depth is *decremented* by one grid point, and this depth and its sound-speed value are added to the profile. This new point on the profile now becomes the test point, and the process is repeated until DEPMAX is reached, where the last point is added to the profile.

Program SPLCYL, in marching the parabolic field in range, may need profiles outside the region defined by RANMIN, RANMAX, ANGMIN, and ANGMAX. At ranges smaller than RANMIN the profiles for RANMIN will be in effect, and at ranges greater than RANMAX the profiles for RANMAX will be in effect. For any range, the profile at angle ANGMIN will be used for angles smaller than ANGMIN, and the profile at angle ANGMAX will be used for angles greater than ANGMAX.

Both versions of program PRO3D can override the tests for determining which of the points on the range-angle grid sound-speed profiles will be recorded. If the input parameter ITYPE is set to 1, the program will record profiles at all locations on the range-angle grid. In this case the parameters M1, N1, ANGTOL, and RANTOL are ignored.

Finally program PRO3D optionally generates sound-speed contour plots on two-dimensional axes. Contours may be plotted for either or both range versus angle (at a fixed depth) and range versus depth (at a fixed angle). Several of each type of plot may be generated.

Inputs Required by Program PRO3D

Program PRO3D reads the bathymetric data file (FT03F001) generated by program BOT3D. The following is a list and brief descriptions of the card-image inputs to PRO3D. Unless otherwise indicated, these inputs are read in the main program PRO3D.

Inputs Affecting Sound-Speed Field Generation

ITYPE	Testing flag. If ITYPE=1, testing is not done, and subroutine QKSTEP is used instead. If ITYPE=0, testing is done when generating profiles.
RANMIN	Minimum range (km).
RANMAX	Maximum range (km).
RANINC	Range increment (km).
ANGMIN	Minimum angle (deg).
ANGMAX	Maximum angle (deg).
ANGINC	Angle step size (deg).
	As in program BOT3D, the ocean region being considered in program PRO3D is marked out by RANMIN, RANMAX, ANGMIN, and ANGMAX. We are considering the region between RANMIN and RANMAX km from the source and bounded in azimuth by ANGMIN and ANGMAX degrees. Also, as in BOT3D, a grid is set up over the region in question, with grid points being determined by RANINC and ANGINC.
DEPINC	Depth increment (m). DEPINC determines the number of depths per profile and the distance between depths. DEPINC is ignored in the high-frequency version.
RANTOL	Sound-speed error tolerance (m/s) with respect to range.

ANGTOL	Sound-speed error tolerance (m/s) with respect to angle.
DEPTOL	Sound-speed error tolerance (m/s) with respect to depth. DEPTOL is ignored in the high-frequency version.
N1	Number of equally spaced test depths.
M1	Number of equally spaced test angles.
NPRO	Number of input profiles. NPRO is read in subroutine SPEED3. $1 < \mathrm{NPRO} < 150$.
NTRI	Number of triangles to be made with input profiles as vertices. NTRI is read in SPEED3. $1 < \mathrm{NTRI} < 200$.
IPRT	Print flag. IPRT$\neq$0 means input sound-speed profiles, along with relevant information about the triangles, will be printed. IPRT is read in SPEED3.
NPTS(J)	Number of data points in the Jth input profile; J=1, NPRO. NPTS(J) is read in SPEED3. $1 < \mathrm{NPTS(J)} < 100$.
RANGE(J)	Range (km) of the Jth input profile; J=1, NPRO. RANGE(J) is read in SPEED3.
ANGLE(J)	Angle (deg) of the Jth input profile; J=1, NPRO. ANGLE(J) is read in SPEED3.
D(I),C(I)	Depth and sound. Each input profile is specified as a sequence of pairs of depth and sound speed. C(I) is the sound speed (m/s) at depth D(I)(m), I=1, NPTS(J). Depth D(1) must be zero. D(I) and C(I) are read in SPEED3.
LTRI(I,J)	Triangle vertices; I=1, NTRI; J=1, 3. As in BOT3D, LTRI(I,J) tells which input profile is at the Jth vertex of the Ith triangle. LTRI(I,J) is read in SPEED3.

Generation of Range-Versus-Depth Contour Plots

NRD	Number of sound-speed contour plots to be drawn of range versus depth at a fixed angle. NRD is read in PLTSS3.
	For each range-versus-depth sound-speed contour plot the following nine inputs are read in PLTSS3:
ANGLE	Fixed angle (deg) at which the plot is being made.
DPLTMN	Minimum depth (m) over which the contour will be plotted.
DPLTMZ	Maximum depth (m) over which the contour will be plotted.
RPLTMN	Minimum range (km) over which the contour will be plotted.
RPLTMX	Maximum range (km) over which the contour will be plotted.
SSMIN	Minimum sound speed (m/s) considered for contours.
SSMAX	Maximum sound speed (m/s) considered for contours.
NCL	Number of contour levels to be plotted. (SSMAX-SSMIN)/(NCL-1) is the step size between contour levels.
KMPI	Plot scale factor (km/in.). (RPLTMX−RPLTMN)/KMPI is the length of the plot in inches.

Generation of Range-Versus-Angle Contour Plots

NRA	Number of sound-speed contour plots to be drawn of range versus angle at a fixed depth. NRA is read in PLTSS3.
	For each range-versus-angle sound-speed contour plot the following nine inputs are read in subroutine PLTSS3. For ease of display, a rectangular range-angle coordinate system is used.
DEPTH	Fixed depth (m) at which the plot is being made.
APLTMN	Minimum angle (deg) over which the contour will be plotted.
APLTMX	Maximum angle (deg) over which the contour will be plotted.
RPLTMN	Minimum range (km) over which the contour will be plotted.
RPLTMX	Maximum range (km) over which the contour will be plotted.
SSMIN	Minimum sound speed (m/s) considered for contours.
SSMAX	Maximum sound speed (m/s) considered for contours.
NCL	Number of contour levels to be plotted. (SSMAX-SSMIN)/(NCL-1) is the step size between contour levels.
KMPI	Plot scale factor (km/in.). (RPLTMX−RPLTMN)/KMPI is the length of the plot in inches.

Formats for the Inputs to Program PRO3D

Input formats are as follows:

Card	Inputs	Format
1	ITYPE	(I5)
2	RANMIN, RANMAX, ANGMIN, ANGMAX	(4F10.3)
3	RANINC, DEPINC, ANGINC	(3F10.3)
	DEPINC is ignored in the high-frequency version.	
4	RANTOL, DEPTOL, ANGTOL	(3F10.3)
	DEPTOL is ignored in the high-frequency version.	
5	NI, MI	(2I5)
6	NPRO, NTRI, IPRT	(3I5)
7	NPTS, RANGE, ANGLE	(I5,2F10.3)
8	(D(I), C(I), I=1, NPTS)	(10F8.2)
	There are five pairs of depth and sound speed per card. Card 8 is repeated until all NPTS pairs are read in. Card 7 and card(s) 8 are repeated NPRO times	
9	((LTRI(I,J), J=1, 3), I=1, NTRI)	(15I5)
	There are vertices for five triangles per card. Card 9 is repeated until all NTRI triangles are accounted for.	
10	NRD, NRA	(2I5)
11	ANGLE, DEPMIN, DEPMAX, RANMIN, RANMAX, SSMIN, SSMAX, NCL, KMPI	(7F10.3,2I5)

Card 11 is repeated once for each of the NRD contour plots. Card 11 is omitted if NRD=0.

12 DEPTH, ANGMIN, ANGMAX, RANMIN, RANMAX, SSMIN, SSMAX, NCL, KMPI (7F10.3,2I5)

Card 12 is repeated once for each of the NRA contour plots. Card 12 is omitted if NRA=0.

Outputs from Program PRO3D

The outputs are as follows:

File FT02F001, the sound-speed profiles. This file serves as input to START3, SPLCYL, and TRNISM.

File FT06F001, the printout from PRO3D. This prints many of the input parameters, the input profiles, information about the triangles, and the sound-speed profiles generated and recorded on FT02F001. On the printout the triangles may not be ordered or labeled as they were input.

File FT59F001, the contour plots.

Use of Program PRO3D

The following is a sample deck for running PRO3D:

```
Card
  1   / JOB jobname,account,usercode,OPT=(T,R),CAT=9
  2   / LIMIT BAND=50,SEC=500
  3   / LIMIT BAND=50,SEC=100
  4   / KEYBOARD
  5   / JSLOPTS OPT=(F,L,M)
  6   / PD PE,USERCAT/D81/L60/PE3D
  7   / PD YOU,USERCAT/D--/B--/usercode
  8   / ASGP FT03F001,YOU/nodename/FT03F001,USE=SHR
  9   / ASG LIB3D,PE/LIB3D/OLIB,USE=SHR
 10   / ASG CONLIB,USERCAT/D81/L60/CONLIB/OLIB,USE=SHR
 11   / ASG PROOBJ,PE/PRO3D/OBJ,USE=SHR
 12   / DISSPLA VERS=8.2
 13   / LNK LSPACE=15000,LOAD=PROGO
 14   INCLUDE PROOBJ/
 15   LIBRARY LIB3D
 16   LIBRARY DISSPOBJ
 17   / FD FT06F001,BAND=1/20/1
 18   / WAIT
 19   / FXQT OPT=(I),GO=PROGO,CPTIME=90000
        ···DATA···
 20   / CAT YOU/nodename/FT02F001,ACNM=FT02F001
 21   / FOSYS FT06F001
 22   / FOSYS FT59F001,TYPE=PLOT
 23   / EOJ
```

To run the high-frequency version, card 9 is changed to

```
9       / ASG LIB3D,PE/HILIB/OLIB,USE=SHR
```

Card 8 assigns file FT03F00l, the bottom profile created in BOT3D. Card 12 assigns the object code of PRO3D. Card 17 is a file definition which allows file FT06F001 more than the default amount of space. The remaining cards are comparable to cards in BOT3D. These cards were explained in the final subsection on BOT3D.

PROGRAM START3:GENERATION OF THE INITIAL PRESSURE FIELD

Program START3 creates an initial, or startup, pressure field that is subsequently propagated, or stepped out in range, by program SPLCYL. The initial field is assumed independent of azimuthal angle. This initial field, which is stored on file FT2lF001, may be generated in a number of ways. A real-valued Gaussian-shaped function of pressure versus depth may be selected, which is generated in a complex-valued FORTRAN function called STFTN. Alternatively, the user may supply his own startup field by writing his own function STFTN and substituting it for the currently existing one. However, the following FORTRAN statements should be used in the created function subprogram:

```
COMPLEX FUNCTION STFTN (Z)
DOUBLE PRECISION R0, C0, F, BSS, SOURD
COMMON /SGOUT/ R0, C0, F, BSS, SOURD
  ...
STFTN=?
RETURN
END
```

Another method of generating a startup field comes from normal-mode theory. The implemented normal-mode algorithm is due to Ingenito and coworkers, and the interested user should consult Refs. 8 and 9. The technique is an iterative one that requires the specification of a convergence parameter EPSILN. This parameter multiplies the maximum pressure, and the resulting value is the allowable accuracy in terms of how well the calculated initial pressure field converges to zero at the surface. In choosing a value for EPSILN, it should be chosen small enough so that the convergence is good but not so small that START3 has to repeat code an excessive number of times to achieve convergence. From experience, EPSILN = 0.001 appears to be a good choice.

Program START3 generates a pressure-versus-depth function which is duplicated at a specified number, (2**MPOW)+1, of equally spaced angles (so the initial pressure field does not vary in azimuth). The values of these angles are specified when running program SPLCYL. The depth function at a specified number, (2**NPOW)−1, of equally spaced depths (and the specified number of angles) is written onto an output file FT2lF00l for subsequent input to program SPLCYL. The associated depth values are obtained from the input bottom profile (FT03F001). The equally spaced depths are spaced over 4/3 of the maximum bottom depth. The extra points are required by SPLCYL for the split-step algorithm. NPOW should be large enough so that this spacing is less than 1/2 wavelength ($\lambda/2$). Also, at this point, the maximum bottom depth should be less than 100λ if the low-frequency version of the 3D PE model is being used.

A shooting algorithm which steps up from the ocean bottom to the surface is used in the normal-mode startup. This algorithm typically requires that the depth points be spaced much less than 1/2 wavelength apart. Thus, the depth function is calculated at a greater number of depths than specified by NPOW, although NPOW still determines the number of depth samples which are output. The greater number of depth points is the number of output points (2**NPOW)−1 multiplied by an input parameter NINT which must be a power of 2. NINT=4 is usually adequate.

Program START3 has the option of writing calculated eigenvalues and eigenfunctions onto an output file FTl7F001. However, unless the user has a particular need for this file, it should not be generated.

Although the normal-mode startup is in theory more accurate than the Gaussian, it is also much more expensive and time consuming to run, especially at high frequencies. Further, the accuracy of the Gaussian startup increases dramatically with increasing range. The user should take into account the range over which SPLCYL is to be run and the amount of money available for computer runs when deciding which startup form to use.

Finally program START3 optionally generates a plot of the initial sound-speed profile and the initial pressure field.

Inputs Required by Program START3

Program START3 reads the bathymetric data file (FT03F001) generated by program BOT3D and the sound-speed data file (FT02F001) generated by program PRO3D. The following is a list and brief descriptions of the card-image inputs to program START3. They are all read in the main program START3.

ISTART	Flag to indicate the method in which the initial startup field is to be generated. If ISTART = 0, the normal-mode method is used. If ISTART = 1, the Gaussian or a user-supplied method is used.
IPLOT	Plot flag. If IPLOT = 1, a pressure-versus-depth plot of the initial pressure field is created. If IPLOT = 0, no plot is created.
NPOW	Variable giving the number of output depth points which will be (2**NPOW)−1, with NPOW < 8 in the low-frequency case and NPOW < 12 in the high-frequency case. NPOW should be large enough so that (4./3.)*(maximum depth)/ (2**NPOW) < λ/2.0, where λ is the wavelength.
MPOW	Variable giving the number of output angle points, which will be (2**MPOW)+1, with MPOW≤7.
F	Source frequency (Hz).
R0	Range (km) of the initial field. If ISTART=0 (normal-mode method), R0 should be several wavelengths (because a far-field approximation is used). If ISTART=1 (Gaussian method), R0 should be 0.001 km.
BSS	Sound-speed (m/s) within the ocean bottom. This is needed because the initial field extends into the bottom in depth.
SOURD	Depth (m) of the source.
	The following inputs are not used if ISTART=1.
ITAPE	Flag for creating file FT17F001. If ITAPE = 0, file FT17F001 of eigenvalues and eigenfunctions is not created. This is the suggested value. If ITAPE = 1, file FT17F001 is created.
NINT	Multiplier of the number of output depth points to determine the depths at which modes are calculated. NINT must be a power of 2, with the suggested value being NINT=4. NINT*(2**NPOW) ≤ 16384.
MAXMOD	Maximum number of modes to be used to calculate initial field. If the actual number of modes is larger than MAXMOD, only the first MAXMOD will be found. If all modes are desired, MAXMOD should be made large. The number of modes will be approximately 2*F*H*SQRT(1./C0**2−1./BSS**2), where H is the water depth at the source, C0 is the average sound speed in the water column, and F and BSS are program inputs. If all modes are desired, MAXMOD should be made much larger than this estimate.

RHO1	Density (g/cm^3) of the water. Usually RHO1=1.0.
RHO2	Density (g/cm^3) of the material below the ocean floor.
	These density parameters are used only by the normal-mode routine. They are needed to properly incorporate the bottom boundary condition.
EPSILN	Convergence parameter. EPSILN=0.001 works well.

Formats for the Inputs to Program START3

Input formats are as follows:

Card	Inputs	Format
1	ISTART, IPLOT, NPOW, MPOW, F, R0, BSS, SOURD	(4I5, 4F10.3)
2	ITAPE, NINT, MAXMOD, RHOl, RHO2, EPSILN	(3I5, 3F10.3)
	Card 2 may be omitted if ISTART=l.	

Outputs from Program START3

The outputs are as follows:

File FT2lF001, the initial pressure field at range R0. This file is used in program SPLCYL as the field from which the solution is propagated.

File FT06F001, the printout from START3. This prints many of the input parameters, the sound-speed profile from file FT02F001 used in the calculations, the eigenvalues found and information about them, and the real and imaginary parts of the initial pressure field.

File FT59F001, the sound-speed profile and initial-pressure-field plot.

Use of Program START3

The following is a sample deck for running START3:

```
Card
 1    / JOB jobname,account,usercode,OPT=(T,R),CAT=5
 2    / LIMIT BAND=50,SEC=400
 3    / LIMIT BAND=50,SEC=200
 4    / KEYBOARD
 5    / JSLOPTS OPT=(F,L,M)
 6    / PD PE,USERCAT/D8l/L60/PE3D
 7    / PD YOU,USERCAT/D--/B--/usercode
 8    / ASGP FT02F00l,YOU/nodename/FT02F00l,USE=SHR
 9    / ASGP FT03F00l,YOU/nodename/FT03F00l,USE=SHR
10    / ASGP LIB3D,PE/LIB3D/OLIB,USE=SHR
11    / ASG STARTOBJ,PE/START3/OBJ,USE=SHR
12    / DISSPLA VERS=8.2
13    / LNK LSPACE=15000,LOAD=STARTGO
14    INCLUDE STARTOBJ/
15    LIBRARY LIB3D
16    LIBRARY DISSPOBJ
17    / REL SYS.PRT
```

```
18   / FD FT06F001,BAND=1/20/1
19   / WAIT
20   / FXQT OPT=(I),GO=STARTGO,CPTIME=6000
       ···DATA···
21   / CAT YOU/nodename/FT21F001,ACNM=FT21F001
22   / FOSYS FT06F001
23   / FOSYS FT59F001,TYPE=PLOT
24   / EOJ
```

To run the high-frequency version, cards 10 and 11 are changed to

```
10   / ASG LIB3D,PE/HILIB/OLIB,USE=SHR
11   / ASG STARTOBJ,PE/HISTART/OBJ,USE=SHR
```

Card 11 assigns the ASC object code of START3. The other cards are comparable to cards used in running BOT3D and PRO3D, as was explained in the last subsection of the section on BOT3D and of the section on PRO3D.

PROGRAM SPLCYL: PROPAGATION OF THE PARABOLIC-EQUATION SOLUTION OUT IN RANGE

Program SPLCYL is the primary program of the 3D PE model. Given descriptions of the ocean environment (on files FT02F001 and FT03F001) and an initial pressure field (on file FT21F001), the split-step algorithm is applied to propagate the acoustic field out in range. The calculated pressure fields at specified ranges, angles, and depths are written onto an output file (FT88F001) for postprocessing by programs TL3D and TRNISM. Attendant with this pressure-field file is another file (FT04F001) which is a management file for the subsequent reading of data from file FT88F001. The output file (FT88F001) can be very large and hence expensive to store on the ASC as a disk file. This file should be cataloged on magnetic tape.

Program SPLCYL has the option of writing the pressure field at one or more intermediate ranges onto output files (FT22F001, FT23F001, ...). These files can then be used to replace file FT21F001 as the initial startup field, and program SPLCYL can be used to propagate the new (restartup) field.

The number of angles to be propagated is determined when running program START3. As was described in the preceding section, an input to START3 is the variable MPOW, and the number of angles to be used is (2**MPOW)+1. The values of these (2**MPOW)+1 angles are determined in program SPLCYL by the parameter TOTANG, which specifies the total angular extent of the field. The angular grid consists of (2**MPOW)+1 equally spaced angles from −TOTANG/2.0 degrees to TOTANG/2.0 degrees inclusive. In START3 the input NPOW determines the number of depths to be used, namely, (2**NPOW)−1 and these depths are evenly spaced over 4/3 of the maximum depth of the water column (although there is no grid point at the surface). Even though NPOW and MPOW are input in START3, they must also be input in SPLCYL, and the values input in SPLCYL must be identical to the values input in START3.

The depth transforms are taken over a region extending down from the ocean surface to 4/3 of the maximum water depth. The region below the maximum water depth serves two purposes: the bottom sound speed can be incorporated into the problem, and near the bottom edge of the transform region an exponential taper is applied to the pressure field to prevent artificial reflections. The boundary condition at the surface ($\Phi(r,\theta,0) = 0$) is met through the use of a sine transform. In the full 3D version of SPLCYL the field is tapered in angle on each side of the transform region analogously to the method used in depth. One-eighth of the total angular region is given over to tapering on each side of the region. Tapering is not necessary when the $N \times 2D$ method is used, since no transforms in angle take place.

In conjunction with program START3 and an input range step size DELKM, a grid of range-angle-depth points is defined where the propagated field will be computed. This grid need not conform to those associated with programs BOT3D or PRO3D (from files FT03F001 and FT02F001). In particular the angular extent of this grid need not match the angular extent established in BOT3D and PRO3D. When TOTANG is such that it exceeds the regions described in BOT3D or PRO3D, the environment outside those regions is extended in angle by assuming it to be constant in angle and equal to the nearest known value in angle.

Since the propagation grid need not coincide with the environment grids, the latter are interpolated to get environment information at the propagation grid points. Bottom depths are linearly interpolated in range and angle. Sound speeds on the depth grid points are generated by linear interpolation in depth. The model also has the option of linearly interpolating sound speeds in angle. If it is decided not to interpolate linearly across angle, SPLCYL uses the value at the nearest environmental-grid angle, in a negative direction. In range, SPLCYL uses the most recent (previous) sound-speed information.

To cut down on the required storage space for the output file FT88F001 on the ASC, not all of the calculated pressure values need to be written onto file FT88F001. The input parameters ISKPD, ISKPR, and ISKPA specify that only the predicted pressures at each ISKPDth depth, each ISKPRth range, and each ISKPAth angle point of the propagation grid will be saved on file FT88F001. For example, if ISKPD=4, ISKPR=3, and ISKPA=2, then only points at every fourth depth, every third range, and every second angle will be recorded on file FT88F001. ISKPD and ISKPA must be powers of 2. If ISKPR=ISKPD= ISKPA=1, all calculated information will be saved.

Transmission loss is calculated and printed (file FT06F001) at each range step for the set of depth-angle points (RECD(I), RECA(I), I=1, NRD) (m,deg).

Finally, the major differences between the low- and high-frequency versions of the 3D PE model occur in program SPLCYL. The low-frequency version has the option (via the input IANG) of performing a full 3D propagation, which requires Fourier transforms and inverse transforms in both depth and angle, or it can do a modified propagation which requires a Fourier transform and inverse transform in depth only, treating each angle as a separate 2D case (the $N \times$ 2D method). Because of the time and central memory required when performing a full 3D propagation, the low-frequency version can be run only if the maximum bottom depth is less than approximately 100 acoustic wavelengths. The high-frequency version of SPLCYL (which uses only the $N \times$ 2D method) is able to propagate over regions with maximum bottom depth much greater than 100 acoustic wavelengths. The $N \times$ 2D method not only reduces the substantial computation time but also maintains certain large matrices on disk (instead of in central memory), which allows finer grids and therefore higher frequencies. This cannot be done efficiently with the full 3D algorithm.

Inputs Required by Program SPLCYL

Program SPLCYL reads the bathymetric data file (FT03F001) generated by program BOT3D, the sound-speed data file (FT02F001) generated by program PRO3D, and an initial-pressure-field file (FT21F001) generated by START3 or a previous run of SPLCYL. The following is a list and descriptions of the card-image inputs to program SPLCYL. All variables are read in the main program SPLCYL.

IBEGIN — Variable, read only in low-frequency version, which indicates on what file the initial pressure field is. IBEGIN=1 means the initial presure field is on file FT21F001. This is the usual case. IBEGIN≠1 means the initial pressure field is on file FT18F001. This implies the initial startup field was not generated using program START3 but was generated by the two-dimensional startup program.[10]

IANG	Variable, read only in the low-frequency version, to determine whether the 3D or $N \times$ 2D method is used. If IANG=0, the full 3D algorithm is used (Fourier and inverse transforms in both depth and angle). If IANG≠0, the $N \times$ 2D method is used.
NRD	Number of depth-angle positions where transmission loss is printed for each range step. NRD < 15.
NPOW	Variable giving the number of depth points to be used during the propagation of the pressure field, which is (2**NPOW)−1. This NPOW input must be the same as the NPOW input in START3.
MPOW	Variable giving the number of angles to be used, which is (2**MPOW)+1. This MPOW input must be the same as the MPOW input in START3.
ISKPD	Skipping parameter for depth values. Only pressure field points at every ISKPDth depth on the propagation grid are recorded on file FT88F001. ISKPD must be a power of 2.
ISKPR	Skipping parameter for range values. Only pressure field points at every ISKPRth range on the propagation grid are recorded on file FT88F001. ISKPR can be any positive integer.
ISKPA	Skipping parameter for angle values. Only pressure field points at every ISKPAth angle on the propagation grid are recorded on file FT88F001. ISKPA must be a power of 2.
INTERP	Variable to indicate whether sound speeds should be interpolated across angle to get sound-speed values at propagation grid points. INTERP=1 means that linear interpolatation is used. INTERP≠1 means that instead of linear interpolation the profile at the nearest angle, in the negative direction, is used.
RANMAX	Maximum range (km). The solution is propagated from R0 km to the range step nearest RANMAX km. (R0 is from START3.)
DELKM	Range step size (km). The solution is found at every DELKM km in range between R0 and RANMAX km.
TOTANG	Total angular extent (deg) of the region over which the solution is being found. The angles actually range from −TOTANG/2.0 to TOTANG/2.0.
RESDEL	Restart field indicator. If RESDEL is positive, then a restart pressure field is created every RESDEL km from the source to RANMAX (on files FT22F001, FT23F001, ...), and these fields can be used to replace FT21F001 and restart SPLCYL, propagating the solution from there. If RESDEL is negative, a single restart field is made at the last range. If RESDEL=0.0, no restart files are created. RESDEL cannot be so small that more than 66 restart files are generated.
DBPKM	Water-attenuation variable. If DBPKM is greater than or equal to zero, then an attenuation factor of DBPKM dB/km is used. Otherwise the attenuation is found from Thorpe's formula.[11]
RECD(I)	Depth (m) paired with angle RECA(I) to determine point for which transmission loss is printed out at every range step; I=1, NRD.
RECA(I)	Angle (deg) paired with RECD(I).

Formats for the Inputs to SPLCYL

Input formats are as follows:

Card	Inputs	Format
1	IBEGIN, IANG	(2I5)
	Card 1 is read in only in the low-frequency version.	
2	INTERP	(I5)
3	NRD, NPOW, MPOW, ISKPD, ISKPR, ISKPA	(6I5)
4	RANMAX, DELKM, TOTANG, RESDEL, DBPKM	(5F10.3)
5	RECD(I), RECA(I)	(8F10.3)
	Card 5 is repeated until all NRD depth-angle pairs are read in, with four pairs being read in per card.	

Outputs From Program SPLCYL

The outputs are as follows:

File FT88F001, the predicted pressure field as a function of range, angle, and depth.

File FT04F001, a management file for file FT88F001. It contains grid-size information.

File FT22F001 (or files FT22F001, FT23F001, ...), the restart file. This file contains a pressure field which can be used to replace FT21F001, and a solution can be propagated from the new file.

File FT06F001, the printout from SPLCYL. This prints out many of the input parameters and the transmission loss over range for each of the NRD depth-angle locations.

Use of Program SPLCYL

The following is a sample deck for running SPLCYL:

Card	
1	/ JOB jobname,account,usercode,OPT=(T,R),CAT=?
2	/ LIMIT BAND=?,SEC=?
3	/ LIMIT BAND=?,SEC=?
4	/ KEYBOARD
5	/ JSLOPTS OPT=(F,L,M)
6	/ PD PE,USERCAT/D81/L60/PE3D
7	/ PD YOU,USERCAT/D--/B--/usercode
8	/ ASGP FT02F001,YOU/nodename/FT02F001,USE=SHR
9	/ ASGP FT03F001,YOU/nodename/FT03F001,USE=SHR
10	/ ASGP FT21F001,YOU/nodename/FT21F001,USE=SHR
11	/ ASG SPLOBJ,PE/SPLCYL/OBJ,USE=SHR
12	/ ASG LIB3D,PE/LIB3D/OLIB,USE=SHR
13	/ FD FT08F001,BKSZ=3944,LREC=136,RCFM=FBA,BAND=1/10/1
14	/ FD FT06F001,BAND=1/20/1
15	/ LNK LOAD=SPLCYLGO
16	INCLUDE SPLOBJ/
17	LIBRARY LIB3D
18	/ REL SYS.PRT
19	/ WAIT

```
20    / FXQT OPT=(I),GO=SPLCYLGO,CPTIME=?,ADDMEM=?
      ···DATA···
21    / MFR name
22    / CAT YOU/nodename/FT88F001,ACNM=FT88F001,DTYP=TAPE
23    / CAT YOU/nodename/FT04F001,ACNM=FT04F001,DTYP=TAPE
24    / CAT YOU/nodename/RS22,ACNM=FT22F001,DTYP=TAPE
25    / MFRE
26    / FOSYS FT06F001
27    / FOSYS FT08F001
28    / EOJ
```

To run the high-frequency version, cards 11 and 12 are changed to

```
11    / ASG SPLOBJ,PE/HIFREQ/OBJ,USE=SHR
12    / ASG LIB3D,PE/HILIB/OLIB,USE=SHR
```

Card 11 assigns the object code of SPLCYL. Depending on whether the low- or high-frequency version is being run, it is necessary to assign the proper version of the SPLCYL object code, and the correct subprogram library. The other cards have been explained in previous last subsections.

When running program SPLCYL, the user must be careful to do the job in a large enough CATEGORY. The CATEGORY must be such that enough central memory pages are available, enough CP time and pseudo-time is available (and asked for in the LIMIT statement) for the program to run to completion, and enough BANDS of storage space are available (also through the LIMIT statement). The user is advised to estimate the amount of CP time and pseudo-time and the number of BANDS (and hence the CATEGORY) needed for the case in question. One way of estimating the CP time and pseudo-time needed is to run SPLCYL over a short-range distance (say 5 km). SPLCYL prints the total time it used at the end of the printout, and the user can judge from that the amount of time needed to run SPLCYL over the range. For example, if SPLCYL were run over 5 km and the run time were 10 s, the user could estimate that it would take 200 s to run SPLCYL over 100 km. (This run time of 10 s for a 5-km range is only an example and is not necessarily representative of an actual run time.) The user should probably allow extra CP time and pseudo-time beyond what was estimated, as a precaution. The approximate number of BANDS needed (for file FT88F001) is

(2**NPOW/ISKPD)*(2**MPOW/ISKPA)*((RANMAX/DELKM)/ISKPR)/16384.

As in the case of CP time, the user should allow for more bands than the preceding estimate indicates. Finally, in the FXQT statement, the user should increase the central memory size (using ADDMEM) to allow for buffers for the various input/output files. For the low-frequency version, ADDMEM should be at least

4*(2**NPOW−1) + 4*(2**NPOW/ISKPD−1)) + 8000,

and for the high-frequency version, ADDMEM should be at least

4*(2**NPOW−1) + ITAPR*ITAPA*(2*ITAPD) + 8000,

where ITAPR is the number of ranges saved on file FT88F001, ITAPA is the number of angles saved, and ITAPD is the number of depths saved.

To restart the program, the user simply assigns the restart file corresponding to the desired starting range (one of the files FT22F001, FT23F001, ...) to FT21F001. The program will propagate from the starting range (call it R1) to RANMAX (which now may be larger than on the previous run). The user replaces RANMAX in the BAND-estimating expression displayed in the preceding paragraph by RANMAX−R1.

PROGRAM TL3D: GENERATION OF TRANSMISSION-LOSS PLOTS

Program TL3D has two main purposes. One is to make plots of transmission loss versus range at fixed depth and angle, transmission loss versus depth at fixed range and angle, and transmission loss versus angle at fixed range and depth. The other main purpose is the creation of output files (FT66F001, FT67F001, ...) which are subsets of file FT88F001 and which may subseqently be used to draw gray-scale plots of pressure field intensity or as input to program TRNISM (next main section).

In producing transmission-loss-versus-range plots, TL3D reads in the parameter NTLR, which indicates the number of transmission-loss-versus-range plots to be made. TL3D then reads in the fixed depth (DEPTH) in meters and angle (ANGLE) in degrees and the beginning and ending ranges (X1, X2) in kilometers for each of the NTLR plots. Also input to TL3D (from file FT04F001) are the parameters relating to input file FT88F001. These include the minimum range, the range, depth, and angle step sizes (DELR, DELZ, and DELA, respectively) for the data on FT88F001, and the number of ranges, depths, and angles stored on FT88F001 (NRAN, NDEP, and NANG respectively). Using all of these inputs, TL3D determines exactly which points on FT88F001 are at the given fixed depth and angle and within the proper range interval. The values at these points are then converted to transmission loss (FT88F001 being a parabolic-pressure-field file) and stored in array TL. Array TL is then sent to subroutine PLOT3D which makes the actual transmission-loss-versus-range plot.

For transmission-loss-versus-depth plots, TL3D reads in the number of plots to be done (NTLZ) and performs a similar process. This time, however, TL3D determines a depth interval at a fixed range and angle in which pressure values are converted to transmission loss. Likewise, TL3D reads the number of transmission-loss-vs-angle plots (NTLA) and again goes through a similar procedure.

After program TL3D has handled all the transmission-loss plots, it then turns to making fixed-range, fixed-depth, and fixed-angle files. These files may then be written to magnetic tape, which can be picked up at NRL's ASC computer center. Programs on the VAX 11/780 computer system of the Large Aperture Acoustics Branch at NRL will draw gray-scale plots of intensity using the fixed-range, fixed-angle, or fixed-depth file as input. These programs are not documented here. The fixed-depth files made by TL3D can also be used as inputs to program TRNISM for finding the bearing angle, array signal gain, and beamwidth for a hypothetical horizontal array positioned at that depth.

When TL3D makes a fixed-range file, it stores the complex parabolic-pressure values for each depth-angle pair, omitting values at the first and last angle. That is, for each depth on file FT88F001 at the input range, TL3D writes the depth on the output file and then writes the pressure-field value for every angle, omitting the values at the first and last angle. Values are stored in order of decreasing angle rather than increasing angle (as might be expected). This is done so that a gray-scale plot made from this file is drawn with the correct orientation.

A similar process occurs when TL3D makes fixed-angle files. For each range on file FT88F001 at the input angle, TL3D writes the range on the output file, and then writes the pressure-field value for every depth. TL3D writes them on the output file in order of increasing depth (from minimum depth to maximum depth).

The last kind of file made by TL3D is a fixed-depth file. For each range on file FT88F001 at the input depth, TL3D writes the range on the output file and then writes the pressure-field value for every angle. As before, the values are stored in order of decreasing angle, and the first and last angles are omitted.

The output files that are made by TL3D are all named in the form FTIJF001, where IJ depends on how many output files have previously been created. For the first file IJ=66, and IJ is increased by one for each subsequent file. IJ takes on values from 66 to (66+NFIXR+NFIXZ+NFIXA−1). No more than 22 of these files can be generated in one run of TL3D.

Inputs Required by Program TL3D

Program TL3D reads the pressure-field file (FT88F001) and its companion file (FT04F001) generated by SPLCYL. The following is a list and brief descriptions of the card-image inputs to TL3D. These inputs are all read in the main program TL3D.

Transmission-Loss-Versus-Range Plots

NTLR	Number of transmission-loss-versus-range plots requested.
	For each transmission-loss-versus-range plot the following nine inputs are required:
NICE	Flag to determine the quality of the plot. NICE=0 yields a poorer quality, less expensive plot. NICE=1 gives a better quality plot.
DEPTH	Fixed depth (m) for the current transmission-loss-versus-range plot. This depth will be rounded to the nearest depth grid position on file FT88F001.
ANGLE	Fixed angle (deg) for the current transmission-loss-versus-range plot. This angle will be rounded to the nearest angle grid point on file FT88F001.
X1,X2	Minimum and maximum ranges (km) respectively for the current transmission-loss-versus-range plot.
XPINCH	Scaling factor (km/in.) to determine the length of the plot. The plot length will be (X2−X1)/XPINCH in.
SMOOTH	Smoothing window (km) for the transmission-loss plot. Gaussian weights are used with 6σ = SMOOTH km. SMOOTH = 0.0 causes no smoothing. The window will be reduced if necessary, so that it does not extend over more than 51 range steps.
DB1,DB2	Minimum and maximum decibel levels respectively for the current plot. These are the lower and upper bounds on the y axis of the plot, and transmission losses not between DB1 and DB2 will not be plotted. IF DB1=DB2 when read in, they are reset to DB1=40.0 dB and DB2=140.0 dB.

Transmission-Loss-Versus-Depth Plots

NTLZ	Number of transmission-loss-versus-depth plots requested.
	For each transmission-loss-versus-depth plot the following nine inputs are required. Because many of these are similar to the corresponding inputs for transmission-loss-versus-range plots, they will not be explained in as great detail.
NICE	Flag to determine the plot quality.
RANGE	Fixed range (km) for the current transmission-loss-versus-depth plot.
ANGLE	Fixed angle (deg) for the current transmission-loss-versus-depth plot.
X1,X2	Minimum and maximum depths (m) respectively for the current transmission-loss-versus-depth plot.
XPINCH	Plot-length scaling factor (m/in.)
SMOOTH	Smoothing window (m).
DB1,DB2	Minimum and maximum decibel levels respectively for the current plot.

3.5.1.3 Transmission-Loss-Versus-Angle Plots

NTLA	Number of transmission-loss-versus-angle plots requested.
	For each transmission-loss-versus-angle plot the following nine inputs are required. Because many of these are similar to the corresponding inputs for transmission-loss-versus-range plots, they will not be explained in as great detail.
NICE	Flag to determine the plot quality.
RANGE	Fixed range (km) for the current transmission-loss-versus-angle plot.
DEPTH	Fixed depth (m) for the current transmission-loss-versus-angle plot.
X1,X2	Minimum and maximum angles (deg) respectively for the current transmission-loss-versus-angle plot.
XPINCH	Plot-length scaling factor (deg/in.).
SMOOTH	Smoothing window (deg).
DB1,DB2	Minimum and maximum decibel levels respectively for the current plot.

Fixed-Range Files

NFIXR	Number of fixed-range files to create from FT88F001.
	For each fixed-range file, the following input is required:
RANGE	Range (km) of the current fixed-range file. This range will be rounded to the nearest range step.

Fixed-Depth Files

NFIXZ	Number of fixed-depth files to be created.
	For each fixed-depth file, the following input is required:
DEPTH	Depth (m) of the current fixed-depth file. This depth will be rounded to the nearest depth grid point.

Fixed-Angle Files

NFIXA	Number of fixed-angle files to be created.
	For each fixed-angle file, the following input is required:
ANGLE	Angle (deg) of the current fixed-angle file. This angle will be rounded to the nearest angle grid point.

Formats for the Inputs to Program TL3D

Input formats are as follows:

Card	Inputs	Format
1	NTLR, NTLZ, NTLA, NFIXR, NFIXZ, NFIXA	(6I5)
	NFIXR + NFIXZ + NFIXA cannot exceed 22.	
2	NICE, DEPTH, ANGLE, X1, X2, XPINCH, SMOOTH, DB1, DB2	(I1,4F10.3,4F7.2)
	Card 2 is repeated NTLR times.	

3	NICE, RANGE, ANGLE, X1, X2, XPINCH, SMOOTH, DB1, DB2	(I1,4F10.3,4F7.2)
	Card 3 is repeated NTLZ times.	
4	NICE, RANGE, DEPTH, X1, X2, XPINCH, SMOOTH, DB1, DB2	(I1,4F10.3,4F7.2)
	Card 4 is repeated NTLA times.	
5	RANGE	(F10.3)
	Card 5 is repeated NFIXR times.	
6	DEPTH	(F10.3)
	Card 6 is repeated NFIXZ times.	
7	ANGLE	(F10.3)
	Card 7 is repeated NFIXA times.	

Outputs from Program TL3D

The outputs are as follows:

File FT66F001 (or files FT67F001, FT68F001, ...), a fixed-range, fixed-depth, or fixed-angle file. The calculated pressures from file FT88F001 at the indicated range, depth, or angle are then written onto the output file.

File FT06F001, the printout from TL3D. This prints the number of plots created, with each plot's individual parameters and smoothing information. It also prints the number of fixed-range, fixed-depth, and fixed-angle files, the file number, and the associated file parameters.

File FT59F001, the transmission-loss-versus-range, -depth, and -angle plots.

Use of Program TL3D

The following is a sample deck for running TL3D:

```
Card
 1    / JOB jobname,account,usercode,OPT=(T,R),CAT=?
 2    / LIMIT BAND=50,SEC=500
 3    / LIMIT BAND=50,SEC=100
 4    / KEYBOARD
 5    / JSLOPTS OPT=(F,L,M)
 6    / PD PE,USERCAT/D81/L60/PE3D
 7    / PD YOU,USERCAT/D--/B--/usercode
 8    / ASGP FT88F001,YOU/nodename/FT88F001,USE=SHR
 9    / ASGP FT04F001,YOU/nodename/FT04F001,USE=SHR
10    / ASG TLOBJ,PE/TL3D/OBJ,USE=SHR
11    / ASG LIB3D,PE/LIB3D/OLIB,USE=SHR
12    / DISSPLA VERS=8.2
13    / LNK LSPACE=15000,LOAD=TLGO
14    INCLUDE TLOBJ/
15    LIBRARY LIB3D
16    LIBRARY DISSPOBJ
17    / REL SYS.PRT
18    / WAIT
19    / FXQT CPTIME=6000,OPT=(I),GO=TLGO,ADDMEM=?
       ···DATA···
```

```
20      / CAT YOU/nodename/FILE66,ACNM=FT66F001
21      / FOSYS FT06F001
22      / FOSYS FT59F001,TYPE=PLOT
23      / EOJ
```

To run the high-frequency version, card 11 is changed to

```
11      / ASG LIB3D,PE/HILIB/OLIB,USE=SHR
```

Card 10 assigns the object code of TL3D. Card 20 catalogs file FT66F001, which is a fixed range, depth, or angle file, on the user's own files for future use. If more than one fixed file is created, the user should remember to catalog all of them so they will be available for future use. In Card 19, the FXQT statement, the user must add enough central memory (ADDMEM) to allow file FT88F001 to be accessed.

PROGRAM TRNISM: CALCULATION OF ARRIVAL ANGLES, THE ARRAY SIGNAL GAIN, AND THE 3-dB BEAMWIDTH

Program TRNISM is the last of the six programs for the 3D PE model. This program finds, prints, and plots the primary arrival angle, array signal gain, and 3-dB beamwidth of a hypothetical array of hydrophones placed in the calculated acoustic pressure field. Program TRNISM also finds and prints any secondary arrival angles that are within a certain user-specified decibel level (DBN) of the primary arrival angle. TRNISM finds the arrival angle, the array signal gain, and the beamwidth over the range interval RANMIN to RANMAX input to the program. These ranges do not have to correspond to similar inputs in earlier programs. In addition TRNISM can produce an isometric plot of intensity versus angle as a function of range.

Program TRNISM can use either a fixed-depth file created and output by program TL3D, or it can use file FT88F001 from program SPLCYL to get pressure-field values used to determine the arrival angles, array signal gain, and 3-dB beamwidth. If one wants to predict the performance of a horizontal array, then a fixed-depth file (from TL3D) corresponding to the depth of the array should be input. To predict the performance of a nonhorizontal array, it is necessary to input the FT88F001 file produced by SPLCYL. The input flag ITLT specifies which case is being considered.

When TRNISM uses a fixed-depth file in the case of a horizontal array, TRNISM uses the inputs NUMHY, the number of hydrophones, and ALENG, the array length, to find the spacing between hydrophones. If ALENG is equal to zero, the spacing will default to half-wavelength spacing, and ALENG is reset so that the array length and number of hydrophones conforms to half-wavelength spacing. If ALENG is positive, then the spacing between hydrophones is ALENG/(NUMHY−1). The entry ANGOFF specifies the azimuthal angle at which the array is to be centered. Using ALENG and ANGOFF, subroutine DAGETR determines the angle of each hydrophone. DAGETR uses quadratic interpolation (in phase and amplitude) between the pressure-field values associated with nearby angles to get the value at each hydrophone's position. The phases are obtained by first removing variations due to differing sound-speed fields along each azimuthal direction and then interpolating and resorting the gross variations. Using those interpolated pressure-field values, the program calculates arrival angle, 3-dB beamwidth, array signal gain, and any secondary arrival angles and prints them. This process is repeated for each of the range steps from RANMIN to RANMAX. TRNISM may need to increase RANMIN so that the array will fit in the sector where the solution was calculated.

In the case of a tilted array, TRNISM must use file FT88F001. In this case TRNISM reads in CENTD and TILT, the depth of the array center in meters and the size of the array tilt in degrees measured positive clockwise as viewed from the source. TILT is the tilt of the array from horizontal, with a horizontal array having TILT=0.0. Again, if TILT=0.0, either a fixed-depth file or FT88F001 can be input; both give the same results. The fixed-depth file would be cheaper and faster. From NUMHY, CENTD, and TILT, TRNISM computes the depths over which the array extends and determines the depth of each hydrophone. As when a fixed-depth file is used, if ALENG equals zero, half wavelength spacing between hydrophones is used and ALENG is reset to conform. TRNISM then finds the angle of each hydrophone. Because the array is tilted, the angle associated with each hydrophone is a function of ANGOFF, ALENG, and TILT. For each hydrophone the subroutine DAGETR finds the two angles and two depths on file FT88F001 which encompass the hydrophone's position. Subroutine DAGETR reads from file FT88F001 the pressure field value at each of the four angle-depth points and interpolates linearly in phase and amplitude between those four points to get a pressure-field value at the hydrophone. Once the program has a pressure-field value for each hydrophone, it calculates arrival angle, 3-dB beamwidth, array signal gain, and any secondary arrival angles and prints them. Again the procedure is repeated for each range step from RANMIN to RANMAX.

TRNISM can plot arrival angle, 3-dB beamwidth, and array signal gain against range. TRNISM reads in the three input pairs BEAMIN, BEAMAX; DB3MIN, DB3MAX; and AMPMIN, AMPMAX. These represent the minimum and maximum values to be plotted of arrival angle, 3-dB beamwidth, and array signal gain respectively. If both entries in any of the three pairs are zero, then the plot associated with that pair is not produced. If both entries of a pair are equal but not zero, the program calculates a suitable vertical scale. TRNISM plots against range all values that lie between the input minimum and maximum. For example, if BEAMIN=BEAMAX=0.0, then no plot will be produced of arrival angle versus range, and if DB3MIN=0.0, DB3MAX=20.0, the vertical scale on the 3-dB width plot will extend from 0.0 to 20.0 degrees.

TRNISM also plots intensity-versus-arrival angle for each range step. TRNISM reads in the variables APLTMN and APLTMX, which are the minimum and maximum angles respectively for this isometric plot. For each range step TRNISM then finds intensity across angle, either at the depth of the fixed depth file or over the depths encompassed by the tilted array, and plots the horizontal distribution of intensity.

Inputs Required by Program TRNISM

Program TRNISM reads a pressure-field file (FT66F001 or FT88F001), its companion file (FT04F001), and the sound-speed data file (FT02F001). The following is a list and brief descriptions of the card-image inputs to TRNISM. These inputs are all read in the main program TRNISM.

F — Source frequency (Hz). This must be the same as in START3.

SOURD — Depth (m) of the source. This must be the same as in START3.

RANMIN — Minimum range (km) on the input pressure field file to consider.

RANMAX — Maximum range (km) to be used.

ANGOFF — Angle (deg) on the range-angle-depth propagation grid associated with file FT88F001 (from SPLCYL) where the array center is.

ALENG — Length (m) of the array. If ALENG=0.0, half-wavelength spacing between the hydrophones is used, and ALENG is reset to (NUMHY−1)*(half-wavelength).

TRNISM calculates arrival angles, 3-dB beamwidth, and array signal gain from RANMIN to RANMAX. It uses ANGOFF and ALENG to determine the total angular extent at each range step.

APLTMN — Minimum angle to be plotted for the intensity-versus-arrival-angle plots.

APLTMX — Maximum angle to be plotted for the intensity-versus-arrival-angle plots.

The intensity-versus-arrival-angle (angle extending from APLTMN deg to APLTMX deg) cuts are plotted (provided SCALE > 0.0) for each range step between RANMIN and RANMAX. If APLTMN > APLTMX, then APLTMN is reset to −90.0 degrees and APLTMX is reset to 90.0 degrees.

NUMHY — Number of hydrophones in the array. If NUMHY is not a power of 2, this will be reset to the nearest power of 2.

LIMMIN — Index for the minimum angle on the range-angle-depth propagation grid to be used in program TRNISM.

LIMMAX — Index for the maximum angle on the range-angle-depth propagation grid to be used in TRNISM.

If LIMMIN=LIMMAX=0, the program resets their values so that all the angles on the propagation grid are used except the first and last one-eighth of the angles. This is a useful option when running the full 3D algorithm, because the information associated with the first and last one-eighth of the angles are of no use, due to tapering. For example, if there are 129 angles stored on file FT88F001 and LIMMIN=LIMMAX=0, LIMMIN is reset to 16 and LIMMAX to 114. Then, for each range, if the array does not lie between the 16th and 114th angle, the arrival angles, 3-dB beamwidth, and array signal gain are not computed for that range. If the $N \times$ 2D method is used, then one can input LIMMIN=1 and LIMMAX=129.

MULT — "Zero-fill" multiplier. The actual size of the transform, and hence the actual number of beams calculated, is MULT*NUMHY. The default value is 1. Using MULT > 1 produces smoother plots but requires more computations. MULT must be a power of 2.

MAX100 — Variable which determines the spacing between data points for the plot of horizontal intensity cuts. It is also used to find the number of points to be linearly interpolated and inserted between data points for the plot of horizontal cuts. The default value is 1024. MAX100 is specified in 100ths of an inch.

IPRT — Print flag. IPRT≠0 means extra information is printed in subroutine DAGETR in the course of calculating arrival angle, beamwidth, and array signal gain. This information is used mainly to gain added insight to an unusual case. Usually IPRT=0.

ITLT — Tilt flag. ITLT=0 implies there is no tilt in the array and a fixed depth file is being used. ITLT≠0 means the array may be tilted and file FT88F001 is used.

DELTA — Vertical separation (in.) of the intensity-versus-arrival-angle cuts of the isometric plot.

HIDDEN — Variable to remove hidden points. If HIDDEN ⩽ 0.0, then hidden points on the intensity-versus-arrival-angle cuts of the isometric plot are not plotted. If HIDDEN > 0.0, then the hidden points are plotted. When HIDDEN > 0.0 is used, it is usually necessary to use DELTA > 1.0 in., so that the individual cuts can be distinguished.

SCALE — Variable to scale the vertical height of the intensity-versus-arrival angle isometric plot. If SCALE=0.0, no isometric plot is created. Otherwise the plot is scaled by a factor of SCALE.

DBN	Parameter to determine which secondary arrival angles are of interest. If the array signal gain associated with a secondary arrival angle is within DBN dB of the array signal gain of the primary arrival angle, that secondary arrival angle is printed. Up to four secondary arrival angles can be output.
CENTD	Depth (m) of the center of a tilted array. If ITLT=0, CENTD is ignored.
TILT	Tilt (deg) of a tilted array measured positive clockwise. If ITLT=0, TILT is ignored.
IHANN	Hanning shading flag. If IHANN=1, Hanning shading is used, otherwise not.
NICE	Flag to determine the plot quality. If NICE=1, plot quality is better and plots are more expensive. If NICE≠1, plots are of poorer quality but are less expensive.
ISMOTH	Flag to determine whether the plot is smoothed. If ISMOTH=0, plots are not smoothed. If ISMOTH≠0, a five-point smoothing with binomial weights is used.
BEAMIN	Minimum arrival angle (deg) to be plotted.
BEAMAX	Maximum arrival angle (deg) to be plotted.
DB3MIN	Minimum 3-dB beamwidth (deg) to be plotted.
DB3MAX	Maximum 3-dB beamwidth (deg) to be plotted.
AMPMIN	Minimum array signal gain (dB re ideal) to be plotted.
AMPMAX	Maximum array signal gain (dB re ideal) to be plotted.
	Default values for the last six inputs have been explained previously.

Formats for the Inputs to Program TRNISM

Input formats are as follows:

Card	Inputs	Format
1	FREQ, SOURD, RANMIN, RANMAX, ANGOFF, ALENG, APLTMN, APLTMAX	(8F10.3)
2	NUMHY, LIMMIN, LIMMAX, MULT, MAX100, IPRT, ITLT	(7I5)
3	DELTA, HIDDEN, SCALE, DBN, CENTD, TILT	(6F10.3)
4	IHANN, NICE, ISMOTH	(3I5)
5	BEAMIN, BEAMAX, DB3MIN, DB3MAX, AMPMIN, AMPMAX	(6F10.3)

Outputs from Program TRNISM

The outputs are as follows:

File FT06F001, the printout from TRNISM. This prints many of the entries to TRNISM and also the value of variables stored on file FT04F001. It also prints the range, bearing angle, 3-dB beamwidth, array signal gain, and any secondary arrival angles for each range step.

File FT59F001, the intensity-versus-arrival-angle isometric plot.

File FT59F002, the arrival-angle-versus-range, beamwidth-versus-range, and array-signal-gain-versus-range plots.

Use of Program TRNISM

The following is a sample deck for running TRNISM:

```
Card
 1   / JOB jobname,account,usercode,OPT=(T,R),CAT=29
 2   / LIMIT BAND=250,SEC=500
 3   / LIMIT BAND=250,SEC=100
 4   / KEYBOARD
 5   / JSLOPTS OPT=(F,L,M)
 6   / PD PE,USERCAT/D81/L60/PE3D
 7   / PD YOU,USERCAT/D--/B--/usercode
 8   / ASGP FT02F001,YOU/nodename/FT02F001,USE=SHR
 9   / ASGP FT04F001,YOU/nodename/FT04F001,USE=SHR
10   / ASGP FT88F001,YOU/nodename/FT88F001,USE=SHR
11   / ASG TRNOBJ,PE/TRNISM/OBJ,USE=SHR
12   / ASG LIB3D,PE/LIB3D/OLIB,USE=SHR
13   / DISSPLA VERS=8.2
14   / LNK LSPACE=15000,LOAD=TRNGO
15   INCLUDE TRNOBJ/
16   LIBRARY LIB3D
17   LIBRARY DISSPOBJ
18   / FD FT06F001,BAND=1/20/1
19   / FD FT08F001,BKSZ=3944,LREC=136,RCFM=FBA,BAND=1/10/1
20   / FD FT59F001,BAND=1/10/1
21   / WAIT
22   / FXQT OPT=(I),GO=TRNGO,CPTIME=6000,ADDMEM=?
      ····DATA···
23   / FOSYS FT59F002,TYPE=PLOT
24   / FOSYS FT06F001
25   / FOSYS FT59F001,TYPE=PLOT
26   / EOJ
```

To run the high-frequency version, card 13 is changed to

```
13   / ASG LIB3D,PE/HILIB/OLIB,USE=SHR
```

When a fixed-depth file is being used, that file should be assigned to FT66F001 on card 10:

```
10   / ASGP FT66F001, YOU/nodename/fixed-depth file,USE=SHR
```

Card 12 assigns the object code of TRNISM. Due to the size of the output, it is necessary to increase the size of files FT06F001 and FT59F001 beyond the default size. This is accomplished by cards 18 and 20. The other cards have been explained in earlier sections.

REFERENCES

1. P.C. Etter and R.S. Flum, Sr., "A Survey of Underwater Acoustics Models and Environmental-Acoustic Data Banks," ASW Systems Project Office Report ASWR-80-115, 1980.

2. R.N. Baer and W.B. Moseley, J. Acoust. Soc. Am. **61**, S11-S12 (A) (1977).

3. J.S. Perkins and R.N. Baer, J. Acoust. Soc. Am. **72**, 515-522 (1982).

4. J.A. Fock, *Electromagnetic Diffraction and Propagation Problems*, Pergamon, 1965.

5. S.T. McDaniel, J. Acoust. Soc. Am. **57**, 307-311 (1975) and **58**, 1178-1185 (1975), with comments by A.O. Williams, Jr., ibid. **58**, 1320-1321 (1975).

6. R.M. Fitzgerald, J. Acoust. Soc. Am. **57**, 839-842 (1975).

7. D.R. Palmer, J. Acoust. Soc. Am. **60**, 343-354 (1976).

8. A.V. Newman and F. Ingenito, "A Normal Mode Computer Program for Calculating Sound Propagation in Shallow Water with an Arbitrary Velocity Profile," NRL Memorandum Report 2381, Jan. 1972.

9. J.F. Miller and S.N. Wolf, "Modal Acoustic Transmission Loss (MOATL): A Transmission-Loss Computer Program Using a Normal-Mode Model of the Acoustic Field in the Ocean," NRL Report 8429, Aug. 1980.

10. J.S. Perkins, R.N. Baer, E.B. Wright, and L.F. Roche, "Solving the Parabolic Equation for Underwater Acoustic Propagation by the Split-Step Algorithm," NRL Report 8607, Aug. 1982.

11. R.J. Urick, *Principles of Underwater Sound (Second Edition)*, McGraw-Hill, 1975.

Appendix A
DERIVATION OF EQ. (4)

The following derivation of Eq. (4) is an extension of the procedure followed by Jensen and Krol* for the two-dimensional problem. The index of refraction n is assumed to be constant.

First transform Eq. (3) in depth:

$$\int_{-\infty}^{\infty}\left[\Phi_{zz} + 2ik_0\,\Phi_r + \frac{\Phi_{\theta\theta}}{r^2} + k_0^2\,(n^2 - 1)\Phi\right]e^{-ilz}\,dz = 0. \tag{A1}$$

Integrate the Φ_{zz} term twice by parts and assume Φ and Φ_z go smoothly to zero as $z \rightarrow \pm\infty$. Then

$$-l^2\hat{\Phi} + 2ik_0\hat{\Phi}_r + \frac{\hat{\Phi}_{\theta\theta}}{r^2} + k_0^2\,(n^2 - 1)\hat{\Phi} = 0, \tag{A2}$$

where $\hat{\Phi}(r,l,\theta) = \int_{-\infty}^{\infty}\Phi e^{-ilz}\,ds$ is the depth transform of Φ.

Now transform Eq. (A2) in angle and integrate by parts as before. We assume that $\hat{\Phi}$ and $\hat{\Phi}_\theta$ go smoothly to zero as $\theta \rightarrow \pm\pi$. Then

$$2ik_0\tilde{\Phi}_r + \left[k_0^2\,(n^2 - 1) - l^2 - \frac{m^2}{r^2}\right]\tilde{\Phi} = 0, \tag{A3}$$

where $\tilde{\Phi}(r,l,m) = \int_{-\pi}^{\pi}\hat{\Phi}(r,l,\theta)e^{-im\theta}\,d\theta$ is the angle transform of $\hat{\Phi}$.

Solve Eq. (A3), a first-order equation, to get

$$\tilde{\Phi}(r,l,m) = C_1\,\exp\left\{\frac{i}{2k_0}\left[k_0^2\,(n^2 - 1)r - l^2 r + \frac{m^2}{r}\right]\right\}. \tag{A4}$$

From this formula, note that

$$\tilde{\Phi}(r + \Delta r,l,m) = \tilde{\Phi}(r,l,m)\exp\left\{\frac{i\Delta r}{2k_0}\left[k_0^2\,(n^2 - 1) - l^2 - \frac{m^2}{r(r + \Delta r)}\right]\right\}. \tag{A5}$$

Inverse-transform Eq. (A5) in depth and angle to obtain Eq. (4).

As in the two-dimensional case, the limitation imposed by the assumption that n is constant is not as severe as might be expected. The error inherent in solving the parabolic equation using this split-step technique can be made arbitrarily small by choosing Δr, the range step, sufficiently small.

---

*F. Jensen and H. Krol, "The Use of the Parabolic Equation Method in Sound Propagation Modelling," SACLANTCEN Memorandum SM-72, La Spezia, Italy, 15 Aug. 1975.

Appendix B
EXAMPLE CASE

The following is an example execution of the NRL 3D PE model. The example problem is test case 1B from the AESD Workshop on Acoustic-Propagation Modeling by Nonray-Tracing Techniques,* except that the sound-speed field has been given a range and azimuth dependence. Also, the example problem is similar to that in a previous NRL report on a 2D PE model.† It indicates the capability of the NRL 3D PE model but is not meant to represent a real ocean environment.

Figure B1 depicts the region over which the acoustic field is propagated. The ocean bottom is assumed to be flat at a depth of 5000 m. Profile A is a characterization of a typical North Pacific sound-speed structure adjusted to include a very deep surface duct. At each depth the sound speed of profile B is 50 m/s less than that of profile A at the corresponding depth. Similarly, the sound speeds of profile C are 50 m/s greater than those of profile A. The acoustic frequency is 25.0 Hz, and 256 (2^8) depth points and 16 (2^4) azimuthal angles have been specified.

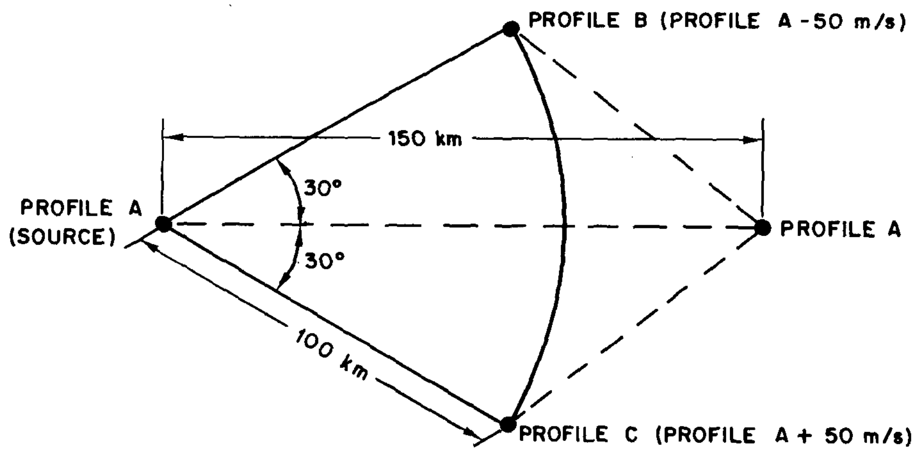

Fig. B1 — Geometry of the sound-speed field

Figures B2 through B14 present the job decks (JSL commands and card-image input data) and associated plots for each of the six programs comprising the NRL 3D PE model. Figures B2 and B3 are the job decks used to run the environment construction programs PRO3D and BOT3D respectively. Due to the coarseness of the azimuthal sampling of the sound-speed field in this case, the contour plot is not illustrative and hence is not presented. Figure B4 presents the job deck used to run the startup program, START3, and Fig. B5 is the resulting plot of the sound-speed profile at the source (profile A) and the initial acoustic field. For this example the startup field was calculated from normal modes.

Figure B6 is the job deck to run the split-step propagation program SPLCYL. Figure B7 is the job deck used to run the transmission-loss program TL3D; and Figs. B8 through B10 are the resulting plots of transmission loss.

Figure B11 is the job deck used to run the array-performance program, TRNISM; and Figs. B12 through B14 are the resulting plots of bearing angle, 3-dB width, and array signal gain respectively.

---

*C.W. Spofford, "A Synopsis of the AESD Workshop on Acoustic-Propagation Modeling by Non-Ray-Tracing Techniques, 22-25 May 1973," Acoustic Environmental Support Detachment, Office of Naval Research, Technical Note TN-73-05, Nov. 1973, p. 14.
†J.S. Perkins, R.N. Baer, E.B. Wright, and L.F. Roche, "Solving the Parabolic Equation for Underwater Acoustic Propagation by the Split-Step Algorithm," NRL Report 8607, Aug. 1982.

```
/ JØB JØBNAME,ACCT#,USERCØDE,ØPT=(T,R),CAT=9
/ LIMIT BAND=50,SEC=500
/ LIMIT BAND=50,SEC=100
/ KEYBØARD
/ JSLØPTS ØPT=(F,L,M)
/ PD PE,USERCAT/D81/L60/PE3D
/ PD ME,USERCAT/...
/ ASGP FT03F001,ME/DØCTEST/FT03F001,USE=SHR
/ ASGP LIB3D,PE/HILIB/ØLIB,USE=SHR
/ ASGP CØNLIB,USERCAT/D81/L60/CØNLIB/ØLIB,USE=SHR
/ ASGP PRØØBJ,PE/PRØ3D/ØBJ,USE=SHR
/ DISSPLA VERS=8.2
/ WAIT
/ LNK LSPACE=15000,LØAD=PRØGØ
INCLUDE PRØØBJ/
LIBRARY LIB3D
LIBRARY CØNLIB
LIBRARY DISSPØBJ
/ REL SYS.PRT
/ FD FT06F001,BAND=1/20/1
/ FXQT ØPT=(I),GØ=PRØGØ,CPTIME=10000
    0
     0.000    100.000    -30.000     30.000
    10.000     10.000     10.000
     3.000      3.000      3.000
   11   11
    4    2    1
    5     0.000      0.000
    0.00 1536.50   152.40 1539.24   406.30 1501.14 1015.90 1471.88 5587.90 1549.60
    5   150.000      0.000
    0.00 1536.50   152.40 1539.24   406.30 1501.14 1015.90 1471.88 5587.90 1549.60
    5   100.000    -30.000
    0.00 1486.45   152.40 1489.24   406.30 1501.14 1015.90 1421.88 5587.90 1499.60
    5   100.000     30.000
    0.00 1586.50   152.40 1589.24   406.30 1551.14 1015.90 1521.88 5587.90 1599.60
    1    2    3    1    2    4
    0    1
  863.498    -30.000    30.000     0.000    100.000   1420.000   1550.000    14    10
/ CAT ME/DØCTEST/FT02F001,ACNM=FT02F001
/ CAT ME/DØCTEST/PRØPRT,ACNM=FT06F001
/ CAT ME/DØCTEST/PRØPLØT,ACNM=FT59F001
/ FØSYS FT06F001
/ FØSYS FT59F001,TYPE=PLØT
/ EØJ
```

Fig. B2 — Sample job deck for program PRO3D

```
/ JØB JØBNAME,ACCT#,USERCØDE,CAT=5,ØPT=(T,R)
/ LIMIT BAND=50,SEC=500
/ LIMIT BAND=50,SEC=100
/ KEYBØARD
/ JSLØPTS ØPT=(F,L,M)
/ PD PE,USERCAT/D81/L60/PE3D
/ PD ME,USERCAT/...
/ DISSPLA VERS=8.2
/ ASGP LIB3D,PE/LIB3D/ØLIB,USE=SHR
/ ASGP BØTØBJ,PE/BØT3D/ØBJ,USE=SHR
/ WAIT
/ LNK LSPACE=15000,LØAD=BØTGØ
INCLUDE BØTØBJ/
LIBRARY LIB3D
LIBRARY DISSPØBJ
/ REL SYS.PRT
/ FXQT ØPT=(I),GØ=BØTGØ,CPTIME=6000
    0
  5587.898
/ CATN ME/DØCTEST
/ CAT ME/DØCTEST/FT03F001,ACNM=FT03F001
/ CAT ME/DØCTEST/BØTPRT,ACNM=FT06F001
/ FØSYS FT06F001
/ EØJ
```

Fig. B3 — Sample job deck for program BOT3D

```
/ JØB JØBNAME,ACCT#,USERCØDE,ØPT=(T,R),CAT=5
/ LIMIT BAND=50,SEC=400
/ LIMIT BAND=50,SEC=200
/ KEYBØARD
/ JSLØPTS ØPT=(F,L,M)
/ PD PE,USERCAT/D81/L60/PE3D
/ PD ME,USERCAT/...
/ ASGP FT02F001,ME/DØCTEST/FT02F001,USE=SHR
/ ASGP FT03F001,ME/DØCTEST/FT03F001,USE=SHR
/ ASGP LIB3D,PE/LIB3D/ØLIB,USE=SHR
/ ASGP STARTØBJ,PE/START3/ØBJ,USE=SHR
/ DISSPLA VERS=8.2
/ WAIT
/ LNK LSPACE=15000,LØAD=STARTGØ
INCLUDE STARTØBJ/
LIBRARY LIB3D
LIBRARY DISSPØBJ
/ REL SYS.PRT
/ FD FT06F001,BAND=1/20/1
/ FXQT ØPT=(I),GØ=STARTGØ,CPTIME=6000
    0    1    8    4     25.000          .250  1555.522    253.898
    0    4  100       1.000       1.917        .001
/ CAT ME/DØCTEST/FT21F001,ACNM=FT21F001
/ CAT ME/DØCTEST/STRTPRT,ACNM=FT06F001
/ CAT ME/DØCTEST/STRTPLØT,ACNM=FT59F001
/ FØSYS FT06F001
/ FØSYS FT59F001,TYPE=PLØT
/ EØJ
```

Fig. B4 — Sample job deck for program START3

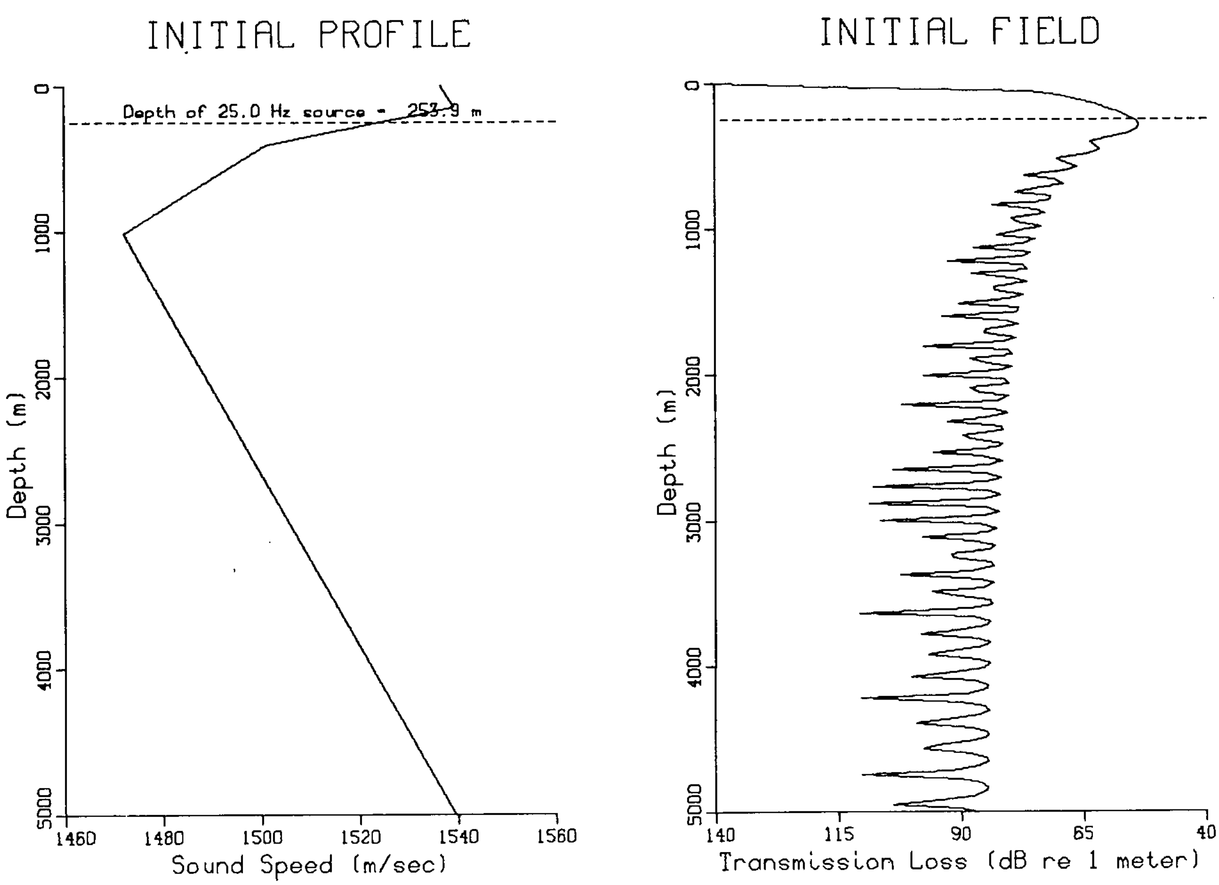

Fig. B5 — Plot generated by program START3

```
/ JOB JOBNAME,ACCT#,USERCODE,OPT=(T,R),CAT=89
/ LIMIT BAND=300,SEC=100
/ LIMIT BAND=300,SEC=200
/ KEYBOARD
/ JSLOPTS OPT=(F,L,M)
/ PD PE,USERCAT/D81/L60/PE3D
/ PD ME,USERCAT/...
/ ASGP FT02F001,ME/DOCTEST/FT02F001,USE=SHR
/ ASGP FT03F001,ME/DOCTEST/FT03F001,USE=SHR
/ ASGP FT21F001,ME/DOCTEST/FT21F001,USE=SHR
/ ASGP SPLOBJ,PE/SPLCYL/OBJ,USE=SHR
/ ASGP LIB3D,PE/LIB3D/OLIB,USE=SHR
/ FD FT08F001,BKSZ=3944,LREC=136,RCFM=FBA,BAND=1/10/1
/ FD FT06F001,BAND=1/20/1
/ WAIT
/ LNK LOAD=SPLCYLGO
INCLUDE SPLOBJ/
LIBRARY LIB3D
/ REL SYS.PRT
/ FXQT OPT=(I),GO=SPLCYLGO,CPTIME=20000,ADDMEM=25K
    1     1
    0
    3     8     4     1     2     1
  100.000        .250      60.000       0.000     -1.000     -1.E-10
  254.000        .000     863.500        .000   1500.000        .000
/ MFR TU78F4
/ CAT ME/DOCTEST/FT88F001,ACNM=FT88F001,DTYP=TAPE
/ CAT ME/DOCTEST/FT04F001,ACNM=FT04F001,DTYP=TAPE
/ CAT ME/DOCTEST/SPLPRT,ACNM=FT06F001,DTYP=TAPE
/ MFRE
/ FOSYS FT06F001
/ EOJ
```

Fig. B6 — Sample job deck for program SPLCYL

```
/ JOB JOBNAME,ACCT#,USERCODE,OPT=(T,R),CAT=25
/ LIMIT BAND=300,SEC=400
/ LIMIT BAND=300,SEC=200
/ KEYBOARD
/ JSLOPTS OPT=(F,L,M)
/ PD PE,USERCAT/D81/L60/PE3D
/ PD ME,USERCAT/...
/ ASGP FT88F001,ME/DOCTEST/FT88F001,USE=SHR
/ ASGP FT04F001,ME/DOCTEST/FT04F001,USE=SHR
/ ASGP TLOBJ,PE/TL3D/OBJ,USE=SHR
/ ASGP LIB3D,PE/LIB3D/OLIB,USE=SHR
/ DISSPLA VERS=8.2
/ WAIT
/ LNK LSPACE=15000,LOAD=TLGO
INCLUDE TLOBJ/
LIBRARY LIB3D
LIBRARY DISSPOBJ
/ REL SYS.PRT
/ FXQT CPTIME=12000,OPT=(I),GO=TLGO,ADDMEM=25K
    1     1     1     0     1     1
1   863.498     0.000     0.000   100.000   20.00    1.00    0.00    0.00
1    60.000     0.000     0.000  5000.000  500.00  100.00    0.00    0.00
1   100.000   863.498   -20.000    20.000    5.00    7.50    0.00    0.00
  863.498
    0.000
/ MFR HJKS
/ CAT ME/DOCTEST/TLPRT,ACNM=FT06F001,DTYP=TAPE
/ CAT ME/DOCTEST/TLPLOT,ACNM=FT59F001,DTYP=TAPE
/ CAT ME/DOCTEST/FILE66,ACNM=FT66F001,DTYP=TAPE
/ CAT ME/DOCTEST/FILE67,ACNM=FT67F001,DTYP=TAPE
/ MFRE
/ FOSYS FT06F001
/ FOSYS FT59F001,TYPE=PLOT
/ EOJ
```

Fig. B7 — Sample job deck for program TL3D

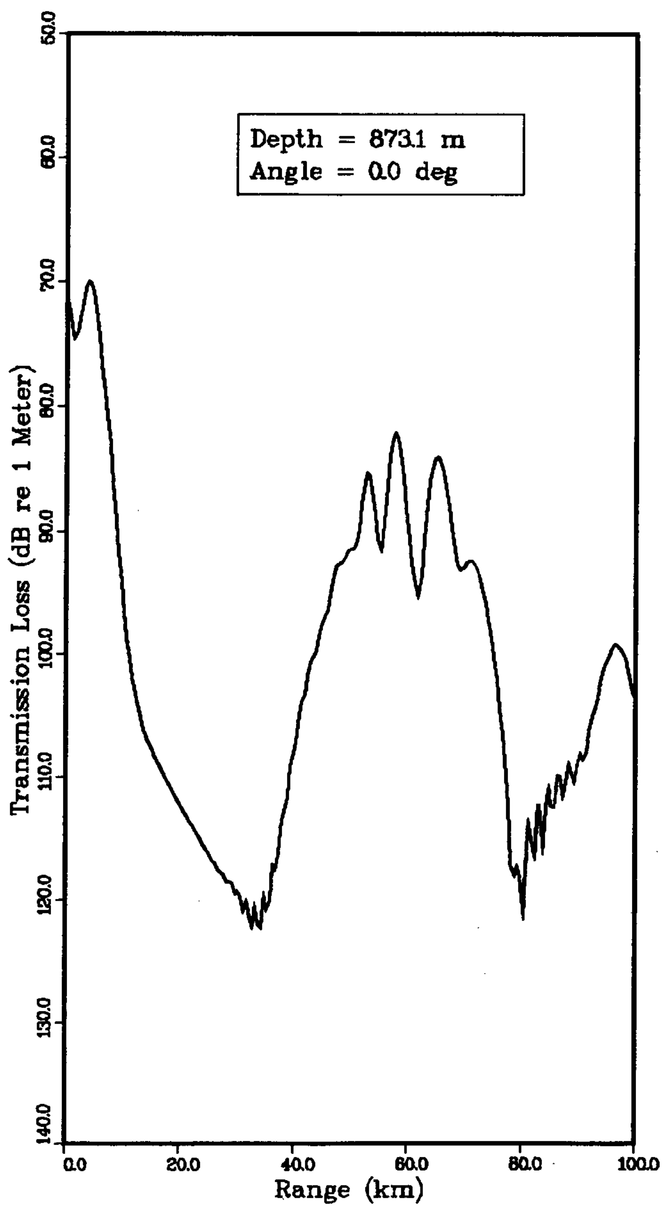

Fig. B8 — Program-TL3D-generated plot of transmission loss versus range, at a fixed depth and azimuth

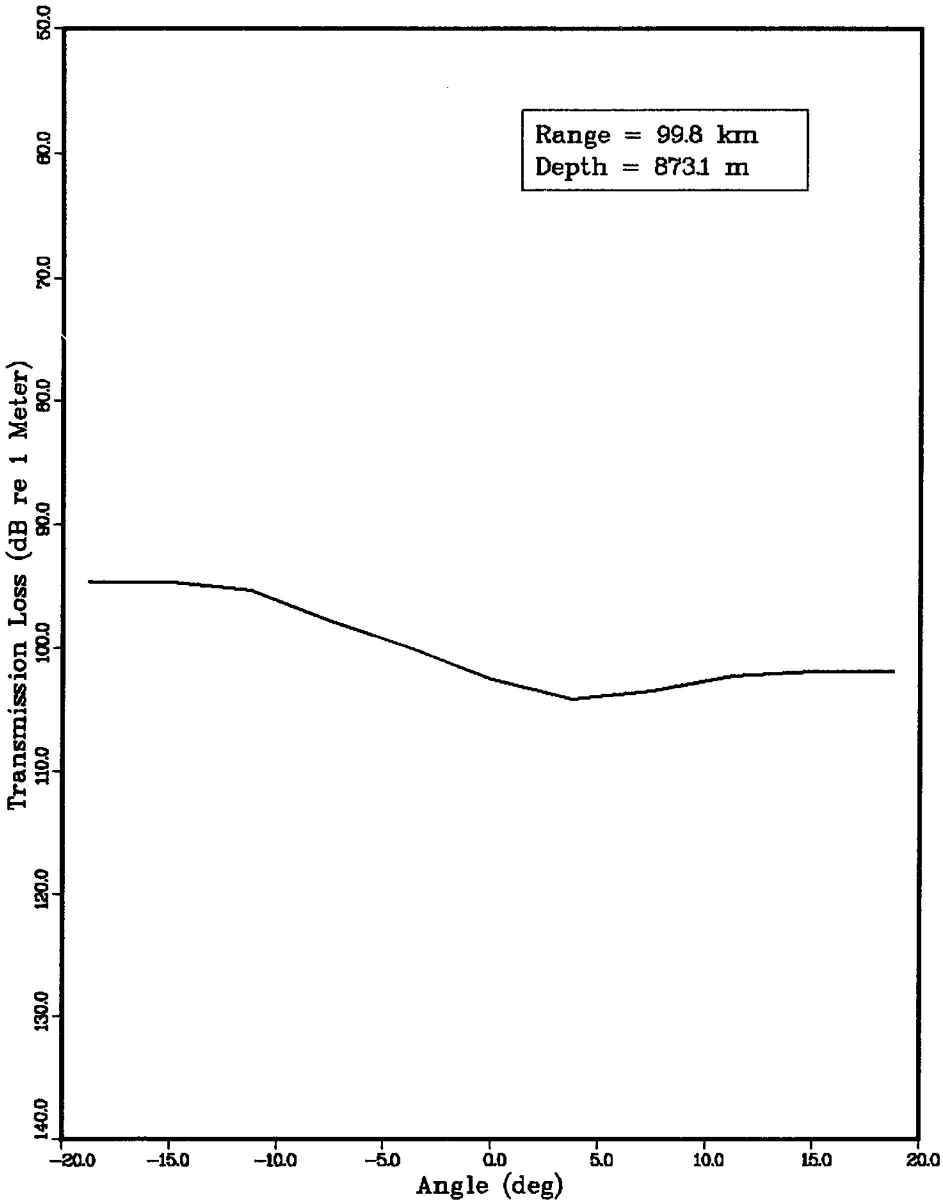

Fig. B9 — Program-TL3D-generated plot of transmission loss versus azimuth, at a fixed range and depth

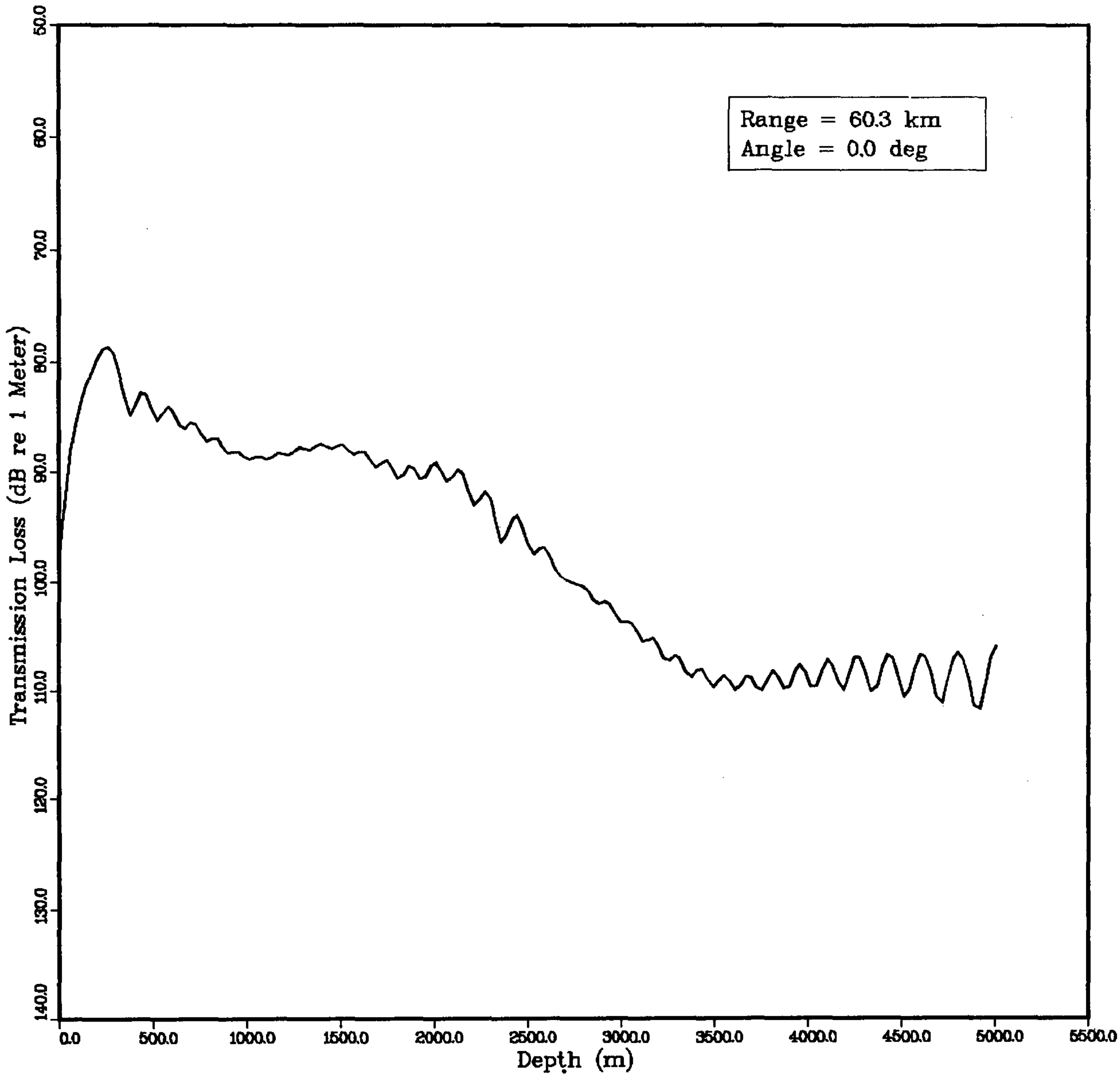

Fig. B10 — Program-TL3D-generated plot of transmission loss versus depth, at a fixed range and azimuth

```
/ JOB JOBNAME,ACCT#,USERCODE,OPT=(T,R),CAT=29
/ LIMIT BAND=250,SEC=500
/ LIMIT BAND=250,SEC=100
/ KEYBOARD
/ JSLOPTS OPT=(F,L,M)
/ PD PE,USERCAT/D81/L60/PE3D
/ PD ME,USERCAT/...
/ ASGP FT02F001,ME/DOCTEST/FT02F001,USE=SHR
/ ASGP FT04F001,ME/DOCTEST/FT04F001,USE=SHR
/ ASGP FT66F001,ME/DOCTEST/FILE66,USE=SHR
/ ASGP TRNOBJ,PE/TRNISM/OBJ,USE=SHR
/ ASGP LIB3D,PE/LIB3D/OLIB,USE=SHR
/ DISSPLA VERS=8.2
/ WAIT
/ LNK LSPACE=15000,LOAD=TRNGO
INCLUDE TRNOBJ/
LIBRARY LIB3D
LIBRARY DISSPOBJ
/ FXQT OPT=(I),GO=TRNGO,CPTIME=6000,ADDMEM=25000
    25.000     253.898      10.000    100.000    -10.000       0.000      0.000      5.000
  128    0     0    4     0    0     0    0
     0.080      0.000       2.000     10.000    253.898       0.000
    0    1     1
     0.000     10.000       0.000     10.000    -10.000       0.000
    0    0     0    0
/ CAT ME/DOCTEST/TRPRT,ACNM=FT06F001
/ CAT ME/DOCTEST/TRPLOT1,ACNM=FT59F001
/ CAT ME/DOCTEST/TRPLOT2,ACNM=FT59F002
/ FOSYS FT06F001
/ FOSYS FT59F001,TYPE=PLOT
/ EOJ
```

Fig. B11 — Sample job deck for program TRNISM

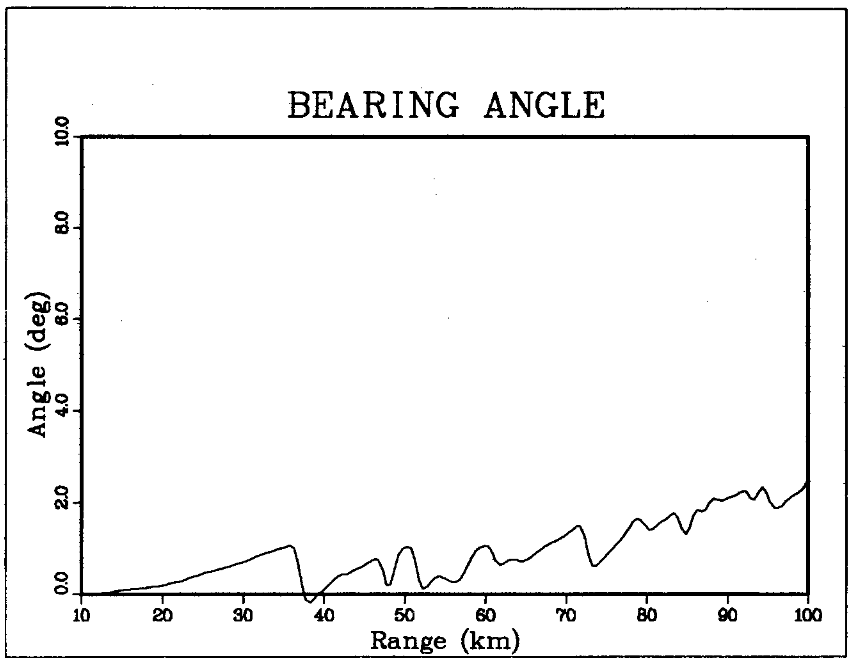

Fig. B12 — Program-TRNISM-generated plot of bearing angle

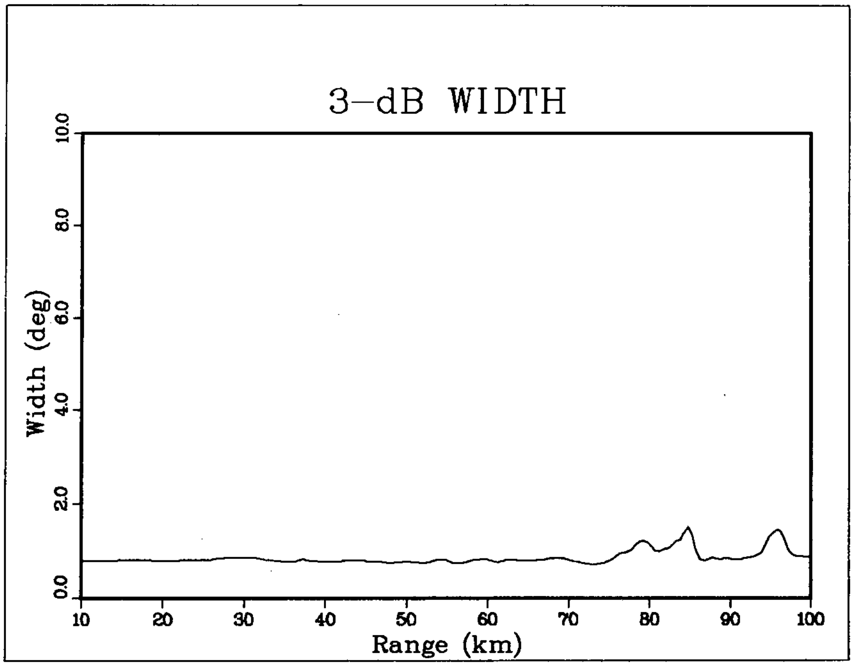

Fig. B13 — Program-TRNISM-generated plot of 3-dB width

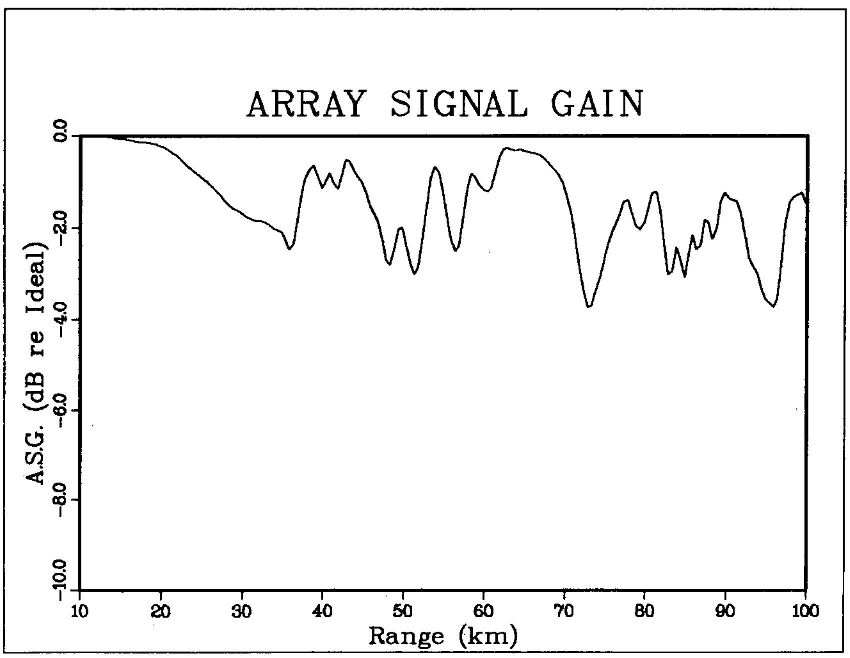

Fig. B14 — Program-TRNISM-generated plot of array signal gain