COMPUTER PROGRAMS FOR THE AUSEX (AIRCRAFT UNDERSEA SOUND EXPERIMENT) AIR-WATER ACOUSTIC PROPAGATION MODEL

BOLT BERANEK AND NEUMAN, INCORPORATED

PREPARED FOR
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

28 JANUARY 1976
BEST AVAILABLE COPY
COMPUTER PROGRAMS FOR THE AUSEX AIR-WATER ACOUSTIC PROPAGATION MODEL

D. Sachs
L. Sledjeski
R. Stern

Contract N00014-75-C-0532
ARPA Order 2909, MOD #2, 7/7/75
BBN Job 10054
BBN Technical Memorandum W307

28 January 1976

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. N00014-75-C-0532.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

Prepared for:

Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

Attention: LCDR Wesley Jordan
Tactical Technology Office

Prepared by:

Bolt Beranek and Newman Inc.
1701 North Fort Myer Drive
Arlington, Virginia 22209
This report is a user's guide for a computerized mathematical model development in support of the Aircraft Undersea Sound Experiment (AUSEX) Program. The AUSEX Program is funded and directed by the Defense Advanced Research Projects Agency (DARPA).

The subject mathematical model is termed the AUSEX Acoustic Propagation Model and it describes the acoustic propagation from a moving sound source in air, to and across a rough air-water interface, and subsequently through the water to an arbitrarily located point-acoustic receiver.

This report describes the computer program architecture and the input and output data associated with the program's use. Two examples of the program's application are included. In addition, the complete program listing for the Fortran IV coding is included.

A description of the mathematical basis for the Acoustic Propagation Model is contained in a companion report, BBN TM W311, "AUSEX Air-Water Acoustic Program Model."
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NOL/WT</td>
<td>NOL/WT</td>
<td>NOL/WT</td>
</tr>
<tr>
<td>Acoustic Propagation Model</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Air-Water Interface</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Program</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
COMPUTER PROGRAMS FOR THE AUSEX AIR-WATER ACOUSTIC PROPAGATION MODEL

D. Sachs  
L. Sledjeski  
R. Stern  

Contract N00014-75-C-0532  
ARPA Order 2909, MOD #2, 7/7/75  
BBN Job 10054  
BBN Technical Memorandum W307

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract N0. N00014-75-C-0532.

28 January 1976

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

Prepared for:  
Defense Advanced Research Projects Agency  
1400 Wilson Boulevard  
Arlington, Virginia 22209  
Attention: LCDR Wesley Jordan  
Tactical Technology Office

Prepared by:  
Bolt Beranek and Newman Inc.  
1701 North Fort Myer Drive  
Arlington, Virginia 22209
ABSTRACT

This report is a user's guide for a computerized mathematical model development in support of the Aircraft Undersea Sound Experiment (AUSEX) Program. The AUSEX Program is funded and directed by the Defense Advanced Research Projects Agency (DARPA).

The subject mathematical model is termed the AUSEX Acoustic Propagation Model and it describes the acoustic propagation from a moving sound source in air, to and across a rough air-water interface, and subsequently through the water to an arbitrarily located point-acoustic receiver.

This report describes the computer program architecture and the input and output data associated with the program's use. Two examples of the program's application are included. In addition, the complete program listing for the Fortron IV coding is included.

A description of the mathematical basis for the Acoustic Propagation Model is contained in a companion report, BBN TM W311, "AUSEX Air-Water Acoustic Program Model".
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>11</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>iv</td>
</tr>
<tr>
<td>SECTION 1: INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Overview and Objective of the Report</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Report Organization</td>
<td>3</td>
</tr>
<tr>
<td>SECTION 2: PROGRAM SUMMARY</td>
<td>4</td>
</tr>
<tr>
<td>2.1 Intended Utilization</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Program Architecture</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Input Data</td>
<td>9</td>
</tr>
<tr>
<td>2.4 Output Data</td>
<td>9</td>
</tr>
<tr>
<td>SECTION 3: TYPICAL RESULTS</td>
<td>18</td>
</tr>
<tr>
<td>3.1 Representative North Atlantic Environmental Conditions</td>
<td>13</td>
</tr>
<tr>
<td>3.2 Rough Air-Water Interface</td>
<td>29</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>36</td>
</tr>
<tr>
<td>APPENDIX A: Surface Duct and Convergence Zone Subroutine</td>
<td>A-1</td>
</tr>
<tr>
<td>APPENDIX B: Direct Path and Bottom Bounce Subroutine</td>
<td>B-1</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE NO.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Principal Elements of the AUSEX Program</td>
<td>2</td>
</tr>
<tr>
<td>2-1</td>
<td>Schematic of Propagation Model Program</td>
<td>8</td>
</tr>
<tr>
<td>2-2</td>
<td>Schematic of Surface and Source/Receiver Variables</td>
<td>12</td>
</tr>
<tr>
<td>2-3</td>
<td>Schematic of Deep Ocean Sound Speed Profiles Permitted in Program</td>
<td>13</td>
</tr>
<tr>
<td>3-1</td>
<td>Time Histories of Transmission Loss and Normalized Receiver Frequency, Summer Profile</td>
<td>22</td>
</tr>
<tr>
<td>3-2</td>
<td>Time Histories of Arrival Angles, Summer Profile</td>
<td>23</td>
</tr>
<tr>
<td>3-3</td>
<td>Time Histories of Transmission Loss and Normalized Received Frequency, Winter Profile</td>
<td>27</td>
</tr>
<tr>
<td>3-4</td>
<td>Time Histories of Arrival Angles, Winter Profile</td>
<td>28</td>
</tr>
<tr>
<td>3-5(a)</td>
<td>Transmission Loss Time History, High Sea State</td>
<td>30</td>
</tr>
<tr>
<td>3-5(b)</td>
<td>Time Histories of Arrival Angle and Received Frequency, High Sea State</td>
<td>31</td>
</tr>
<tr>
<td>A-1</td>
<td>Block Data Format</td>
<td>A-23</td>
</tr>
<tr>
<td>B-1</td>
<td>Block Diagram Showing Flow of Calculations (Direct Path and Bottom Bounce Subroutines)</td>
<td>B-3</td>
</tr>
</tbody>
</table>
### LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE NO.</th>
<th>page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1 Program Parameter Space</td>
<td>6-7</td>
</tr>
<tr>
<td>2-2 Propagation Model Program Inputs</td>
<td>10-11</td>
</tr>
<tr>
<td>2-3 Program Outputs</td>
<td>14-15</td>
</tr>
<tr>
<td>3-1 Example Input Parameters</td>
<td>19-20</td>
</tr>
<tr>
<td>3-2 For Given Sea State, Transmission Gain Relative to a Smooth Interface</td>
<td>32</td>
</tr>
<tr>
<td>4-1 Outline of Subroutines (Surface Duct and Convergence Zone)</td>
<td>A-5</td>
</tr>
</tbody>
</table>
SECTION 1
INTRODUCTION

1.1 Overview and Objective of the Report

The Defense Advanced Research Projects Agency (DARPA) has been sponsoring a research and development program termed AUSEX (Aircraft Undersea Sound Experiments). Briefly, the objective of the AUSEX Program is the development of generic algorithms for the detection, classification, and tracking of air vehicles by their underwater acoustic signatures.

The principal elements and their functional relationship are indicated in Fig. 1-1. One of the primary inputs to the detection, classification, tracking algorithm design is the outputs from the air-water acoustic propagation model. This model was designed to provide the algorithm designer with a detailed description of the signal field at a point in the ocean as a function of time, the signal being the radiated acoustic energy associated with an air vehicle (i.e., fixed and rotary wing aircraft and cruise missiles).

Other AUSEX Program elements have been concerned with the development of detailed acoustic source characterizations of air vehicles of interest. And another dealt with the quantification of the effects of real ocean surfaces on the propagation of sound from air to water.

The air-water propagation model was computer programmed as a tool for the detection algorithm designers. This technical memorandum is intended to serve as a user's guide and as such

#Superscript numbers identify references as listed on page R-1.
it describes the computer program, its required input, and subsequent output. Typical examples of its utilization are given. The program listing in Fortran IV for the BBN TENEX System is also included. A description of the mathematical basis for the Propagation Model is given in BBN Technical Memorandum TM W311, "AUSEX Air-Water Program Model". This report is in publication.

1.2 Report Organization

Section 2 is a summary of the overall air-water acoustic propagation program. The program is designed as an executive with several subroutines which concern such things as the air-water interface model, the several distinct underwater acoustic propagation modes, the atmospheric acoustic propagation, and the dynamics of the air-source underwater-receiver encounter.

Section 3 displays some typical results in manually plotted form. In addition, this section includes some discussion of the individual results for the example cases chosen.
SECTION 2
PROGRAM SUMMARY

2.1 Intended Utilization

The ultimate objective of the AUSEX Program is the development of generic algorithms for the detection, classification, and tracking of air vehicles by their underwater acoustic signatures. The AUSEX air-water acoustic propagation model is intended to provide the detection algorithm designer with a detailed, time dependent description of the acoustic field produced at a (moving) point in the ocean by an air vehicle. The model output provides a primary information base which the designer will utilize to construct and optimize detection/classification/tracking algorithm schemes.

The user specifies atmospheric and ocean environmental parameters and air vehicle and receiver track parameters. The model code then marches through time as the encounter unfolds, calculating time histories of the following quantities:

- Transmission loss from air vehicle to the receiver for each underwater mode of propagation (i.e., direct path, bottom bounce, surface duct and convergence zone).
- Depression/elevation and azimuthal arrival angles for each propagation mode.
- Received frequency by propagation mode.
- Range variables.

In obtaining these variables, the model accounts for atmospheric propagation, air-water interface transmission, underwater propagation and source/receiver geometry dynamics.
The parameter space covered by the model is given in Table 2-1.

The family of model outputs for anticipated scenarios provides a hypothesis space for the algorithm designer to use in the synthesis of detection schemes.

2.2 Program Architecture

Figure 2-1 shows the basic program structure. There are two independent programs, one of which calculates direct path and bottom bounce outputs, while the other computes convergence zone and surface duct quantities.

For each of the programs, the input data is inserted and the program begins at a user-specified initial time and calls upon each of a pair of propagation subroutines, in turn. The subroutines are almost completely self-contained and mutually independent. (The subroutines are preceded by a sub-program which calculates a small number of common quantities.) Each of the propagation subroutines calculates all the pertinent output for only one type of underwater propagation mode. Incorporated in the subroutines are atmospheric propagation, air-water interface transmission, underwater propagation and source-receiver geometry dynamics. For a given time instant, each of the subroutines outputs time variables (emission time, arrival time), transmission loss, depression/elevation and azimuthal arrival angles, received frequency and range variables (range at emission time, range at arrival time). The time is then increased by a user-specified increment and the calculation process repeated until the incremented time variable exceeds a user-specified final time.
TABLE 2-1
PROGRAM PARAMETER SPACE

<table>
<thead>
<tr>
<th>Air source parameters</th>
<th>Receiver parameters</th>
<th>Range interval</th>
<th>Sea state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character:</td>
<td>point, omnidirectional</td>
<td>Closest approach to inner edge of second convergence zone.</td>
<td></td>
</tr>
<tr>
<td>Speed:</td>
<td>any speed not large compared to the sound speed in air</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heights:</td>
<td>any</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lateral offsets:</td>
<td>&gt; 0 yards</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radiation frequency:</td>
<td>&gt; 0 - 10,000 Hz</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Track:</td>
<td>straight line motion at constant height in any direction</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>point, omnidirectional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speed:</td>
<td>small compared with water sound speed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Depths:</td>
<td>any small fraction of ocean depth (for instance, ≤ 600 ft)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Track:</td>
<td>straight line motion at constant depth</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>fully developed or smooth (partially arisen seas and swells not included)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wind speed:</td>
<td>any</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wind direction:</td>
<td>any</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 2-1 continued.

**Atmosphere**

Air temperature: \(-10^\circ F \text{ to } 90^\circ F\)

**Ocean**

Bottom: flat

Bottom types: LRAPP types 3 and 5

Sound speed profiles: any two or three linearly segmented profile with a depth excess approximating a deep ocean winter profile (three segments) or summer profile (two segments)

Depth: deep ocean depths

shallow depths out to horizontal ranges where triple bottom bounce dominates
INPUT
1. ENVIRONMENTAL PARAMETERS
2. SOURCE AND RECEIVER TRACK PARAMETERS
3. SOURCE FREQUENCY
4. TIME STEP SIZE

\[ t = t_{\text{initial}} \]

PROPAGATION SUBROUTINES
DIRECT PATH AND BOTTOM BOUNCE OR CONVERGENCE ZONE AND SURFACE DUCT

PRINT
1. TIME VARIABLES
2. TRANSMISSION LOSS BY MODE
3. D/E AND AZIMUTHAL ARRIVAL ANGLE BY MODE
4. RECEIVED FREQUENCY BY MODE
5. RANGE

\[ t = t + \Delta t \]

\[ t > t_{\text{final}} \] ?

No

Yes

STOP

FIGURE 2-1 Schematic of Propagation Model Program
Initial and final time and time increments are selected by the user on the basis of the range interval to be covered and the temporal detail required.

No computer plotting capabilities are included in the current model code; all results must be hand plotted.

The program is written in FORTRAN IV. It is operational on the Interactive Sciences Corporation system and the BBN TENEX system.

Detailed descriptions of the component programs are given in Appendices A and B.

2.3 Input Data

The program accepts and inputs the quantities listed in Table 2-2. Refer to Figs. 2-2 and 2-3 for a pictorial definition of the variables.

The air source/receiver/time inputs define the encounter scenario and radiation frequency of the source. The atmospheric inputs quantify atmospheric propagation effects and the character of the air-water interface. Sound velocity profiles and bottom reflection losses are specified by the ocean medium input group.

Appendices A and B contain detailed descriptions of the input data format.

2.4 Output Data

Table 2-3 lists the major program outputs. See Fig. 2-2 for a schematic definition of some of the output variables.

For each value of absolute time, the output is grouped by propagation mode.
TABLE 2-2
PROPAGATION MODEL PROGRAM INPUTS

<table>
<thead>
<tr>
<th>Times</th>
</tr>
</thead>
<tbody>
<tr>
<td>starting time of encounter (in secs)</td>
</tr>
<tr>
<td>final time (in secs)</td>
</tr>
<tr>
<td>step size in time (in secs)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Air Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_x$, $v_y$ — cartesian components, in kts, of air source velocity</td>
</tr>
<tr>
<td>$x_{ia}$, $y_{ia}$ — vector position of air source, in ft, at time $t = 0$ (&quot;initial coordinates&quot;)</td>
</tr>
<tr>
<td>$h$ — height, in ft, of air source</td>
</tr>
<tr>
<td>$f$ — frequency of air source, in Hz</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$ — depth (ft)</td>
</tr>
<tr>
<td>$v_s$ — receiver speed along $x$-axis (in kts)</td>
</tr>
<tr>
<td>$x_{is}$ — $x$-position of receiver (in ft) at time $t = 0$ (&quot;initial coordinates&quot;)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Atmosphere</th>
</tr>
</thead>
<tbody>
<tr>
<td>air temp. — either greater or less than 50°F</td>
</tr>
<tr>
<td>$U$ — wind speed (kts)</td>
</tr>
<tr>
<td>$n_x$, $n_y$ — cartesian direction cosines of wind velocity vector (dimensionless)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ocean Medium</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_s$ — surface sound speed (ft/sec)</td>
</tr>
<tr>
<td>$c_{min}$, $d_{min}$ — sound speed at sound speed profile minimum (ft/sec), depth of profile minimum (ft)</td>
</tr>
</tbody>
</table>
Table 2-2 continued.

- $c_b, D_B$ — sound speed at ocean bottom (ft/sec), depth to bottom (ft)
- $c_{max}, d_s$ — sound speed at bottom of surface duct (if one exists) (ft/sec), depth of surface duct (ft)
- Bottom type — either LRAPP type 3 or 5
FIGURE 2-2  Schematic of Surface and Source/Receiver Variables.
FIGURE 2-3  Schematic of Deep Ocean Sound Speed Profiles Permitted in Program. (a) is a Representative Summer Profile with no Surface Duct. (b) is a Representative Winter Profile with a Surface Duct. A Depth Excess is Always Assumed ($c_b > c_S$ in (a), $c_b > c_{MAX}$ in (b)). When a Surface Duct is Present, $c_{MIN}$ is always Taken to be less than any Other Sound Speed.
TABLE 2-3
PROGRAM OUTPUTS

Propagation Mode Characterization
- Propagation mode type (i.e., single bottom bounce, direct path, etc.)

Time Variables
- absolute time
- arrival time when ray reaches receiver
- time difference between arrival time and time of arrival of direct ray from CPA point

Range Variables
- horizontal source/receiver range at time ray is emitted by air source \( R'(t) \) in Fig. 2
- horizontal source/receiver range at time ray is received

Arrival Angle Variables
- depression/elevation angle(s) of arriving ray(s) at arrival time of ray(s) for each propagation mode \("D/E" in Fig. 2\)
- azimuthal arrival angle of arriving ray(s) at arrival time of ray(s) for each propagation mode \(\phi \) in Fig. 2

Received Frequency
- ratio of received frequency to source frequency of arrival(s) at arrival time for each propagation mode
Table 2-3 continued.

Transmission Loss Variables

- transmission loss of received pressure level for each separate propagation mode, at appropriate arrival time, referred to air source level at 1 yd from the source, including water volumetric and bottom losses (where appropriate) and air volumetric losses, for both smooth and wind-driven ocean surfaces.
Following the mode identification, a number of time variables are given. The first is the time when the source radiates. Since the source is assumed to be continuously radiating, this time is the same as the absolute time. The arrival time when the ray reaches the receiver is given next. Finally, the time difference between the arrival time and an arbitrarily defined reference time is given. This reference time, which only has physical significance if the distance of closest approach is less than several kiloyards, is the time of arrival of the direct path ray. The program calculates direct path information as though the ocean were isospeed, regardless of the actual sound velocity profile. Physically, this is a good approximation for source/receiver ranges of several kiloyards or less. For greater ranges, a direct path connecting the effective surface source and the receiver will not exist because of refraction. The program, however, will still calculate a direct path arrival time. The user is consequently advised to ignore the time difference output variable when the encounter CPA is more than one or two kiloyards.

Two range variables are calculated. The first is the horizontal source/receiver range at the time the source radiates (same as absolute time). The second is the horizontal source/receiver range when the signal is received (i.e., at the arrival time).

Angle information includes the depression/elevation and azimuthal arrival angle(s) at the appropriate arrival time.

The ratio of received frequency to source frequency (Doppler shift) for each arrival is given.
The transmission loss for each arrival is calculated for both a smooth and wind-driven air/water interface, at a single source frequency. The transmission loss is defined as $10 \log_{10}$ of the ratio of the mean squared received pressure to the square of the air source pressure at one yard from the source.

Broadband noise transmission loss may be treated by subdividing the noise spectrum into a number of frequency bands and using the single frequency results as an estimate of the transmission loss for each sub-band.
SECTION 3
TYPICAL RESULTS

Sample outputs of the programmed algorithm are presented and discussed in this section.

3.1 Representative North Atlantic Environmental Conditions

The input parameters selected for the first example are summarized in Table 3-1.

The sound speed profiles are chosen to be representative of North Atlantic mid-latitude profiles, and represent a compromise between the FNWC profiles of area 50° and the profiles for Marsden square 078, as given in the LRAPP volumes. The bottom type is a low loss type which occurs frequently in the North Atlantic area.

Sea State 3 (average wind speed 8.8 kts) occurs with 44% probability in the summer and 51% probability in the winter in Marsden square 078 and is taken to be representative of the area's general wind conditions. There is no reported predominant wind direction; a direction of 45° with respect to the air source track is assumed arbitrarily and should provide sufficient generality.

Air temperatures are taken to be less than 50°F for the winter profile and greater than 50°F for the summer profile. (The program actually uses air loss curves representative of 0°F for the case when the temperature is less than 50°F; when the air temperature is greater than 50°F, it uses air loss curves representative of 70°F.)

* Fleet Numerical Weather Central (FNWC) and Long Range Acoustic Propagation Project (LRAPP) have surveyed the deep ocean archival data and established representative sound speed profiles and bottom loss curves.
TABLE 3-1
EXEMPLARY INPUT PARAMETERS

SOURCE/RECEIVER

Air Source:
  - Speed = 220 kts
  - Height = 10,000 ft
  - Frequency = 150 Hz

Receiver:
  - Speed = 7 kts
  - Depth = 400 ft

Source/Receiver Tracks:
  - Track directions: parallel
  - Closest point of approach: 4 kyds
  - Range: extends from CPA to beyond first convergence zone (≈ 120 kyd)
  - Time of CPA: \( t = 0 \)

OCEAN

Sound Speed Profiles:

<table>
<thead>
<tr>
<th>Depth (ft)</th>
<th>Sound Speed (ft/sec)</th>
<th>Winter</th>
<th>Depth (ft)</th>
<th>Sound Speed (ft/sec)</th>
<th>Summer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4990.4</td>
<td></td>
<td>0</td>
<td>5052</td>
<td></td>
</tr>
<tr>
<td>420</td>
<td>4998.5</td>
<td></td>
<td>3440</td>
<td>4875</td>
<td></td>
</tr>
<tr>
<td>3720</td>
<td>4886.5</td>
<td></td>
<td>15660</td>
<td>5053</td>
<td></td>
</tr>
<tr>
<td>15540</td>
<td>5050</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bottom Type: LRAPP bottom type 3
Table 3-1 continued.

**ATMOSPHERE**

**Wind:**
- Wind speed: 8.8 kts
- Direction: 45° with respect to source (receiver) track

**Air Temperature:**
- Winter: Less than 50°F
- Summer: Greater than 50°F
3.1.1 Summer

The program output for the summer profile is shown in Figs. 3-1 and 3-2 (CPA = 4 kyds). Rough surface effects are, in general, insignificant for the parameters of this example and will not be discussed. (Section 3.2 presents an example for which wind effects are substantial and are elaborated upon in detail.)

All results are plotted against time on the abscissa. The time shown in the figure is the difference, in secs, between the arrival time of the signal and the arrival time of the direct ray from the source at the CPA point. This latter arrival time is merely an arbitrarily selected time origin. It is generally very close to the time of CPA, if the CPA distance is not exceedingly large. With the source/receiver track configuration assumed, negative time differences occur when the source is approaching the receiver and positive differences occur when the source is moving away from the receiver.

Also shown on the abscissa are two ranges, both in kyds = (1) $R'(t)$, the horizontal source/receiver range at the time the sound energy leaves the air source and (2) $R'(t_A)$, the horizontal source/receiver range at the time the signal arrives at the receiver. Since a signal leaving the source at a given time may travel to the receiver by different propagation modes, with different propagation times, $R'(t_A)$ may differ between the modes.

The upper portion of Fig. 3-1 displays the transmission loss time history, where the transmission loss is referred to the air source level at 1 yd from the source. To obtain absolute intensity levels, the air source level, at 1 yd, must
FIGURE 3-1  TIME HISTORIES OF TRANSMISSION LOSS AND NORMALIZED RECEIVED FREQUENCY, SUMMER PROFILE.
Figure 3-2: Time Histories of Arrival Angles, Summer Profile
be added to the TL values shown. The TL for each propagation mode is shown separately, both for the smooth surface conditions (wind speed = 0 kts) and for the most probable sea state (wind speed = 8.8 kts). Three propagation mode types are identifiable:

- single bottom bounce
- convergence zone
- double bottom bounce.

The direct path, if it occurs, will be important only near the time origin (near CPA time) and, although always calculated by the program, should be ignored for horizontal ranges beyond several kyds since for ranges greater than this, the direct path mode cannot exist physically. When the direct path does not exist, bottom bounce and/or surface duct modes (if a duct is present) must be considered. Note that although the direct path mode does not actually exist, we have still used the arrival time of the direct path from the CPA point as an arbitrary time origin. One could just as well plot all results versus the actual arrival time.

For the summer profile, a surface duct is not present, and bottom bounce modes will be the only available propagation paths from times near CPA time out to the times corresponding to the convergence zone arrival. In most cases, the single bottom bounce mode will dominate until the convergence zone arrival appears. However, for this example, the double bottom bounce arrival is significant even within the convergence zone because the bottom loss is almost negligible for the grazing angles associated with this arrival. While the bottom loss for the single bottom bounce arrival is also small for arrivals originating near the convergence zone, the spreading loss increases rapidly as the range approaches the convergence zone.
range. Thus, in this example, the single bottom bounce arrival becomes weaker than the double bottom bounce arrival as the inner edge of the convergence zone is approached. The single bounce signal eventually disappears completely within the convergence zone.

In this example, the sound speed excess is very small \((c_b \approx c_s)\). Consequently, the convergence zone width is very narrow and the associated transmission loss relatively large, as a result of the fact that only a narrow tube of rays emitted from the virtual surface source can reach the zone. If the sound speed excess were greater, the zone width would be broader and the TL would be less (in magnitude).

Beyond the convergence zone, the single bottom bounce mode does not exist and double bottom bounce arrivals will generally dominate. In the current example, the signal level of the double bounce mode is rather high because the bottom losses are negligible.

It may be observed that there is an asymmetry in the TL time histories about the time origin, which becomes more pronounced for larger times. This effect is a consequence of the ray travel time.

The time history of the Doppler shift (ratio of received frequency to source frequency) is shown below the TL time history. The shift will, in theory, be different for each propagation mode. For large times, however, the shifts for all modes will be essentially the same because of the large distances involved, as the example illustrates. As the CPA point is approached, the shifts usually differ. Generally, the bottom bounce shift will be more gradual than the direct path shift, if a direct path exists. This is a consequence
of the fact that the effective source for the bottom bounce arrivals lies at one ocean depth below the ocean bottom for the single bounce mode (three ocean depths below the bottom for the double bounce arrivals).

The D/E arrival angle time history (Fig. 3-2) displays the angles mode by mode. Positive angles correspond to surface-reflected arrivals, negative to direct arrivals.

Azimuthal arrival angle time histories (bottom of Fig. 3-2) generally display very gradual changes for large times and a rather rapid transition near the CPA time. Because of ray travel time differences, the time histories near the time origin may be mode-dependent.

3.1.2 Winter

Results for the winter profile are displayed in Figs. 3-3 and 3-4.

The winter profile has a surface duct and can therefore permit an additional propagation mode, the surface duct mode, which may be important if the receiver depth is less than the surface duct depth. In the present example, the receiver is in the surface duct, but the diffractive losses due to below-layer leakage are extremely large. Consequently, the surface duct TL is greater than 200 dB (in magnitude) and this mode may be ignored. In general, surface duct propagation will be negligible for low frequencies, as illustrated by the example.

The only essential difference between the winter and summer profile results arises from the fact that the sound speed excess is considerably larger for the winter profile. Consequently, the convergence zone is broader and the TL less (in magnitude).
FIGURE 3-3  TIME HISTORIES OF TRANSMISSION LOSS AND NORMALIZED RECEIVED FREQUENCY, WINTER PROFILE.
FIGURE 3-4  TIME HISTORY OF ARRIVAL ANGLES, WINTER PROFILE.
3.2 Rough Air-Water Interface

Figure 3-5 presents the results of the algorithm for a case in which rough surface effects are significant. (These results were computed by hand at any early stage of the program development. All the equations used are the same as in the programmed algorithm; however, the results may differ in some minor details from the program output because of calculational errors in the hand calculation. A limited set of spot checks indicates that, on the whole, the results shown in Fig. 3-5 and the program output will essentially agree.) In this example, the direct path does exist close to the receiver (within several kyds) since the CPA is 1/3 kyd.

Arrival angle and Doppler time histories are unchanged by rough surface effects. In contrast, the TL levels may be substantially different, although the general shape of the TL history is unchanged except in the direct path region (see, for example, the 1000 ft source height results). A number of effects are operating simultaneously to produce the differences in the TL curves. These are summarized in Table 3-2. (See Ref. 1 for a more detailed discussion of the table parameters.) In the following, each of the TL curves will be considered in turn, and the effects of the parameters in Table 3-2 will be discussed. The virtual source plots given in Ref. 1 should be kept in mind.

Consider first the 50 ft source height curve, with the wind equal 24.5 kts. In the double bottom bounce regions, the emission angles (at the virtual surface source) are moderately small (15°-18°) and the effects of the most probable slope are still important. With the assumed wind and source/receiver track configuration, the receiver sees an arrival which has been emitted parallel to the wind for negative times (source
FIGURE 3-5(b) TIME HISTORIES OF ARRIVAL ANGLE AND RECEIVED FREQUENCY, HIGH SEA STATE
TABLE 3-2

FOR GIVEN SEA STATE, TRANSMISSION GAIN RELATIVE TO A SMOOTH INTERFACE DEPENDS ON:

\[ \theta \] - Depression angle of ray leaving virtual source
\[ \phi_w \] - Angle between plane of acoustic path and direction of waves
\[ f \] - Source frequency
\[ h \] - Source height
\[ \sigma \] - RMS slope (effective)
\[ L \] - Correlation length of surface slopes (effective)
\[ N \] - Average number of specular paths
\[ \psi \] - Most probable slope
approaching receiver) while the arrival for positive times (source moving away from the receiver) was emitted anti-parallel to the wind. The intensity of the outgoing ray at the virtual source is consequently somewhat greater than the flat surface intensity for negative times and somewhat smaller for positive times. The average number of specular paths is about unity for this low source height. Thus the TL lies slightly above the flat surface TL for negative times and slightly below for positive times.

In the single bottom bounce domain, the emission angles are moderately small for large negative or positive times and increase as the source/receiver range decreases (smaller negative/positive times). For large negative/positive times, the same effects operate as in the double bottom bounce case. As the time origin is approached, the emission angles get very steep and the rough surface effects diminish. Thus the TL approaches flat surface values as the source/receiver range diminishes.

Going back to the convergence zone regions, it can be seen from the D/E angle time histories that emission angles from 0° to 13° are involved. All the energy emitted by the source in this angular range appears in the zone. For large negative times, the received arrivals are emitted parallel to the wind and hence a modest gain, over the smooth surface, is seen; for large positive times, the arrivals are emitted anti-parallel to the wind and a modest loss, relative to smooth surface, is the result. As in the bottom bounce case, the average number of specular paths is about unity.

The direct path region displays the effects of the most probable slope most strongly. Over most of the direct path region, only very small emission angles are involved. Furthermore, the angle between the plane of the acoustic path and the wind changes
rapidly. At the outer edge of the region for negative times, the receiver sees rays emitted at very small depression angles in a direction almost parallel to the wind. Consequently, there is a large gain relative to the smooth interface results. At the other edge for positive times, this gain is substantially diminished because the arrivals are emitted in the direction anti-parallel to the wind. As the CPA point is approached, the received rays are emitted more and more in a direction perpendicular to the wind and the effect of the most probable slope becomes increasingly less pronounced. In addition, the emission angle increases. Hence only slight gains are predicted.

When the source height is increased, the effective rms slope decreases somewhat, and in the absence of any other effects, the gains would actually decrease relative to the 50 ft case. However, the average number of specular paths goes up substantially and results in generally large gains, relative to the smooth surface, for all times. This effect is most pronounced for negative times, where the received rays are emitted parallel to the wind, and the favorable disposition of the most probable slope further enhances the gains due to the large $N$. For positive times, where the received rays are emitted anti-parallel to the wind, the effect of the multiple specular paths is diminished by the unfavorable orientation of the most probable slope in regions where the emission angles are small (direct path and convergence zone), and only slightly affected in the bottom bounce regions, where the emission angles are larger.

* The formalism used to calculate the average number of specular paths is probably not correct when the Fresnel zone size is on the order of, or greater than the surface slope correlation length. If this is the case for the parameters of Fig. 3-5(a), then the decrease in transmission loss for the 1000 ft source height may not be correct. The resolution of this problem requires further investigation of the model. A study is currently in progress.
For very large source heights, the effective rms slope would become very small, and the average number of specular paths again approaches unity since the Fresnel zone size increases faster than the correlation length of surface slopes. The surface, in effect, again looks smooth and the TL should approach the smooth surface results.* Physically, the most probable slope should also vanish for large heights (large Fresnel zone size); however, in the algorithm, the most probable slope depends only on wind speed and hence some residual effects for small emission angles will still be predicted. The correction of this artifact must await a suitable theory for the most probable slope.

* Provided that the acoustic Rayleigh parameter for the sea surface is small. Further development of the model to account for scattering (as opposed to specular transmission) is currently underway.
REFERENCES


APPENDIX A
SURFACE DUCT AND CONVERGENCE ZONE SUBROUTINE

Introduction and Summary

The program described below was written to calculate the transmission loss (and some other pertinent variables) between a moving source above the surface of the water and a moving receiver lying beneath the surface. The program calculates (1) the transmission loss for sound rays which travel in a surface duct, if one is present, when the receiver is in the duct, and (2) the transmission loss for sound rays which travel to the first convergence zone when the receiver is within the convergence zone. For both cases, the program also computes arrival angles and the Doppler shift of the arrival.

The program user supplies the parameters which describe the positions and velocities of the source and receiver at the time origin ("initial") as well as the conditions of the air, water and surface between them. The program then determines the propagation modes which apply and calculates the appropriate transmission loss and other transmission variables. A number of outputs are printed for the user. These include, besides the TL, various arrival times, the depression/elevation angle, azimuthal angle, Doppler shift, and horizontal and slant ranges for the pertinent propagation mode. If the surface is rough, both smooth and rough surface results are given. The program is driven by supplying time as the continuing parameter. It is possible to have certain spans of time during which some propagation modes are not possible. In this case the transmission loss must be considered to be infinite and no results are printed. This situation arises when (1) there is no surface duct, or (2) there is a surface duct but the range is too close.
(<5280 feet) for the surface duct algorithm used to be valid, and (3) the source and receiver are too close or too far for convergence zone propagation.

A.1 Glossary

**Input**

**Wind:**
- speed in knots (WS)
- x,y direction cosines; -1 ≤ 0 ≤ 1 (XWC, YWC)

**Air source:**
- x, y components of velocity in kts (VAX, VAY)
- initial x, y position (at t=0) in ft (XIA, YIA)
- source height in feet (H)
- frequency in Hz (F)

**Receiver:**
- velocity along x-axis in kts (VS)
- initial x position (at t=0) in ft (XIS)
- depth in feet (D)
  (y coordinate of receiver always assumed to be zero)

**Sound velocity profile:**
- depths in ft (0, DS, DMIN, DB)
- velocities in ft/sec (CS, CMAX, CMIN, CB)

**Sound velocities (ft/sec):**
- CS speed at surface
- CMAX speed at SVP maximum
- CMIN speed at SVP minimum
- CB speed at ocean bottom

**Depths (ft):**
- DS depth of surface duct
- DMIN depth of SVP minimum
- DB depth to ocean bottom

**Time:**
- starting, ending and step size in seconds (TI, TF, DT)

**Questions:**
- is all input data already in computer? (DATAIN)
  Y(es) or N(o)
- is air temperature greater than 50°F? (QTEMP) Y(es) or N(o)
- is there a surface duct? (LANS) Y(es) or N(o)
- rough or smooth surface? (RSS) R or S

Output:

Propagation mode:
- SD  surface duct, smooth surface
- SDR surface duct, rough surface
- CZS convergence zone with surface duct, smooth surface
- CZSR convergence zone with surface duct, rough surface
- CZ  convergence zone, no surface duct, smooth surface
- CZR convergence zone, no surface duct, rough surface

Times (in secs):
- \( t, t_A, t_A - t_{\text{AMAX}}, t_0 \) (T, TA, TDIFF, TO)
  \[ t = \text{time of emission} \]
  \[ t_A = \text{arrival time of propagated energy} \]
  \[ t_A - t_{\text{AMAX}} = \text{difference in arrival times between arrival at } t_A \text{ and arrival time of direct path from CPA} \]
  \[ t_0 = \text{time of CPA} \]

Ranges:
- \( R', R \) (in ft - RP, R; in kyd - RP3, R3)
- \( R't=t_A \) (in kyd - RPA3)
- \( R_0 \) (in ft - RO)

Doppler shift:
- dimensionless (DOP)

Arrival angles:
- D/E (depression/elevation, in deg - DE)
- \( \phi \) (azimuthal, in deg - PHI)

Transmission loss:
- squared pressure ratio re air source level \( p^2 \)
  at 1 ft, smooth or rough surfaces (w/o atmospheric attenuation) (PRSQ, PRS)

A-3
A.2 Description

The program consists of a main section plus numerous subroutines. The main section calculates some parameters and directs the flow of the calculations. The subroutines are used to input data, evaluate certain functions, and to calculate the transmission losses. Table A-1 gives a general outline of these subroutines. Details of the calculations performed in the main program and the subroutines which calculate transmission loss follow.

The following sections give further details on the calculations performed by the main program and its subroutines.

A.2.1 Main program

The first event in the main program is inputting the data. This may be done within the program or partially done by using a BLOCK DATA subroutine. In either case the following values will be input: \( C_1, C_2, n, \) wind speed, and \( n' \) for the wind, \( v_{ax}, v_{ay}, x_{ia}, y_{ia}, h, v_s, x_{is}, D, C_s, C_b, \) \( \phi, \) \( C_{min}, \) and \( d_{min} \).

The program also demands some non-numerical input. It must be told whether or not the air temperature is greater than 50°F, if a surface duct exists and if the surface of the sea is rough or smooth. If a surface duct exists, values must be supplied for \( C_{max} \) and \( d_S \). Finally, values for the range of time to be covered, \( t_1, t_f, \Delta t \), are input along with the frequency \( f \). The following values are now calculated by the program:
### TABLE A-1

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD</td>
<td>Calculates $t_a$, $t_a - t_{AMAX}$, $\phi$, D/E, doppler, slant range and horizontal range, pressure$^2$ ratio; and transmission loss for sound traveling in the surface duct when the sea surface is smooth.</td>
</tr>
<tr>
<td>SDR</td>
<td>Calculates the corrected values of pressure$^2$ ratio and transmission loss for sound traveling in the surface duct when the sea surface is rough.</td>
</tr>
<tr>
<td>CZS</td>
<td>Calculates $t_a$, $t_a - t_{AMAX}$, $\phi$, doppler, slant range and horizontal range and location of the convergence zone in the presence of a surface duct for a smooth surface. If the receiver lies within the convergence zone, the subroutine also calculates D/E, the pressure$^2$ ratios and the transmission loss.</td>
</tr>
<tr>
<td>CZSR</td>
<td>Calculates the corrected values of pressure$^2$ ratio and transmission ratio when the surface is rough and there is a surface duct and the receiver lies within the convergence zone.</td>
</tr>
<tr>
<td>CZ</td>
<td>Calculates $t_a$, $t_a - t_{AMAX}$, $\phi$, doppler, slant range and horizontal range, and the location of the convergence zone when the sea surface is smooth and there is no surface duct. If the receiver lies within the convergence zone, the subroutine also calculates D/E, the pressure$^2$ ratio and the transmission loss.</td>
</tr>
<tr>
<td>CZR</td>
<td>Calculates the corrected values of pressure$^2$ ratio and transmission loss when the surface is rough, there is no surface duct and the receiver lies in the convergence zone.</td>
</tr>
<tr>
<td>WOUT</td>
<td>Calculates corrected values of transmission loss by including the effects of the air path. This subroutine also converts ranges to kyds and gives the transmission loss re 1 yd.</td>
</tr>
</tbody>
</table>
Table A-1 continued.

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subroutines which evaluate functions</td>
<td></td>
</tr>
<tr>
<td>INT</td>
<td>Evaluates the function [ I(\theta, \Psi, \sigma, \phi) = \frac{x^3}{4} \left[ \frac{\sigma_x}{2} + \frac{3}{2} \left( \frac{A_x}{\sigma_x} \right)^2 \right] \cdot \text{erf} \left( \frac{A_i}{\sigma_x} \right) \cdot \left( 1 + \frac{3}{2} \left( \frac{A_i}{\sigma_x} \right)^2 \right) \cdot \left( 1 + \frac{3}{2} \left( \frac{A_i}{\sigma_x} \right)^2 \right) \cdot \frac{1}{3\sqrt{\pi}} \cdot e^{-\left( \frac{A_i}{\sigma_x} \right)^2} \cdot \left( 1 + \frac{3}{2} \left( \frac{A_i}{\sigma_x} \right)^2 \right) \right]</td>
</tr>
<tr>
<td>ERF</td>
<td>Evaluates the error function, [ \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt ]</td>
</tr>
<tr>
<td>Subroutines which input data</td>
<td></td>
</tr>
<tr>
<td>WIND</td>
<td>Inputs wind speed in knots and the dimensionless x and y direction cosines of the wind.</td>
</tr>
<tr>
<td>AIRV</td>
<td>Inputs the aircraft velocity as vectors in the x and y directions. The velocities are input in knots and converted to ft/sec.</td>
</tr>
<tr>
<td>AIRC</td>
<td>Inputs the initial (i.e., t=0) x and y position of the air source.</td>
</tr>
<tr>
<td>AIRH</td>
<td>Inputs the height of the aircraft in ft.</td>
</tr>
<tr>
<td>AIRF</td>
<td>Inputs the frequency of aircraft radiation in Hz.</td>
</tr>
<tr>
<td>SUBV</td>
<td>Inputs the velocity of the receiver in the x direction in knots and converts to ft/sec. (The y-direction velocity is assumed to be zero.)</td>
</tr>
</tbody>
</table>
Table A-1 continued.

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SUBC</strong></td>
<td>Inputs the x position of the receiver at t=0 in ft. (The y-position is assumed to be zero.)</td>
</tr>
<tr>
<td><strong>SUBD</strong></td>
<td>Inputs the depth of the receiver in ft.</td>
</tr>
<tr>
<td><strong>ENVS</strong></td>
<td>Inputs the sound velocity at the surface of the sea in ft/sec.</td>
</tr>
<tr>
<td><strong>ENVC</strong></td>
<td>Inputs the sound velocity at the bottom in ft/sec. Also inputs the depth of the bottom in ft.</td>
</tr>
<tr>
<td><strong>ENVD</strong></td>
<td>Inputs the minimum sound velocity (deep sound channel) in ft/sec along with the depth at which it occurs in ft.</td>
</tr>
<tr>
<td><strong>ENVA</strong></td>
<td>Inputs the sound velocity at the bottom of the surface duct in ft/sec (maximum sound velocity). The depth of the surface duct is also input in ft.</td>
</tr>
<tr>
<td><strong>TIM</strong></td>
<td>Inputs the starting and ending times for the calculations as well as the step size in time. All these values are in seconds.</td>
</tr>
<tr>
<td><strong>WATIN</strong></td>
<td>Subroutine used to change values of the input and then rerun the program.</td>
</tr>
</tbody>
</table>
\[ e_0 = \frac{(x_{ln} - x_{ln})(v_{a} - v_{ax}) - y_{ln} v_{ay}}{(v_{a} - v_{ax})^2 + v_{ay}^2} \] \hspace{1cm} (1)

\[ R'_{(x \neq x_0)} = \sqrt{[(v_{a} x_{0} + x_{ln}) - (v_{ax} x_{0} + x_{ln})]^2 + [v_{ay} x_{0} + y_{ln}]^2} \] \hspace{1cm} (2)

\[ R_{(x \neq x_0)} = \sqrt{[R'_{(x \neq x_0)}]^2 + D^2} \] \hspace{1cm} (3)

If the surface is rough, the following values also are calculated:

\[ f = \left( \frac{g_p (c/f) q^2}{g^2 U^*_c} \right)^{1/4} \] \hspace{1cm} (4)

where \( g_p \) is the acceleration due to gravity in ft/sec\(^2\) and \( U^*_c \) is the wind speed in ft/sec.

\[ u_w = \frac{1}{\sqrt{2}} \left\{ 1.15 \times 10^4 \sqrt{\frac{q^2}{g_c^2}} \frac{u_c}{g_c} \left[ 1 - \text{erf}(f) \right] \right\}^{1/2} \] \hspace{1cm} (5)

where \( u_c \) and \( g_c \) are \( U_p \) and \( g_p \) converted to cgs units. The program also calculates

\[ \tilde{\gamma} = 2.3 \times 10^6 \frac{q}{g_c u_c} \left( \frac{q}{g_c} \right)^{1/2} \left\{ 1 - \text{erf}(f) \right\} + \frac{q^2}{5} \left\{ \frac{1}{2 f^2} - 1 \right\} \] \hspace{1cm} (6)
where \( h_c \) is the source height in cgs units.

\[
\frac{m}{m'} = \frac{m}{m'} = \left[ e^{-\frac{m}{m'}} + \frac{m}{m'} \left\{ 1 + \text{erf} \left( \frac{m}{m'} \right) \right\}^2 \right]^{\frac{1}{2}}
\]  
(8)

\[
\bar{\Psi} = 2.86 \times 10^{-3} = \frac{\bar{m}}{100}
\]  
(9)

where the error functions, erf, are evaluated by subroutine ERF.

All these parameters are not functions of time. The program now is ready to march through time. The parameter \( t \) is set equal to the starting value of time \( t_1 \) (not necessarily equal to zero; may be plus or minus) and the following parameters are calculated:

\[
x_2 = v_2 t + x_{i2}
\]  
(10)

\[
x_3 = v_3 t + x_{i3}
\]  
(11)

\[
y_2 = v_{2y} t + y_{i2}
\]  
(12)

\[
x^2 = \sqrt{(x_2 - x_{e2})^2 + y_{i2}^2}
\]  
(13)
\[ R = \sqrt{(R')^2 + D^2} \]  
\[ A = \left[ \frac{\epsilon_1 f_0^2}{f + f_0^2} + \frac{40 f_0^4}{4100 + f_0^2} \right] \frac{1}{3000} \]

where \( f \) is the frequency in kHz.

\[ \theta_\alpha = \cos^{-1} \left( \frac{c_1}{c_b} \right) \]  
\[ g_b = \frac{c_b - c_{min}}{d_b - d_{\text{min}}} \]

If there is a surface duct, the program will check to see if the receiver is within it. If the receiver is not in the surface duct the program skips to the convergence zone calculation. If the receiver does lie in the surface duct, the SD subroutine will be called, and the calculated values will be typed out. The subroutine WOUT will be called next to convert the values just calculated to other units and to include the effects of air losses. If the sea surface is rough, the subroutine SDR will now be called to calculate the effects of this roughness on the transmission loss. WOUT will be called again to operate on this new transmission loss.

The program next will go to the convergence zone calculation. If there is a surface duct, subroutine CZS will be called. In the absence of a surface duct, CZ will be called. In either case the subroutine returns a tag which indicates whether or not the receiver lies within the convergence zone.
If the tag says no, the program skips further calculations and chooses a new value of t. If the receiver is in the convergence zone the calculated parameters are typed out and WOUT is then called. If the sea surface is rough the subroutine CZSR or CZR will be called, depending on the presence or absence of a surface duct. The values calculated will be typed out and WOUT will again be called.

Next a new value of t is calculated by adding \( \Delta t \) to the present value. This value is checked against the final value, \( t_f \), and if it is smaller or equal to it, the program will start calculating again at the point where dependence on t begins. If the new value of t is beyond the final value allowed for the time parameter, subroutine WATIN is called. This subroutine allows changes to be made in many of the input parameters at which point the program may be continued. WATIN also allows the program to end, if that is the desired option.

### A.2.2 SD subroutine

The SD subroutine calculates the following:

\[
\begin{align*}
    t_a &= t + \left( \frac{4}{c_s} \right) + \left( \frac{R'}{c_s} \right) \\
    t_{a - t_{\text{max}}} &= t - t_o + \left( \frac{R' - R(t_o t_s)}{c_s} \right)
\end{align*}
\]

\[
\phi = \cos^{-1} \left( \frac{x_{a} - x_{d}}{R'} \right) \\
\text{doppler} = \sqrt{\frac{1 - \frac{2 \nu_s}{c_s} \left( \frac{x_{a} - x_{d}}{R} \right)}{1 - \frac{2 \nu_s}{c_s} \cos(\psi)}}
\]
where \( \cos(\psi) = \frac{v_{ax}(x_e - x_a) - v_{ay} y_a}{v_{ax} R} \)  

\[ D/E = \sqrt{\frac{2(C_{max} - C_e)}{C_{max}}} \]  

\[ A_{tot} = A' + [\frac{7.44 \times 10^5}{f_a} q^2 \frac{1}{d_a^3} + SSM \sqrt{\frac{f_a}{d_a^2}}] \]  

where \( SSM = \begin{cases} 9.0 & \text{if } \frac{U_e}{(\text{wind speed in kts})} > 10 \\ 4.5 & \text{if } \frac{U_e}{(\text{wind speed in kts})} < 7 \end{cases} \)  

\[ q_s = \frac{C_{max} - C_{min}}{d_{min} - d_e} \]  

\[ \frac{P_s}{P_e} = \frac{3n^2}{8R^2d_s} \left( \frac{2[C_{max} - C_o]}{C_{max}} \right)^{3/2} - \left( \frac{A_{max} C_e}{10} \right) \]  

\[ TL = 10 \log_2 \left( \frac{P_s}{P_e} \right) \]  

A.2.3 SDR subroutine

This subroutine uses the values calculated by SD as well as the original input to calculate the following:

\[ \cos(\phi_o) = \frac{n_x(x_e - x_a) - n_y y_a}{R'} \]
\[ \bar{N} = \text{(coefficient of } \bar{N}) \left\{ \text{erf} \left( \frac{\tan \left( \frac{\pi x}{a} \right)}{\sqrt{2} / \sigma_w} \right) + \text{erf} \left( \frac{\left(1 - \sin \left( \frac{\pi x}{a} \right) \right)}{\sqrt{2} / \sigma_w} \right) \right\} \]

\[ \frac{\Phi_{\text{rec}}}{\mu^2} = \frac{8 \pi^2}{R^2 \rho^2} I \left( \phi/e, \bar{\psi}, \sigma_w, \phi_w \right) \bar{N} = 10^{-\left(\frac{\Delta_{\text{max}} R^2}{10}\right)} \]

\[ TL = 10 \log_{10} \left( \frac{\Phi_{\text{rec}}}{\mu^2} \right) \]

where \( I(D/E, \bar{\psi}, \sigma_w, \phi_w) \) is evaluated by using subroutine INT and the error functions are evaluated in ERF.

### A.2.4 CZS subroutine

The first part of this subroutine performs the same calculations listed under the SD subroutine, equations (18) through (23) and (26). In addition, the following are calculated:

\[ q_i = \frac{C_{\text{max}} - c_i}{d_i} \]

\[ R_{\text{co}} = 2C_{\text{max}} \left[ \left( \frac{1}{q_i} \sqrt{1 - \left( \frac{c_i}{C_{\text{max}}} \right)^2} \right) \times \left( \frac{1}{q_i} + \frac{1}{q_o} \right) \right] \]

\[ R_o = 2C_b \left[ \left( \frac{1}{q_i} \sqrt{1 - \left( \frac{c_i}{C_b} \right)^2} \right) \times \left( \frac{1}{q_i} + \frac{1}{q_o} \right) \right] \]

\[ - \left( \frac{1}{q_i} + \frac{1}{q_o} \right) \sqrt{1 - \left( \frac{C_{\text{max}}}{C_b} \right)^2} \]

A-13
Now the program can decide if the receiver is in the convergence zone or not. If the following criterion is not met, the tag 'EXIST' will be no, 'N', and the subroutine will return to the main program. If

\[ R_{sd} \neq R' \neq R_b \quad \text{or} \quad R_b \neq R' \neq R_{sd} \]  

then 'EXIST' will be 'Y' (yes) and the following calculations are performed:

\[ \theta_{sd} = \cot^{-1} \left( \frac{t_i}{c_{man}} \right) \]  

\[ t_1 = t_i + \frac{t}{|t|} \sqrt{t_o^2 + \left[ \frac{(x_{i_a} - x_{i_b})^2 + y_{i_b}^2 + R_{i_b}^2}{(v_{i_a} - v_{i_b})^2 + y_{i_b}^2} \right]} \]  

\[ t_2 = t_i + \frac{t}{|t|} \sqrt{t_o^2 + \left[ \frac{(x_{i_a} - x_{i_b})^2 + y_{i_b}^2 + R_{i_b}^2}{(v_{i_a} - v_{i_b})^2 + y_{i_b}^2} \right]} \]  

\[ \frac{D/E}{E} = \frac{\theta_b + (t - t_i)}{(t_2 - t_i)} \left( \theta_{sd} - \theta_b \right) \]  

\[ R_{sd}^2 = \frac{32 \pi^2}{3} \frac{\left( \theta_{sd}^2 - \theta_b^2 \right)}{\left( \theta_b + \theta_{sd} \right) |R_b^2 - R_{sd}^2|} \cdot 10^{-\left( \frac{10^{10}}{10} \right)} \]  

\[ TL = 10 \log \left( \frac{R_{sd}^2}{R_i^2} \right) \]
A.2.5 CZSR subroutine

This subroutine uses values calculated in subroutine CZS as well as data from the main program. The following values are calculated:

\[
\cos(\phi_w) = \frac{n_x(x_2 - x_1) - n_y y_2}{R'}
\]

\[
N = (\text{coefficient of } R) \left\{ \text{erf} \left( \frac{\tan(\frac{\theta_w + \theta_b}{2})}{\sqrt{2} \sigma_w} \right) + \text{erf} \left( \frac{1 - \sin(\frac{\theta_w + \theta_b}{2})}{\cos(\frac{\theta_w + \theta_b}{2})} \right) \right\}
\]

\[
\frac{A_{15}}{P_{15}} = \frac{32n^2\{I(\theta_b, \overline{\theta}, \sigma_w, \phi_w) - I(\theta_b, \overline{\theta}, \sigma_w, \phi_w)\}}{[\sigma_{\theta - \theta_b}] \left| R^2_{15} - R_{30}^2 \right|} \frac{N_{10}}{10} - (45')
\]

\[
TL = 10 \log \left( \frac{A_{15}}{P_{15}} \right)
\]

A.2.6 CZ subroutine

The first part of this subroutine performs the same calculations listed under the SD subroutine, equations (18) through (23). Then the following are calculated:

\[
\frac{C_f - C_{\text{min}}}{C_{\text{min}}}
\]
\[ E_o = 2c_o \left[ \left( \frac{1}{q_e} + \frac{1}{q_i} \right) \sqrt{1 - \left( \frac{c_{\text{min}}}{c_o} \right)^2} - \frac{1}{q_e} \sqrt{1 - \left( \frac{c_i}{c_o} \right)^2} \right] \]  

\[ R_{(\theta, \theta o)} = 2c_o \left( \frac{1}{q_e} + \frac{1}{q_i} \right) \sqrt{1 - \left( \frac{c_{\text{min}}}{c_o} \right)^2} \]  

Now the subroutine can decide whether or not the receiver is in the convergence zone. If it isn't, the tag, 'EXIST' is set to 'N', no, and control returns to the main program. If one of the following criteria are met:

\[ R_e \neq R \neq R_{(\theta, \theta o)} \quad \text{or} \quad R_{(\theta, \theta o)} \neq R' \neq R_e \]  

then 'EXIST' is set to 'Y', yes, and the following are calculated:

\[ t_1 = t_0 + \frac{t}{|t|} \sqrt{t_1^2 - \left[ \frac{(x_{e_1} - x_{i_1})^2 + y_{e_1}^2 - R_{e_1}^2}{(v_{e_1} - v_{i_1})^2 + v_{e_1}^2} \right]} \]

\[ t_2 = t_0 + \frac{t}{|t|} \sqrt{t_2^2 - \left[ \frac{(x_{e_2} - x_{i_2})^2 + y_{e_2}^2 - R_{e_2}^2}{(v_{e_2} - v_{i_2})^2 + v_{e_2}^2} \right]} \]

\[ \theta_0 = \Theta_0 \left( 1 - \frac{(t-t_1)}{(t_2-t_1)} \right) \]

\[ \frac{P_{e_1}}{P_e} = \frac{32 \pi^2 \Theta_0^2}{3} \left[ \frac{R_{e_1}^2 - R_{(\theta, \theta o)}^2}{R_{e_1}^2} \right] 10^{-\left( \frac{A_0'}{10} \right)} \]

\[ TL = 10 \log_{10} \left( \frac{P_{e_1}}{P_e} \right) \]
A.2.7 CZR subroutine

The values calculated in CZ are used in this subroutine along with values from the main program. The following are calculated:

\[
\cos(\phi_w) = \frac{n_x (x_e - x_c) - n_y y_c}{R^1}
\] (56)

\[
\bar{N} = \text{(coefficient of } \bar{N}) \left\{ \text{erf} \left( \frac{\tan(\alpha_w)}{\sqrt{2} \sigma_w} \right) + \text{erf} \left( \frac{-\sin(\phi_w)}{\sqrt{2} \sigma_w} \right) \right\}
\] (57)

\[
\frac{\phi_w}{\phi_{10}} = \frac{32 \pi n^2 I(\theta_e, \psi, \sigma_w, \phi_w) \bar{N} \times 10^{-\frac{(AB')}{10}}}{\theta_e |R_0^1 - R_{\theta_e=0}|}
\] (58)

\[
\tau_\lambda = 10 \log_{10} \left( \frac{\phi_{10}}{\bar{N}} \right)
\] (59)

A.2.8 WOUT subroutine

The following values are converted from ft to kyd:

\[
R_3' = \frac{R}{3000}
\] (60)

\[
R_3 = \frac{R}{3000}
\] (61)
The value of $R'$ is calculated at $t_a$ and then expressed in kyds:

$$R'_a(t_a) = \left\{ \frac{1}{\sqrt{(v_x t_a + x_i a) - (v_x t_a + x_i a)^2 + [\text{deg} t_a + y_i a]^2}} \right\} \text{sec}$$

(62)

The incoming TL is now corrected from dB re 1 ft to dB re 1 yd by the factor

$$\text{TL}_y = 10 \log_2 (9)$$

(63)

and by a factor depending on temperature:

$$\text{corr} = \begin{cases} 1.25 \times 10^6 \times f \times h & \text{if temperature} < 50{}^\circ F \\ 7.48 \times 10^6 \times f \times h \div 164.05 & \text{if temperature} > 50{}^\circ F \end{cases}$$

(64)

The resulting transmission loss is:

$$\text{TL}_a = \text{TL} - \text{TL}_y - \text{corr}$$

(65)
A.3 Directions for Running the Program

The main program and all the subroutines are collected together under the name WATER. It may be run in either of two ways:

1. The data may be entered by the user when running the program.

2. The data may be set up in a BLOCK DATA file. When the user is ready to run the program, the BLOCK DATA file is loaded at the same time as WATER. This option is useful if many parameters remain constant for different values of frequency and time.

An illustration of the first method is shown in Example 1. The user tells the computer to EXecute WATER F4. All user typing in the examples has been underlined. The computer now will compile the program, if the compiled version was not stored. Then the compiled version is loaded into the working area in the computer and the program is started.

The first question is whether the data has been included; since it hasn't, the user would type N or NO. Next the user must indicate if the air temperature is greater than 50°F. If the answer is to be "yes", the user would type Y or YES. Now the program will ask for several parameters:

- wind speed (kts) and direction cosines (dimensionless)
- aircraft velocity vectors (kts)
- aircraft \((x,y)\) position and height above water (ft) at \(t=0\)
- submarine velocity (kts)
- submarine position and depth (ft) at \(t=0\)

The program now asks if there is a surface duct. The user says no by typing N or NO. In answer to the next question, the
EXAMPLE 1

*EX WATER EA*  
LOADING

LOADER 5K CORE  
6*3K MAX SPA WORDS FREE  
EXECUTION

IS DATA IN COMPUTER? Y

TEMPERATURE GREATER THAN 50 DEG F? Y

ENTER THE APPROPRIATE PARAMETERS IN THE DIMENSIONS INDICATED.

WIND SPEED = 8.8  
X DIRECTION COSINE OF WIND = .707  
Y DIRECTION COSINE OF WIND = .707

VELOCITY VECTOR X-DIRECTION (KTS) = 223
VELOCITY VECTOR Y-DIRECTION (KTS) = 0

INITIAL X-COORDINATE OF AIRCRAFT (FT) = 0.  
INITIAL Y-COORDINATE OF AIRCRAFT (FT) = 12000.

HEIGHT OF AIRCRAFT FROM SFA SURFACE (FT) = 10000.

VELOCITY VECTOR X-DIRECTION SURF (KTS) = 7
INITIAL X-COORDINATE OF SURF (FT) = 0.

DEPTH OF SURF (FT) = 400.

IS THERE A SURFACE DUCT? N
ROUGH OR SMOOTH SURFACE (K OR S)? S

SURFACE SOUND SPEED (FT/SEC) = 5850

BOTTOM SOUND SPEED (FT/SEC) = 5850

DEPTH FOR BOTTOM SOUND SPEED (FT) = 15400.

MIN. SOUND SPEED (FT/SEC) = 4075

DEPTH AT MIN. SOUND SPEED (FT) = 3441.

INITIAL TIME = -1000

FINAL TIME = 1000

TIME INCREMENTS = .25

FREQ. OF AIRCRAFT RADIATION (HZ) = 150.

* Here, "initial" refers to t=0
† In this statement, "initial" refers to the starting time of the encounter.  
A-20
SMOOTH SURFACE; NO SURFACE DUCT

\[ T_0 = 0.0, R'(T_0) = 12000, R(T_0) = 12007, F = 150.000 \]
\[ H = 10000.0 \text{ FT}, \text{ WS} = 3.8 \text{ KTS}, \text{ NX} = 0.71, \text{ NY} = 0.71 \]

**MODE** T TA TDF K' K DOP D/E PHI P RATIO TL

To change a run parameter, enter the appropriate number:
1. Aircraft Vel. Vectors
2. Aircraft Initial Position
3. Aircraft Height
4. Aircraft Radiated Energy
5. Sub VEL. Vector
6. Sub Initial Position
7. Sub Depth
8. Surface Sound Speed
9. Bottom Sound Speed and Depth
10. Minimum Sound Speed and Depth
11. Max. Sound Speed and Depth
12. Time of Events
13. Run
14. Stop
15. Wind Parameters

**ENTER THE APPROPRIATE NUMBER = 12**

**INITIAL TIME = -1000**
**FINAL TIME = 10000**
**TIME INCREMENTS = .**

CHANGE PARAMETER = 13

... Continue with the rest of the table...

--

*Here, "initial" refers to the encounter starting time.*
user types S to indicate a smooth sea surface (i.e., windspeed = 0) or R to indicate a rough surface (windspeed ≠ 0). Now the program will ask for the appropriate sound speed profile quantities:

- sound speed (ft/sec)
- at depth (ft)

Finally the program asks for the starting and final values of the time parameter, the step size in time, and the aircraft frequency of radiation. This concludes the data input and the calculations now proceed. In the example illustrated, no results are outputted because the convergence zone is very narrow and the time step selected was too coarse.

After going through all values of time input, the program comes back to the user to ask if any changes in parameter are desired. If the user wants to change the times used, 12 must be typed, and then new values for the times used are entered. If no other changes are desired, the program can be rerun with the new values by typing 13. Some typical output is shown. Assuming that the user gets satisfactory results and no more changes are desired, the program may be stopped by typing 14. (An explanation of the output format is given in Section A.4.)

There is much less input at the time of running the program when BLOCK DATA is used. The BLOCK DATA file must be created in the format of Fig. A-1. Data are entered in the file using the same conventions and units as when the program is run without the file, with the following exceptions and additions:

1. source and receiver velocities are entered in ft/sec.

2. Cl is the sound speed in air (ft/sec)
Figure A-1. BLOCK DATA FORMAT

C

BLOCK DATA

C

COMMON /VAX, VAY, X1A, Y1A, H, F, VS, X1S, D1AXN, W1S, YWC, XWC
COMMON /OUT/ T, K, KP, T, TD1F, T1, DOP, DF, PHI, PKSU, A, MODE
COMMON /CC/ C1, C2, CS, CR, DR, DS, CMIV, CMAX, DMIN, ETA, GS, GR, KO, IO, PI,
           IGI
COMMON/MM/MM

C

DATA VAX/371.36/
DATA VAY/0.0/
DATA X1A/0.0/
DATA Y1A/12000.0/
DATA H/10000.0/
DATA F/150.0/
DATA VS/11.9/16/
DATA X1S/0.0/
DATA D/400.0/
DATA AXN/0.22/
DATA WS/5.8/
DATA XWC/.707/
DATA YWC/.707/

C

DATA C1/1100.0/
DATA C2/5000.0/
DATA CS/4990.4/
DATA CR/5050.0/
DATA DR/15540.0/
DATA DS/420.0/
DATA CMIV/4486.5/
DATA CMAX/499.5/
DATA DMIN/3720.0/
DATA ETA/3.0E-27/
DATA PI/3.14159265/

C

DATA M/1/

C

FND
3. C2 is the water sound speed in the vicinity of the surface (ft/sec). Any number close to 5000 ft/sec will be adequate.

4. ETA is a test number, against which the computed value of the ratio of the squared received pressure to squared source pressure at unit distance is compared before the TL is computed. If the ratio is less than ETA, the TL is set equal to -999.

5. M is an internally used tag which tells the computer to delete printing out the full list of data questions when M = 1.

In Example 2 the BLOCK DATA file is stored as B2, therefore execution of the program is begun by typing EX WATER,F4,B2,F4. In reply to the question on whether data is in the computer or not, the user would type Y or YES. The next question concerns the air temperature. If the temperature is less than 50°F the user would type N or NO.

The program now skips all the beginning input and asks if there is a surface duct. If there is, the user would type Y or YES. The user must be sure that this answer agrees with the sound speed profile included in the BLOCK DATA file. The type of sea surface is determined and then the program skips over the entry of sound speed profile data directly to the values of time and frequency to be used. In the example, an error was made in typing in the time increment. It was corrected by typing (control A) for each character to be deleted and then supplying the correct values. As soon as this data is entered, the calculations begin.
### EXAMPLE 2

LOADFRE 5K CORF
6+3K MAX 524 WORDS FREE
EXECUTION

IS DATA IN COMPUTER? Y

TEMPERATURE GREATER THAN 50 DEG F? N

IS THERE A SURFACE DUCT? Y
ROUGH OR SMOOTH SURFACE (R OR S)? R

INITIAL TIME = 440.
FINAL TIME = 440.
TIME INCREMENTS = 40.

FREQ. OF AIRCRAFT RADIATION (HZ) = 150.

**ROUGH SURFACE: SURFACE DUCT**

<table>
<thead>
<tr>
<th>T</th>
<th>TA</th>
<th>T1</th>
<th>TIF</th>
<th>R'</th>
<th>R</th>
<th>DOP</th>
<th>D/F</th>
<th>PHI</th>
<th>PHA</th>
<th>TL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD</td>
<td>440.</td>
<td>-603.</td>
<td>-615.237402</td>
<td>237403</td>
<td>1.081</td>
<td>3.26177</td>
<td>0.000E+00</td>
<td>999.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>440.</td>
<td>79.201</td>
<td>79.201</td>
<td>72.425</td>
<td>-999.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDR</td>
<td>440.</td>
<td>1.000E+00</td>
<td>0.000E+00</td>
<td>999.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDR</td>
<td>440.</td>
<td>-999.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZS</td>
<td>440.</td>
<td>-403.</td>
<td>-415.237602</td>
<td>237603</td>
<td>0.000E+00</td>
<td>4.8177</td>
<td>8.263E-12</td>
<td>120.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZS</td>
<td>440.</td>
<td>79.201</td>
<td>79.201</td>
<td>72.425</td>
<td>-113.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZSR</td>
<td>440.</td>
<td>1.000E+00</td>
<td>1.043E-12</td>
<td>119.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZSR</td>
<td>440.</td>
<td>-113.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>430.</td>
<td>-576.</td>
<td>-587.226830</td>
<td>226831</td>
<td>1.081</td>
<td>3.26177</td>
<td>0.000E+00</td>
<td>999.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>430.</td>
<td>75.610</td>
<td>75.610</td>
<td>69.094</td>
<td>-999.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDR</td>
<td>430.</td>
<td>1.000E+00</td>
<td>0.000E+00</td>
<td>999.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDR</td>
<td>430.</td>
<td>-999.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZS</td>
<td>430.</td>
<td>-576.</td>
<td>-587.226830</td>
<td>226831</td>
<td>1.081</td>
<td>7.70177</td>
<td>8.260E-13</td>
<td>120.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZS</td>
<td>430.</td>
<td>75.610</td>
<td>75.610</td>
<td>69.094</td>
<td>-113.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZSR</td>
<td>430.</td>
<td>1.000E+00</td>
<td>1.043E-12</td>
<td>119.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZSR</td>
<td>430.</td>
<td>-112.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>420.</td>
<td>-548.</td>
<td>-595.216040</td>
<td>216040</td>
<td>1.081</td>
<td>3.26177</td>
<td>0.000E+00</td>
<td>999.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SD</td>
<td>420.</td>
<td>72.020</td>
<td>72.020</td>
<td>65.769</td>
<td>-999.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDR</td>
<td>420.</td>
<td>1.000E+00</td>
<td>0.000E+00</td>
<td>999.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDR</td>
<td>420.</td>
<td>-999.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**CHANGE PARAMETER = 4**

**FREQ. OF AIRCRAFT RADIATION (Hz) = 5000.**

**CHANGE PARAMETER = **

**ROUGH SURFACE; SURFACE DUCT**

- $T_0 = 0.0$, $R'(T_0) = 12000$, $R(T_0) = 12007$, $F = 5000.000$
- $H = 10000$ FT, $WS = 8.8$ KTS, $NX = 0.71$, $NY = 0.71$

<table>
<thead>
<tr>
<th>MODE</th>
<th>$T$</th>
<th>$TA$</th>
<th>$TDIF$</th>
<th>$R'$</th>
<th>$R$</th>
<th>$DOP$</th>
<th>$D/F$</th>
<th>$PHI$</th>
<th>$P$</th>
<th>$RATIO$</th>
<th>$TL$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD</td>
<td>400</td>
<td>-603</td>
<td>-615</td>
<td>237603</td>
<td>1.081</td>
<td>3.26177</td>
<td>6.400F-29-281.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDR</td>
<td>400</td>
<td>-601</td>
<td>79.201</td>
<td>79.201</td>
<td>72.425</td>
<td>1.085F+00</td>
<td>1.398F-28-278.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDR</td>
<td>400</td>
<td>-601</td>
<td>79.201</td>
<td>79.201</td>
<td>72.425</td>
<td>1.085F+00</td>
<td>1.398F-28-278.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZS</td>
<td>400</td>
<td>-603</td>
<td>-615</td>
<td>237603</td>
<td>1.081</td>
<td>4.61777</td>
<td>1.796F-15-147.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZS</td>
<td>400</td>
<td>-603</td>
<td>79.201</td>
<td>79.201</td>
<td>72.425</td>
<td>1.085F+00</td>
<td>2.514F-15-146.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZSR</td>
<td>400</td>
<td>600</td>
<td>600</td>
<td>600</td>
<td>1.085E+00</td>
<td>2.514F-15-146.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**CHANGE PARAMETER = **

*A-26*
In the example the transmission loss in the surface duct is given as -999. This is not the actual value, but is a convention used to show that the loss is so small it is not possible for the computer to work with such small numbers.

At completion of the time values given, the program asks if there are any parameter changes desired. If a different frequency is needed, the user would type 4 and then give the new value. If no more changes are desired, the user would tell the program to rerun by typing 13. After completion of this new output, the user would indicate the end of the run by typing 14 in answer to the change parameters question.
A.4 Output Description - 1

A description of the output produced by the program is given in detail below. An example is given on which each item has been tagged. These tags are used in the following explanation to indicate the item under discussion.

EXAMPLE

Output Description - 2

As soon as all input has been entered, the program prints out a heading which indicates whether the sea surface is smooth or rough (1), and if there is a surface duct or not (2). Next several parameters which describe the data as well as some parameters which do not depend on time are printed. \( t_0 \) is given in seconds (3), \( R' \) and \( R \), calculated for \( t=t_0 \) are printed in ft, (4) and (5). Frequency is given in Hertz (6), aircraft height in ft (7). Finally the wind parameters are printed: wind speed in kts (8) and the x- and y-direction cosines, (9) and (10), which are dimensionless.
The program now types out a heading for the time dependent calculations. Next a line of data starts with a tag indicating the type of propagation (11). The names are SD for surface duct, CZS for convergence zone in the presence of a surface duct, and CZ for convergence zone when there is no surface duct. The addition of the suffix R means the rough sea surface case; the absence of a suffix refers to the smooth sea surface case. The next item in the line is the time used in seconds (12).

Data listed beyond these two items varies with the line. (In every case a minimum two lines of data are output. If there is a rough sea surface an additional two lines are output.)

In the first line the third item is $t_A$ in seconds (13), followed by $t_{A-t}AMAX$ in seconds (14) $R'$ in ft (15), and $R$ in ft (16). Next come the doppler value (17) which is dimensionless, $D/E$ in degrees (18), and azimuthal angle, $\phi$, in degrees (19). Finally, the pressure ratio is printed (re 1 ft and without air losses) (20) and then the transmission loss (21) in dB re 1 ft (without air losses).

The third and fourth items in the second line are $R'$ and $R$ in kyds, (22) and (23). Next the range, calculated for $t = t_A$ is printed in kyds (24) and finally the transmission loss is printed (25). This transmission loss is in dB re 1 yd and is corrected to include air losses.

The third line only appears if the sea surface is rough. The third item in the line is $N$ (26). Next the pressure ratio (re 1 ft and without air losses) is printed (27) and finally the transmission loss in dB re 1 ft without air losses (28).

The fourth line also appears only for rough sea surface conditions. The only term after the tag and time is the transmission loss in dB re 1 yd, corrected for air losses (29).
A.5 Program Listing
TRANSMISSION LOSS CALCULATIONS (100541)

COMMON  /IN/ VAX, VAY, XIA, YIA, H, F, VS, XIS, D, AXN, KS, XMC, YMC, TI, TE, DT
COMMON /INIT/ TR, KP, TA, TDFL, TL, DOP, DF, PHI, PRSU, A, MODE
COMMON /C/ CI, CP, CS, CH, PR, HS, CMIN, CM, DMIN, ETA, CS, AR, R0, 10, PI, I 61
COMMON /MM/
COMMON /RS/ SH, CPW, PSIR, FPN, CI, PRS, TLRS, RN

CI=1100.0
CR=5000.0
AXN=0.0
ETA=1.032
PI=3.14159265
SPI=SQRTP(PI)
ARSO=32.174
SNC=1.1544*SPI/(SORT(P)*((981.0**3))
FRSC=2.364**(4.0/SORT(P)/(3.0+981.0))

TYPE 987
987 FORMAT( ' IS DATA IN COMPUTER? ', E)
ACCEPT 3, DAIN
TYPE 952
952 FORMAT( ' TEMPERATURE GREATER THAN 50 DEG F? ', E)
ACCEPT 3, OTEMP
IF ((DAIN.EQ. 'Y')) ON, (DAIN.EQ. 'YES')) GO TO 985
M=0

TYPE 1
1 FORMAT( ' ENTER THE APPROPRIATE PARAMETERS IN ' ,!
       ' THE DIMENSIONS INDICATED', ')!
CALL WINOWS(XMC, YMC)
CALL AIRU(VAX, VAY)
CALL AIRG(XIA, YIA)
CALL AIRH(H)
CALL SIR(VS)
CALL SIRG(XIS)
CALL SIRH(R0)

985 TYPE 9
99 FORMAT( ' IS THERE A SURFACE NCT?', PX, E)
ACCEPT 3.LANS
FORMAT(A3)
RSSE='S'

1
FORMAT('ROUGH OR SMOOTH SURFACE (R OR S)? 'S)
ACCEPT 3.RSS
IF((LANS.FO. 'YES') OR ((LANS.FO. 'Y')) GO TO 4
IF((DATAIN.FO. 'Y')) OR ((DATAIN.FO. 'YES')) GO TO 98A
CALL FNUS(CS)
CALL FNVC(CH,DA)
CALL FNVD(CMIN,DMIN)
GO TO 5

4
IF((DATAIN.FO. 'Y')) OR ((DATAIN.FO. 'YES')) GO TO 98A
CALL FNUS(CS)
CALL FNVC(CH,DA)
CALL FNVD(CMIN,DMIN)
CALL ENVA(FMAX,DS)
CONTINUE
98A
CALL TIN(TI,TF,DT)
CALL AIRF(F)
CONTINUE
c
TO=((XIA-X1S)*(VS-VAX)-YIA*VAY)/(((VS-VAX)**P)*VAY*AY)
X=N+TO*XIS-VAX-T0-XIA
Y=VAY+TO*YIA
RPO=SORT(XO+XO*Y0+Y0)
RON=SORT(RPO+RON+RDN)
IF (RSS.NF.'K') GO TO 740
WSF=1.4RACWS
AL=CI/F
XI=(R.0+AL*X0)/(1+HI*(WSF**2))**0.5
CALL ENH(XI,FDXI)
W=WSF*30.4K
SM=SORT(SM*U(1,0-SDK)/2.0)
XIP=X1*X1
FRS='FSCO(SPI+1.0-FDK)X+EXP(-X12)*((1.0/(2.*X1P)-1.0)
1 /X1)/X1**3
SM=0.
IF (FRS.CF.0) SM=0.48*SORT(FRS)
OM=1.4B
IF (ARSDSM'.GT.1.E-18) OM=1.0/SM
OM=OM/OM
CALL FRCOM,FPCM)
FOM=8.8
FPW=0.8
IF (SNMPL.T.0) FPN=(SN+SM*(FOM+SPI)*OMX*(1.0+FKDF1))/
1 P.6**2/(P.6**p1)
PSHX=(P.6*XF**3)*U/P/100.0

c
C
TYPEF 101 XI,SW,SM,PS1A
101 FORMAT(' XI=1PF10.3', S1GW=' ',1PF10.3, '
1 ', PS1A=' ',1PF10.3)
C
TYPE 105,NOM,FNS,FCCOM,FPN
105 FORMAT(' NOM='1PF10.3', FRS='1PF10.3', FOM='1PF10.3,
1 ', FPN='1PF10.3)
C
IF (CLANS.FO. 'YES') OR (CLANS.FO. 'Y') GO TO 74A
TYPE 98A
FORMAT(' HOUGH SURFACE NO SURFACE DATA')
GO TO 751
74A
TYPE 979
979 FORMAT(' HOUGH SURFACE SURFACE DATA')
GO TO 751
C
740 IF (CLANS.FO. 'YES') OR (CLANS.FO. 'Y') GO TO 750
TYPF 701

701 FORMAT('"SMOOTH SURFACE: NO SURFACE DUCT"')
GO TO 751

750 TYPE 70P

70P FORMAT('"SMOOTH SURFACE: SURFACE DUCT"')

751 TYPF 70A TO 1, KPD, K0, F

70A FORMAT 'T0=\$F1\$", K"(T0)=\$F7\$, R(T0)=\$F7.0\$, F=F, F=0.3")

TYPF 7AC=1, K5S, XM, XC, YWC

7AC FORMAT 'K=\$F1\$, FT K5=\$F4.1\$, KTS, N=\$F5.5\$, NY=\$F5.2")

TYPF 703

703 FORMAT('"MDF T TA TDF K" K MDF D/F"')

C

T=TI

XSA=VS*T+XIS
XA=VAX*T+XIA
YA=VAY*T+YIA
XSA=XS-KA

RF=SORT(XSA+XSA+YA+YA)

R=SORT(RP+KPD+D)

FPR=FPR+(1.0F-1)

A=(A1+F02)/(1.0F+02)+(A0+F02)/(A100.0+F02))/3000.1

IF (RPR<LT, SPOP) GO TO 900

THFR=ACOS(CCS/GR)

GR=(GRM1)/S000(M1)

IF (.NOT.(LANS.EQ.,YES')) GO TO 900

IF(RPL=0) GO TO 740

CALL SDR(XSA, YA, ANH)

TYPF 70S, MDF, T, TA, TDF, RH, K5, DOP, PH1, PKS0, TL

FORMATCH K00, K01, K02, FR, FA, F, E1, E2, E3, E4, E5, E6, IFREE

CALL WHTG(Temp, TL, FR, FA, E0, E1, E2, E3, E4, E5, E6)

TYPF 725, MDF, T, TA, TDF, RH, K5, DOP, PH1, PKS0, TL

FORMATCH X00, X01, FA, FC, FC, F7, F7, F7, F7, F7, IFREE

IF(RRSSS, NF, "K") GO TO 740

CALL SDR(XSA, YA, ANH, MDF)

TYPF 97, MDF, T, RA, PK5, TLR5

FORMATCH A01, F02, .33X0, .4100, .39, IFREE

CALL WHTG(Temp, TL, RS, PK3, R3, PKA3, T3)

TYPF 754, MDF, T, T3

FORMATCH A04, F05, .33X0, 1)

CALL C2S(XSA, YA, THF, THFSD, KSN, NH, EXIST)

C

TYPF 100, THER, THFSD, KSN, "K"


IF (EXIST, NF, "Y") GO TO 50

TYPF 705, MDF, T, TA, TDF, RH, DOP, NF, PH1, PKS0, TL

CALL WHTG(Temp, TL, RP, K3, RP3, TL3)

TYPF 70S, MDF, T, RP, K3, RP3, TL3

IF (RSS, NF, "K") GO TO 50

CALL C2S(XSA, YA, THF, THFSD, KSN, NH, MDF)

TYPF 97, MDF, T, RP, PH, PK5, TLR5

CALL WHTG(Temp, TL, RS, PK3, R3, PKA3, T3)

TYPF 54, MDF, T, T3

GO TO 50

CALL C2S(XSA, YA, THF, KTN, KA, EXIST)

IF (EXIST, NF, "Y") GO TO 100

TYPF 705, MDF, T, TA, TDF, RH, DOP, NF, PH1, PKS0, TL

CALL WHTG(Temp, TL, RP, K3, RP3, TL3)

TYPF 953, MDF, T, RP, K3, RP3, TL3

IF (RSS, NF, "K") GO TO 50

CALL C2S(XSA, YA, THF, KTN, KA, MDF)

TYPF 97, MDF, T, RP, PH, PK5, TLR5

CALL WHTG(Temp, TL, RS, PK3, R3, PKA3, T3)
C

TYPE 954, MD0FR, T, TL3

C

B50

CONTINUE

C

990

T = T + DT

IF (T = LF, TF) GO TO 700

CCCC

TYPE 102, RA, KSD, KTN

102

FORMAT (' T , KA= ''F9.1'' , KSD= ''E9.1'' , KTN= ''C9.1'' )

C

2M=1

CALL WATIN(MS;1)

IF (T < . E0 - 13.) GO TO 30

C

END

C

SUBROUTINE SD(XSA, YA, ARPH)

C

COMMON /IN/ VAX, VAY, XIA, YIA, H, F, VS, AS, D, AXN, VS, XNC, YNC, TL, TF, DT

COMMON /OUT/ T, K, KP, TA, TDIF, TL, DOP, DF, PH1, KTSD, AR, X1, X2

COMMON /CC/ C1, C2, CS, CP, DR, DS, CMIN, MAX, DMIN, ETA, CS, CH, NO, TL, PI, I, GI

C

MOPD='SD'

COMMON '180, PI'

TAX+((H/C1)+(RP/C2))

TDIF=T-To+(KP-KO)/C2

PH=CONVACOS((-XSA)/KP)

VAX=SQR((VAX+VAX+VAY)/YAX)

CPS+((VAX-XSA-VAY+YA)/(VAX+YAX))

DOP=SQRT((1.0-2.0*VS+XSA/(C2+H))/((1.0-0.0*VAX+PS1/C2))

DOP=SQRT(2.0*(CMAX-CS)/CMAX)

IF(CON(DOP)

SG+((((CMAX-CMIN)/(DMIN-DS))*(1.0/3.0))

SSM=9.0

IF(WS-LT.10.0) SSM=4.75

IF(WS-LT.7.0) SSM=4.5

F0=F1*100.0

ATOT+((1.440+((F1**((K-3.0))

I+SG+(DS-DS))/SSM*SQRT((F1/DS))/3000.0)

ARPH=ATOT*KP

ARPH=AT

IF (ARPH<.CT.(-900.0)) ARPH=10.0*(ARPH/10.0)

PKS+((K-0.30)*AXN+AXV+(DOE+3.0)/CP)*US+AR

P+490.0

IF (PKS<.CF.ETA) TL=10.*ALOC10(PK5)

C

SUBROUTINE C2S(XSA, YA, THER, THEP, KSD, RA, EXIST)

C

COMMON /IN/ VAX, VAY, XIA, YIA, H, F, VS, AS, D, AXN, VS, XNC, YNC, TL, TF, DT

COMMON /OUT/ T, K, KP, TA, TDIF, TL, DOP, DF, PH1, KTSD, AR, X1, X2

COMMON /CC/ C1, C2, CS, CP, DR, DS, CMIN, MAX, DMIN, ETA, CS, CH, NO, TL, PI, I, GI

C

MOPD='C2S'

COMMON '180, PI'

TAX+((H/C1)+(RP/C2))

TDIF=T-To+(KP-KO)/C2
INPUT ACOS((-XSA)/RP)
VA=SRT(VAX+VAX+VAY+VAY)
CPSI=(VAX*VAX+VAY+VAY)/(VAX+VAX)
DOP=SRT((1.0-2.0*VX*VSA/(2*R))/((1.0-2.0*VX*CPSI/C2)))
AS=(CMAX-CS)/P5
R=(CMAX-CMIN)/(CMIN-C5)
RFAC=(1.0/G1)+(1.0/FR)
RFACP=(1.0/G5)+(1.0/G1)
NSD=P.0*CMAX+(CFAC*SORT(1.0-CMIN-CMIN/(CMAX-CMAX)))+ 
I SRT(1.0-CS/CS/(CMAX-CMAX)/AS)
RR=P.0*CCH+(CFAC*SORT(1.0-CMIN-CMIN/(CMAX-CMAX)))+ 
I SRT(1.0-CS/CS/(CMAX-CMAX)/AS)*RFACP*SORT(1.0-CMAX-CMAX/(CMAX-CMAX))
IF ((((MP+GE+RSN).AND.((RP+FR+RP)) GO TO 400
IF ((((MP+GE+RH).AND.((RP+LF+KFSD)) GO TO 400
500 FXIST='Y'
RETURN
C
400
EXIST='Y'
V=(VS-VAX)*(VS-VAX)+VAY+VAY
XY=(AIS-XIA)*(AIS-XIA)+YIA+YIA
ST=TRAS(T)
TI=ST=SRT(TN-TT-10-42-YR-RR)/V
TP=ST=SRT(TN-TT-10-42-YR-RR)/V
THFSD=ACOS(C5/CMAX)
DOE=THFSA*(T-T1)+(THFSD-THFSA)/(T2-T1)
DEF=COV*DOE
ARP=A+RP
ARPM=5.0
IF (ARP/GT.5.0) ARPM=5.0+(ARP/50.0)
PKSO=2.0+MAXN+AXN+(THFSA-THFSD+3.0)*4PM/
I (3.0*(THFSA+THFSD)*AXN(MPNKH-KSNKH-KSNKH))
TL=999.0
IF (PRSO.PF.FTA) TL=10.0*ALOC00(PKSO)
RETURN
C
C
SUBROUTINE CZ(XSA,YA,THFSA,RTA,THA,FXIST)
COMMON /CM/ VAX,YAXA,XIA,YIA,H,F,VS,XIS,D,AXN,WS,XHE,YMC,TI,TF,OT
COMMON /RUT/ T,K,P,T,R,TI,F0,F1,T0,FD,PKS,P0,AK,AL,
COMMON /CC/ C1,C2,C3,C4,DS,CM1,CM2,DMIN,FTA,CR,CR,KT,PL,PL
C
I MDF='CZ'
C
C
C
FROM T/ABS(T)
T1=T0+ST*SORT(T0*70-(XY-RR*RR)/V)
T2=T0+ST*SORT(T0*70-(XY-KTO*KTO)/V)
DIF=THF+1.0-(T-T1)/(T2-T1)
DIF=CON*DOF
APR=A*APR
AKPM=0.0
IF (AR=AT,(=-500.0)) AKPM=10.0*(AKP/10.0)
PRSD=3P.0*XNN+XNN*THF+THF+AKPM/(3.0+AR(CHARP+HR-KTO*KTO))
TL=999.0
IF (PRSD.GF.FTA) TL=10.0*ALOG10(PRSD)
C
RETURN
C
END

SUBROUTINE SDR(XSA,YA,AKPM,MODE)
COMMON /IN/ VAX,VAY,XTA,YTA,H,F,VS,XIS,D,AXN,WS,XNC,YNC,TH,TF,IT
COMMON /OUT/ T,T,RP,Ta,TDF,TL,DDP,DF,PHI,PRSD,AKPM
COMMON /CC/ C1,C2,CS,CR,DE,DS,CMIN,CMAX,FTA,CS,CR,HR,TO,PI
I GI
COMMON/RS/ SW,CPW,PSRH,PRN,CI,PRS,TLKS,RO

MODE=' SR'
CPW=(XNC*XSA-YWC*YAA)/RP
THF=DEF.XA.XS/Y25/XR0.0
CALL INT(THF)
FC=1.0/FIB
IF (SW.NE.0.0) FC=1.0/(SORT(2.0)*SV)
CTA=COS(THF/P.0)
STA=SIN(THF/P.0)
ARG1=STA*COS/CTA
ARG2=FC/(0.0-ST4)/CTA
CALL FRF(ARG1,FRF1)
CALL FRF(ARG2,FRF2)
N=FPN*FRF1+FRF2
PKS=(R.0*AXN+XNN+C1*RN+AKPM)/(RH+DS)
TLRS=999.0
IF (PRSD.GF.FTA) TLRS=10.0*ALOG10(PRSD)
C
RETURN
C
END

SUBROUTINE CZSR(XSA,YA,THF,THESD,KS1,KA,MODE)
COMMON /IN/ VAX,VAY,XTA,YTA,H,F,VS,XIS,D,AXN,WS,XNC,YNC,TH,TF,IT
COMMON /OUT/ T,T,RP,Ta,TDF,TL,DDP,DF,PHI,PRSD,AKPM
COMMON /CC/ C1,C2,CS,CR,DE,DS,CMIN,CMAX,FTA,CS,CR,HR,TO,PI
I GI
COMMON/RS/ SW,CPW,PSRH,PRN,CI,PRS,TLKS,RO

MODE=' C2SR'
CPW=(XNC*XSA-YWC*YAA)/RP
CALL INT(THF)
CI=CI
CALL INT(THESD)
FC=1.0/FIB
IF (SW.NE.0.0) FC=1.0/(SORT(2.0)*SV)
THS=THF+THESD
CTSA=COS(THS/P.0)
STA=SIN(THS/P.0)
ARG1=STA*COS/CTS
ARG2=FC/(0.0-ST5)/CTS
APR=FC/(1.0-STS)/CTS
CALL FRF(ARG1,FRF1)
CALL FHF(ANKP, FRP2)
RM=FPW(FHF1+FRF2)
ANKP=0,0
APR=ANKP
IF(APK.CI=.500.0) ANP=10.0*(APK/10.0)
PKS=32.0*AAX*AAX*(C10-CI)*AHPI*PN/(THS*APR(KP-KN=KS0*KS1))
TIRS=-999.0
IF (PKS,CF,ETA) TIRS=10.0*ALOG10(PKS)
NRETURN
FND

SUBROUTINE CZK(XSA, Y+1, THEN, KTO, NR, MODE)
COMMON /VAX/VAX, VAY, XI, Y1, X2, Z1, Z2, XL, AN, YK, XN, YC, TI, TN, T
COMMON /OUT/ T. K, KF, TA, TDF, TL, DOP, DF, PHI, PK, SU, A, WOLF
COMMON /CC/ CI, CP, CS, CP, BB, CS, CPU, CMAX, IN, ETA, F, G, H, KO, TN, PI, 
1 A1
COMMON/KS/ SW, CPW, PS10, FPW, CI, PKS, T. KS, RV
COMMON/A0/ SW, CPW, PS10, FPW, CI, PKS, T. KS, RV

MODE=1 CZK
CPW=(XKC+XSA+YXC+YA)/K
CALL INT(THEN)
FC=1.F1R
IF (SW,AN,0.0) FC=1.0/(SORT(2.0)*SW)
ANKP=THF/2.0
CTP=COS(ANKP)
STP=SIN(ANKP)
ARKP=1.STP/FC/CTP
ARKP=90.0/STP/CTP
CALL FHF(ANKP, FRF1)
CALL FHF(ANKP, FRF2)
APR=ANKP
IF(APK,CR,-.500.0) ANP=10.0*(APK/10.0)
PKS=32.0*AAX*AAX*(C10-CI)*ANKP/(THF*ANKP(KP-KN=KS0*KS1))
TIRS=-999.0
IF (PKS,CF,ETA) TIRS=10.0*ALOG10(PKS)
NRETURN
FND

SUBROUTINE INT(TH)
COMMON/KS/ SW, CPW, PS10, FPW, CI, PKS, T. KS, RV

IF (ABS(SW),CT,1.0,E-14) GO TO 1P
C=(TH+PS1)0+CPW)**3 -(PS1+CPW)**3)/3.0
RETURN

1P
SOP=SRT(3.14159265)
SU=SRT(2.0)
T=90.0/3.0
ET=5.0/3.0

PC1=PS1+CPW/(SOP+SW)
PC2=PCP+PCP
TC=TH/(SOP+SW)
ARKP=TC
ARKP=ANKP
CALL FHF(ANKP, FRF1)
CALL FHF(ANKP, FRF2)
C=((SOP+SW)**3)/A.0)*((TS+TT)AKP2*ANK-P CSP-PCP)*FRF1*(1.0+TT 1)*AKP2)*ANK-FRF2*(1.0+TT)CSP*PCP-FRF2*(AKP-FF)/(SU^2)
C RFTURN
C FND
C SUBROUTINE FRF(ARG,FRFS)
C SARG=ARG/ANS(ARG)
C X=AHS(ARG)
C FR=1.0
C IF (X<.FT.4.2) GO TO 200
C F=1.0+.070523074*X+.049920572*(X**2)
C * .000155143*(X**4) +.0002765672*(X**5) +.000040135*(X**6)
C FR=1.0-1.0/(F+1A)
C FRFS=FR*FRF
C RETURN
C END
C SUBROUTINE WOUT(UTFMP,TL3,RP3,R3,KPA3,TL3)
C COMMON /IN/ VAX,VAY,XIA,YIA,*F,VS,X'S,DAAXNS,WS,XWC,YWC
C COMMON /OUT/ TR,RP,TA,TDIF,TL,DP,D,PHI,PSO,A,MODE
C RP3=RP/3000.0
C R3=R/3000.0
C X3=(VS*TA*XIA)-(VAX*TA*XIA)
C Y3=(VAY*TA*YIA)
C KPA3=SQRT(X3*X3+Y3*Y3)/3000.0
C TL3=-999.0
C IF (T.L.UTMP<9.99) RETURN
C C=0.5(-0.9+ALOG10(9.9))
C C=0.5(-0.9+ALOG10(9.9))
C IF (OUTMP.EQ.0 .Y) OR .NOT.(OUTMP.EQ.0 .Y) C=K=.7,4,4,4,4
C TL3=TLU+TL9-CORK
C RETURN
C END
C SUBROUTINE WATIN(IN4,K4)
C COMMON /IN/ VAX,VAY,XIA,YIA,*F,VS,X'X',DAAXNS,WS,XWC,YWC,TI,TL,DT
C COMMON /OUT/ TR,RP,TA,TDIF,TL,DP,D,PHI,PSO,A,MODE
C COMMON /CC/ C1,C2,C3,C4,CH,DR,DS,CMIN,CMAX,DMIN,DMAX,TA,CS,CH,NO,FL,FL
C COMMON /MM/MM
C M=IN4
C IF(M<.AT.1) GO TO 3000
C TYPE 3000
C FORMAT('TO CHANGE A RUN PARAMETER, ENTER THE'
C \!* APPROPRIATE NUMBER:/',
C \!* 1 AIRCRAFT VEL. VECTORS',
C \!* 2 AIRCRAFT INITIAL POSITION',
C \!* 3 AIRCRAFT HEIGHT',
C \!* 4 AIRCRAFT RADIATED FREQ',
C \!* 5 SUR VFL. VECTOR',
C \!* 6 SUR INITIAL POSITION',
C \!* 7 SUR DEPTH',
C \!* 8 SURFACE SOUND SPEED',
C \!* 9 BOTTOM SOUND SPEED AND EPS',
C \!* 10 MINIMUM SOUND SPEED AND DEPTH',
C \!* 0 END RUN'
C \!')
**Bolt Beranek and Newman Inc.**

```
1 11 MAX SHINE SPEED AND HETIN/
1 12 TIME OF EVENTS/
1 13 RUN/
1 14 STOP/
1 15 WIND PARAMETERS/
1 ENTER THE APPROPRIATE NUMBER = , S
ACCEPT 2001, XS

2001 FORMAT()
GO TO (47, 48, 49, 10, 11, 12, 13, 14, 15, 16, 19, 20, 36, 18, 984) X 3

3001 FORMAT(' CHANGE PARAMETER = ', $)
ACCEPT 3002, X4

3002 FORMAT()
GO TO (47, 48, 49, 10, 11, 12, 13, 14, 15, 16, 19, 20, 36, 18, 984) X 4

47 CALL AIRV(VAX, VAY)
GO TO 3000

48 CALL AIRC(XIA, YIA)
GO TO 3000

49 CALL AIRH(N)
GO TO 3000

10 CALL AIRE(F)
GO TO 3000

11 CALL SHRV(US)
GO TO 3000

12 CALL SIRC(XIS)
GO TO 3000

13 CALL SHRD(D)
GO TO 3000

14 CALL FVND(VS)
GO TO 3000

15 CALL FVNC(CR, DR)
GO TO 3000

16 CALL FVNC(CMIN, DMIN)
GO TO 3000

17 CALL FVNA(CMAX, DS)
GO TO 3000

20 CALL TIM(T1, TF, DT)
GO TO 3000

18 CALL EXIT

984 CALL WIND(KS, XWC, YWC)
GO TO 3000

C 30 W3 = 13
RETURN

C  END

C SUBROUTINE WIND(KS, XWC, YWC)
TYPE 200

200 FORMAT(1H, ' WIND SPEED = ', $)
ACCEPT 2001, WS

201 FORMAT(F)
TYPE 202

202 FORMAT(' X DIRECTION COSINE OF WIND = ', $)
ACCEPT 2001, XWC

203 FORMAT(' Y DIRECTION COSINE OF WIND = ', $)
ACCEPT 2001, YWC
RETURN

END

SUBROUTINE AIRV(VAX, VAY)
TYPE 200

200 FORMAT(1H, ' VELOCITY VECTOR X-DIRECTION(KTS) = ', 2X, $)
ACCEPT 2001, VA XT

201 FORMAT(F)
VAX = 1.488 * VA XT
```
TYPE 202
FORMAT('+VFLATENCY VECTOR Y-DIRECTION(KTS)=','2X,')
ACCEPT 201,VAYKT
VAY=1.0ARDS*VAYKT
RETURN
END
SUBROUTINE AIRC(XIA,YIA)
TYPE 200
FORMAT('*INITIAL X-COORDINATE OF AIRCRAFT(FT)=','2X,')
ACCEPT 201,XIA
201 FORMAT(F)
TYPE 202
FORMAT('*INITIAL Y-COORDINATE OF AIRCRAFT(FT)=','2X,')
ACCEPT 201,YIA
RETURN
END
SUBROUTINE AIRH(H)
TYPE 200
FORMAT('HEIGHT OF AIRCRAFT FROM SEA SURFACE(FT)=','2X,')
ACCEPT 201,H
201 FORMAT(F)
RETURN
END
SUBROUTINE AIRF(F)
TYPE 200
FORMAT('FAC(0) OF AIRCRAFT RADIATION(HZ)=','2X,')
ACCEPT 201,F
201 FORMAT(F)
RETURN
END
SUBROUTINE SUBV(VS)
TYPE 200
FORMAT('VFLATENCY VECTOR X-DIRECTION SUR(KTS)=','2X,')
ACCEPT 201,VSKT
201 FORMAT(F)
VS=1.0ARDS*VSKT
RETURN
END
SUBROUTINE SUBC(XIS)
TYPE 200
FORMAT('INITIAL X-COORDINATE OF SUR(FT)=','2X,')
ACCEPT 201,XIS
201 FORMAT(F)
RETURN
END
SUBROUTINE SUBD(D)
TYPE 200
FORMAT('DEPTH OF SUR(FT)=','2X,')
ACCEPT 201,D
201 FORMAT(F)
RETURN
END
SUBROUTINE SFNS(CS)
TYPE 200
FORMAT('SURFACE SOUND SPEED(FT/SFC)=','2X,')
ACCEPT 201,CS
201 FORMAT(F)
RETURN
END
SUBROUTINE SFNC(CD,DC)
TYPE 200
FORMAT('BOTTOM SOUND SPEED(FT/SFC)=','2X,')
ACCEPT 201,CH
201 FORMAT(F)
TYPE 202
FORMAT('DEPHT FOR BOTTOM SOUND SPEED(FT)=','2X,')
A-40
ACC, PT 201, DMIN
RETURN
END
SUBROUTINE ENVLD(CMIN, DMIN)
TYPE 200

FORMAT(1H , 'MIN. SOUND SPEED(FT/SEC)= ', 2X, 1)
ACCEPT 201, CMIN

RETURN
END
SUBROUTINE ENVMD(MAX, DSH)
TYPE 200

FORMAT(1H , 'MAX. SOUND SPEED(FT/SEC)= ', 2X, 1)
ACCEPT 201, MAX

RETURN
END

SUBROUTINE TIV(TI, TF, DT)
TYPE 200

FORMAT(1H , 'INITIAL TIME= ', 2X, 1)
ACCEPT 201, TI

RETURN
END

RETURN
END
APPENDIX B

DIRECT PATH AND BOTTOM BOUNCE SUBROUTINE

Introduction

To calculate the transmission loss as a function of time for a moving source above the sea surface to a moving receiver below the sea surface through varying sea states (smooth to very rough), a computer program was developed. The sections which follow will be concerned with two modes of propagation from the source to receiver; direct and bottom bounce rays. The bottom bounce rays will be further subdivided into single bottom bounce or double bottom bounce rays.

Finally, two types of sound speed profiles, one with a surface duct present and one without, will be utilized.

To perform the calculations for the different modes of propagation involved, 15 separate subroutines are used with one main program as the control. Below is a listing of the main and subroutine programs used.

<table>
<thead>
<tr>
<th>MAIN PROGRAM</th>
<th>SUBROUTINES</th>
</tr>
</thead>
<tbody>
<tr>
<td>WATER.F4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BOUN(THE0)</td>
</tr>
<tr>
<td></td>
<td>SBOUN(THE0)</td>
</tr>
<tr>
<td></td>
<td>BS(THE0)</td>
</tr>
<tr>
<td></td>
<td>BD(THE0)</td>
</tr>
<tr>
<td></td>
<td>BSS(THE0)</td>
</tr>
<tr>
<td></td>
<td>BDS(THE0)</td>
</tr>
<tr>
<td></td>
<td>BSR(THE0)</td>
</tr>
<tr>
<td></td>
<td>BDR(THE0)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BSSR(THE0)</td>
</tr>
<tr>
<td></td>
<td>BDSR(THE0)</td>
</tr>
<tr>
<td></td>
<td>DP(THE0)</td>
</tr>
<tr>
<td></td>
<td>DPR(THE0)</td>
</tr>
<tr>
<td></td>
<td>ERFS.F4</td>
</tr>
<tr>
<td></td>
<td>BT3.F4</td>
</tr>
<tr>
<td></td>
<td>BT5.F4</td>
</tr>
</tbody>
</table>

B-1
Figure E-1 is a block diagram showing the flow of the calculations. This chart gives an overall picture of the flow of the program calculations during the actual execution.

B.1 Brief Description and Purpose of Program

**MAIN PROGRAM - WATER.F4**

The main program has several functions. One of these is accepting all the input parameters which initialize the problem. The input parameters are divided into four main categories: (1) source parameters, (2) receiver parameters, (3) environmental parameters, and (4) time parameters. Once all the parameters are inputted, a series of calculations are performed to obtain a set of constants for each particular time. These are stored in a common block with all the input parameters so that they can be shared by all the subroutines. The third function of the main program is to route the flow through 4 main loops by a series of conditional statements. Finally, the main program controls the output of values from each individual subroutine so that the values are displayed in a pre-assigned format.

**BOUND(THE8)**

This subroutine calculates the angle of the arriving ray which reaches the receiver with no surface duct present. It first determines whether there is a single or double bounce which reaches the receiver. An iteration routine is then used to finally calculate the acquired ray if there is one. (NOTE: if no single bounce exists, the program defaults to double bounce; if the double bounce does not exist, the control is sent back to main program where DP.F4 is called.) Now depending on whether the solution is a single or double bounce, the main control is either sent to BS(THE8) or BD(THE8).
SBOUN(THE0)

The difference between SBOUN(THE0) and BOUN(THE0) is that a surface duct is present for the calculation of the angle of the arriving ray which reaches the receiver. Again, it also has to determine first whether there is a single or double bounce which reaches the receiver with the surface duct present. An iteration routine is then used to finally calculate the acquired ray, if there is one. Depending on whether the solution is a single or double bounce, the main control is sent either to BSS(THE0) or BDS(THE0).

BS(THE3)

The single bounce rays with no surface duct present, are handled within this subroutine. Besides calculations to obtain the TL for this particular mode of propagation, other important quantities of interest are also calculated. The final output values from this subroutine are sent to the main program where they are printed out using the main program format.

BD(THE9)

This is the double bounce subroutine which performs all the calculations for double bounce rays with no surface duct present. Many of the formula are modified versions of the single bounce equations. The final output values from this subroutine are again sent to the main program where they are printed out.

BSS(THE0)

For bottom bounce rays which are in the presence of a surface duct, the calculations are performed within this subroutine. The final output values are returned to the main program where they are printed out.
**BCS**(THE8)

This is the subroutine which calculates double bounce rays propagation with a surface duct present.

**BSR**(THE8)

All subroutines with the letter R contained in the name are used for rough surface calculations. This subroutine modifies the smooth surface results for the single bounce mode of propagation. It has the necessary equations to calculate TL for a rough sea surface. Once all the output values are calculated, they are sent to the main control program to be printed out. No surface duct is present.

**BDR**(THE8)

This is the subroutine which handles rough surface double bounce propagation (no surface duct present). Again it returns to main program with a new PRATIO and TL for this mode of propagation in a rough sea.

**BSSR**(THE8)

To take into account a surface duct in a rough sea, subroutine BSSR(THE8) handles the single bottom bounce mode of propagation for this condition. It calculates a new PRATIO and TL due to the change in sea surface conditions. It returns to main program with output values.

**BDSR**(THE8)

This subroutine calculates the double bounce ray for a rough sea surface and surface duct present. It modifies the smooth surface formula and calculates a new PRATIO and TL due to the rough sea state. It returns to main program with output values.
For direct path propagation all calculations are handled within this subroutine for smooth surface conditions.

The direct path propagation TL with a rough sea surface are handled by the formula which are modified to take into account the rough sea surface.

To obtain the statistics for the rough surface subroutines, the error function is necessary. This subroutine calculates the error function for a given input.

These two subroutines are used to obtain the bottom reflection loss as a function of frequency and grazing angle. Bottom types 3 or 5 are typical of most of the ocean bottoms of the world. This data was obtained from the FACT model.

To execute the calculations, 6 programs must be loaded. They are WATER.F4, B0.F4, SBO.F4, BT3.F4, BT5.F4, and ERFS.F4. Once loaded, the main program starts the control which asks the user to input all the parameters necessary for the calculation. The program asks for the following inputs listed below: (The inputs must be floating point numbers.)
### INPUT

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aircraft velocity x-direction</td>
<td>knots</td>
<td>VAX</td>
</tr>
<tr>
<td>Aircraft velocity y-direction</td>
<td>knots</td>
<td>VAY</td>
</tr>
<tr>
<td>Initial x-coordinate of aircraft (at t=0)</td>
<td>feet</td>
<td>XIA</td>
</tr>
<tr>
<td>Initial y-coordinate of aircraft (at t=0)</td>
<td>feet</td>
<td>YIA</td>
</tr>
<tr>
<td>Height of aircraft from sea surface</td>
<td>feet</td>
<td>H</td>
</tr>
<tr>
<td>Frequency of aircraft radiation</td>
<td>Hz</td>
<td>F</td>
</tr>
<tr>
<td>Receiver velocity x-direction</td>
<td>knots</td>
<td>VS</td>
</tr>
<tr>
<td>Initial x-coordinate of receiver (at t=0)</td>
<td>feet</td>
<td>XIS</td>
</tr>
<tr>
<td>Depth of receiver</td>
<td>feet</td>
<td>D</td>
</tr>
<tr>
<td>Wind speed</td>
<td>knots</td>
<td>V</td>
</tr>
<tr>
<td>x-direction cosine of wind direction</td>
<td>dimensionless</td>
<td>WSX</td>
</tr>
<tr>
<td>y-direction cosine of wind direction</td>
<td>dimensionless</td>
<td>WSY</td>
</tr>
<tr>
<td>Bottom type (3 or 5)</td>
<td>dimensionless</td>
<td>BT</td>
</tr>
<tr>
<td>Starting time</td>
<td>seconds</td>
<td>TI</td>
</tr>
<tr>
<td>Final time</td>
<td>seconds</td>
<td>TF</td>
</tr>
<tr>
<td>Time increments</td>
<td>seconds</td>
<td>DT</td>
</tr>
<tr>
<td>Is there a surface duct</td>
<td>yes/no</td>
<td></td>
</tr>
<tr>
<td>Surface sound speed</td>
<td>ft/sec</td>
<td>CS</td>
</tr>
<tr>
<td>Bottom sound speed</td>
<td>ft/sec</td>
<td>CB</td>
</tr>
<tr>
<td>Depth for bottom sound speed</td>
<td>ft</td>
<td>DB</td>
</tr>
<tr>
<td>Minimum sound speed</td>
<td>ft/sec</td>
<td>CMIN</td>
</tr>
<tr>
<td>Depth of minimum sound speed</td>
<td>ft</td>
<td>CMAX</td>
</tr>
<tr>
<td>Maximum sound speed</td>
<td>ft/sec</td>
<td>CMAX</td>
</tr>
<tr>
<td>Depth at maximum sound speed</td>
<td>ft</td>
<td>DS</td>
</tr>
</tbody>
</table>

The output is formatted such that a heading is printed first with the following names: MODE, T, TA, TDIF, R', R, LDF, D/E, PHI, PRATIO, TL.

- **MODE** - Mode of propagation
- **T** - Start time of ray
- **TA** - Time of arrival of ray
TDIF - difference between TA and time of arrival of direct path ray from CPA point
R' - Lateral range between source and receiver
R  - Slant range between source and receiver
DOP - Doppler
D/E - Arrival angle at receiver
PHI - Azimuthal arrival angle
PRATIO - Received mean square pressure
TL  - Transmission loss between source and receiver

The corresponding value for each heading is given in a column below the headings.

Once the set of calculations have been performed the program prints out a message to ask the user whether he would like to run the program again with any of the input parameters changed. The user may opt to run with a new set of parameters or stop the execution totally by typing the pre-assigned number.

The following pages are a listing of a sample run showing how the input parameters are given and the actual formatted output.
Here, "initial" refers to t=0.

+ In this statement, "initial" refers to the starting time of the encounter.
TRYING A SURFACE DUCT WITH DOUBLE BOUNCE SOLUTION

**OUTPUT PARAMETERS**

<table>
<thead>
<tr>
<th>THEO</th>
<th>0.131E+02</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODE</td>
<td>T</td>
</tr>
<tr>
<td>BDS</td>
<td>1100</td>
</tr>
<tr>
<td>0.3544E+00</td>
<td>0.3833E+00</td>
</tr>
<tr>
<td>BDSR</td>
<td>1100</td>
</tr>
<tr>
<td>DP</td>
<td>1100</td>
</tr>
<tr>
<td>0.3544E+00</td>
<td>0.3833E+00</td>
</tr>
</tbody>
</table>

**EXTRANEOUS: ONLY FOR CHECKING MECHANICS OF PROGRAM**

**TO CHANGE A FUN PARAMETER, ENTER THE APPROPRIATE NUMBER:**

1. AIRCRAFT VEL. VECTORS
2. AIRCRAFT INITIAL POSITION
3. AIRCRAFT HEIGHT
4. AIRCRAFT RADIATED FREO.
5. SUB VEL. VECTOR
6. SUB INITIAL POSITION
7. SUB DEPTH
8. SURFACE SOUND SPEED
9. BOTTOM SOUND SPEED AND DEPTH
10. MINIMUM SOUND SPEED AND DEPTH
11. WIND SPEED AND DIRECTION COSINES
12. BOTTOM TYPE
13. MAX. SOUND SPEED AND DEPTH
14. TIME OF EVENTS
15. RUN
16. STOP

ENTER THE APPROPRIATE NUMBER = 16

EXIT

. K.0E

User 2017-11 Job 27 ISCH2 OFF TTY15 at 10:24 AM Thu 18-Dec-75
Connect Time 0:12 CPU's 354
B.3 Example of Program Flow

For the user to understand the program flow, an actual case will be simulated. This way the user may follow through to see which subroutines and formula are used for one specific case.

As shown in the section on execution of the program, all the inputs are manually inserted first. Once this is done the main program calculates a series of constants set up by the main program internally. These are constants which do not change with time. A listing of these constants are given below:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Computer Program Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS = (CS-CMIN)/DMIN</td>
<td>GS (surface gradient)</td>
</tr>
<tr>
<td>GB = (CB-CMIN)/(DB-DMIN)</td>
<td>GB (bottom gradient)</td>
</tr>
<tr>
<td>C1 = 1100</td>
<td>C1 (sound speed in air, in ft/sec)</td>
</tr>
<tr>
<td>AXN = C1/C2</td>
<td>AXN (ratio of air sound speed and water sound speed)</td>
</tr>
<tr>
<td>C2 = 5000</td>
<td>C2 (average sound speed in water in ft/sec)</td>
</tr>
</tbody>
</table>

\[
t_0 = \frac{(X_1 - X_0)(V_1 - V_0) - (Y_1 - Y_0)}{[2(V_1 - V_0)^2 + (Y_1 - Y_0)^2]}\]

TO time of closest approach

\[
R(t_0) = [(x_s(t_0) - x_a(t_0))^2 + y_a^2(t_0) + D^2]^{1/2}
\]

RO slant range at closest approach

\[
A = \left[ \frac{1(F_{kHz})^2}{1 + F_{kHz}^2} + \frac{40(F_{kHz})^2}{4100 + F_{kHz}^2} \right] \frac{1}{3000}
\]

A volumetric attent (DB/ft)

\[
F_{kHz} = F/1000
\]
\[ R'(t) = \left[ (X_s(t) - X_a(t))^2 + Y_a^2(t) \right]^{1/2} \text{ RP Lateral Range} \]

\[ R(t) = \left[ (X_s(t) - X_a(t))^2 + Y_a^2(t) + D^2 \right]^{1/2} \text{ R slant range} \]

If a surface duct present then:

\[ G_1 = \frac{(C_{\text{MAX}} - C_{\text{MIN}})}{(D_{\text{MIN}} - D_S)} \]

\[ G_S = \frac{(C_{\text{MAX}} - C_S)}{D_S} \]

With all of the necessary parameters inputted and all the constants initialized, the main program sets up a common storage area so that all this data may be shared with all other subroutines.

Now with all the necessary parameters initialized and set up so that they can be shared by all subroutines, the main program now tests for whether there is a surface duct or not. For our case there is no surface duct so the subroutine BCUN(THEO) is called. This subroutine calculates the arrival angle of the ray which reaches the receiver. At a time instant \( t \), the subroutine first tests whether there is a single or double bounce and this is done by testing whether the distance to the convergence zone is less than or greater than the lateral distance between the receiver and the virtual source.

\[ R_b = 2c_b \left[ \frac{1}{g_S} \left( \sqrt{1 - \left( \frac{c_{\text{MIN}}}{c_b} \right)^2} - \sqrt{1 - \left( \frac{c_S}{c_b} \right)^2} \right) + \frac{1}{g_b} \sqrt{1 - \left( \frac{c_{\text{MIN}}}{c_b} \right)^2} \right] \]

Lateral receiver/virtual source range at \( t \)

\[ B(\theta) = \left[ \left( X_a(t - \frac{h}{c_i}) - X_s(t) \right)^2 + Y_a^2(t - \frac{h}{c_i}) \right]^{1/2} \]

B-12
If $B(\theta) > R_b$ then it is a double bounce, if not it is a single bounce. Now an iteration routine is used to finally calculate the acquired ray angle. Once this is obtained and for purposes of simulation a single bounce solution will be used so that subroutine BS(THE0) is called. The following calculations are performed within BS(THE0).

Bounce distance: \[
B(\theta_0) = \frac{2c_s}{\cos \theta_0} \left[ \frac{1}{g_s} \left( \sqrt{1 - \left(\frac{c_{\min}}{c_s}\right)^2 \cos^2 \theta_0} + \sin \theta_0 \right) \right] + \frac{1}{g_b} \left( \sqrt{1 - \left(\frac{c_{\min}}{c_s}\right)^2 \cos^2 \theta_0} - \sqrt{1 - \left(\frac{c_b}{c_s}\right)^2 \cos^2 \theta_0} \right)
\]

Travel time along BB ray: \[
t(\theta_0) = \frac{2}{g_s} \left[ \cosh^{-1} \left( \frac{c_s}{c_{\min} \cos \theta_0} \right) - \cosh^{-1} \left( \frac{1}{\cos \theta_0} \right) \right]
- \frac{2}{g_b} \left[ \cosh^{-1} \left( \frac{c_s}{c_{\min} \cos \theta_0} \right) - \cosh^{-1} \left( \frac{c_b}{c_{\min} \cos \theta_0} \right) \right]
\]

Arrival angle: \[\Phi/E = \Theta_0\]

\[
\frac{dB(\theta_0)}{d\theta_0} = B(\theta_0) \tan \theta_0 + 2c_s \left[ \frac{1}{g_s} \left\{ \frac{(c_{\min})^2 \sin \theta_0}{c_s} \sqrt{1 - \left(\frac{c_{\min}}{c_s}\right)^2 \cos^2 \theta_0} - 1 \right\} \right]
+ \frac{\sin \theta_0}{g_b} \left\{ \frac{(c_{\min})^2}{c_s} \sqrt{1 - \left(\frac{c_{\min}}{c_s}\right)^2 \cos^2 \theta_0} - \frac{(c_b/c_s)^2}{\sqrt{1 - \left(\frac{c_b}{c_s}\right)^2 \cos^2 \theta_0}} \right\}
\]

*If no solution for either single or double bounce, the control is sent back to main program where DP.F4 is called.
PATH LENGTH: \( s(\theta_0) = \frac{2c_s}{\cos \theta_0} \left[ \left\{ \frac{1}{g_s} + \frac{1}{g_b} \right\} \sin^{-1} \left( \frac{c_{\min} \cos \theta_0}{c_s} \right) + \frac{1}{g_b} \sin^{-1} \left( \frac{c_b \cos \theta_0}{c_s} \right) \right. \right.
\[ \left. + \frac{1}{g_s} \left( \frac{\pi}{2} - \theta_0 \right) \right] \]

AZIMUTHAL ARRIVAL ANGLE: \( \cos \phi(t_A) = \frac{X_a(t - \frac{1}{v_c}) - X_s(t + t_A(\theta_0))}{\left[ \left( X_a(t - \frac{1}{v_c}) - X_s(t + t_A(\theta_0)) \right)^2 + y_a^2(t - \frac{1}{v_c}) \right]^{1/2}} \]

AUXILIARY ANGLE FOR DOPPLER:
\[ \cos \psi = \frac{\bar{\nu}_ax \left[ X_s(t + t_A(\theta_0)) - X_a(t - \frac{h}{v_c}) \right] + \bar{\nu}_ay \left[ -y_a(t - \frac{h}{v_c}) \right]}{|\bar{\nu}_a| \left[ \left( X_s(t + t_A(\theta_0)) - X_a(t - \frac{h}{v_c}) \right)^2 + y_a^2(t - \frac{h}{v_c}) + (2D_E - D)^2 \right]^{1/2}} \]

DOPPLER SHIFT:
\[ \frac{f_z}{f_s} = \left[ 1 + \frac{\bar{\nu}_ax^2}{v_c^2} - 2 |\bar{\nu}_a| \cos \psi \right]^{1/2} \cdot \left[ 1 + \frac{\bar{\nu}_ay^2}{v_c^2} - \frac{2 |\bar{\nu}_a| \left[ X_s(t + t_A(\theta_0)) - X_a(t - \frac{h}{v_c}) \right]/v_c}{\left[ \left( X_s(t + t_A(\theta_0)) - X_a(t - \frac{h}{v_c}) \right)^2 + y_a^2(t - \frac{h}{v_c}) + (2D_E - D)^2 \right]^{1/2}} \right]^{1/2} \]

TIME OF ARRIVAL MINUS TIME OF ARRIVAL OF DIRECT PATH FROM CPA POINT:
\[ t_A(t) - t_{A_{\max}} = t + t_A(\theta_0) - t_0 - \frac{R(t_0)}{v_c} - \frac{h}{v_c} \]
At this point, the angle of the grazing ray to the bottom is calculated and depending on which bottom type (3 or 5) is inputted, subroutine BT3.P1 or BT5.P1 is called with the grazing ray angle and returns with a bottom loss*. The subroutine BT3.P1 or BT5.P1 has a series of linear equations which are good for specific frequency and grazing angle ranges. The subroutine interpolates to find the best fit. The bottom loss values were programmed using LRAPP data.

It is now possible by combining several of the parameters calculated already to obtain the mean square pressure at the receiver and the TL.

\[
\frac{P^2}{P_0^2} = \frac{8 m^2 \sin \theta_0 \cos \theta_0}{B(\theta_0)} \left\{ \frac{\partial B(\theta_0)}{\partial \theta_0} \right\} - \frac{\{A S(\theta_0) + R B L(\theta_0)\}}{10} 
\]

\[
TL = 10 \log_{10} \left\{ \frac{8 m^2 \sin \theta_0 \cos \theta_0}{B(\theta_0)} \left\{ \frac{\partial B(\theta_0)}{\partial \theta_0} \right\} \right\} - A S(\theta_0) - R B L(\theta_0)
\]

A correction for air attenuation is added to the TL.*

The control is now sent back to the main program where the values of MODE, T, TA, TDIF, R', R, DOP, D/E, PHI, PRATIO, TL are printed in a preassigned format.

The main program now goes into a series of conditional statements to find out which mode was just calculated for the smooth sea surface case, (which in this case the mode = BS) and routes the program to call the subroutine for the rough surface case which for BS is BSR(THET0).

* See Eq. 64, Appendix A.
Once within this subroutine a series of rough surface formula to set up the statistics for the rough surface case are calculated as given below:

Azimuthal angle between plane of acoustic path and wind direction at arrival time $t_a$

$$\cos \phi_w(t_a) = \frac{m_{sx} \left\{ X_s(t+t(a)) - X_s(t-h\xi_i) \right\} - m_{sy} y_a(t-h\xi_i)}{\left[ (X_s(t+t(a)) - X_s(t-h\xi_i))^2 + y_a(t-h\xi_i)^2 \right]^{1/2}}$$

AUXILIARY VARIABLE:

$$\xi = \left\{ \frac{8h \lambda_{air} g^2}{\pi^5 U^4} \right\}^{1/4}$$

Having a value of zeta ($\xi$), the subroutine now calls a separate subroutine ERFS( ) to obtain a value of the error function for that particular value of $\xi$. Once the value is obtained it is returned to the calling subroutine.

Mean square sea slope (fully arisen sea)

$$\sigma^2 = 1.15 \times 10^8 \sqrt{\frac{U}{g}} \left[ 1 - \text{erf}(\xi) \right]$$

rms slope in wind direction

$$\sigma_w = \sigma / \sqrt{2}$$
Most probable slope estimate

$$\bar{\psi} = \frac{2.86}{1000} U \left( \frac{cm}{sec} \right) \frac{\pi}{180}$$

Mean square sea surface curvature:

$$\bar{\eta}^2 = 2.3 \times 10^4 \frac{g}{g U^3} \frac{4}{3} \left[ \sqrt{\pi} \left\{ 1 - \text{erf} \left( \frac{1}{2} \right) \right\} + \frac{1}{2} \left( \frac{1}{x^2} - 1 \right) \right]$$

Auxiliary variable:

$$m = h \sqrt{\bar{\eta}^2}$$

Again the subroutine $\text{erfs}(\ )$ is called and values of the error function are obtained for:

$$\frac{1}{m}, \frac{\tan \theta}{\sqrt{2} \sigma_w}, \left[ \frac{1 - \sin \theta}{\cos \theta \sigma_w} \right]$$

Once these values are obtained, the equation for the average number of refracting paths (fully arisen sea) can be calculated:

$$\bar{N} = \left\{ \frac{m^2}{4 \pi} \left[ e^{-1/m^2} + \frac{\sqrt{\pi}}{2m} \left\{ 1 + \text{erf} \left( \frac{1}{m} \right) \right\} \right]^2 \right\} \left[ \text{erf} \left( \frac{\tan \theta}{\sqrt{2} \sigma_w} \right) + \text{erf} \left( \frac{1 - \sin \theta}{\cos \theta \sigma_w} \right) \right]$$
For the rough surface case only the mode, the mean square pressure and the TL are modified from the smooth surface case; all other output is exactly the same. The new modified mean square pressure and TL are given in the equations below:

\[
\frac{P_{BBR}}{P_a^2} = \frac{8m^2 \cos \theta_o}{\sin \theta_o B(\theta_o) \left| \frac{\partial B(\theta_o)}{\partial \theta_o} \right|} \left[ \frac{1}{2} \left\{ 1 + \text{erf} \left( \frac{\tan \theta_o + \overline{\psi} \cos \phi_w}{\sqrt{2} \sigma_w} \right) \right\} \cos^2 \theta_o \left[ \sigma_w^2 + (\tan \theta_o + \overline{\psi} \cos \phi_w)^2 \right] \right] \cdot \left[ \frac{\cos \theta_o}{\sqrt{2\pi}} \left( \tan \theta_o + \overline{\psi} \cos \phi_w \right) e^{-\left( \tan \theta_o + \overline{\psi} \cos \phi_w \right)^2 \over 2 \sigma_w^2} \right] \cdot \left( \overline{\psi}(\theta_o) + RBL(\theta_o) \right) / 10 \]

\[
TL = 10 \log_{10} \left[ \frac{P_{BBR}}{P_a^2} \right]
\]

With the two new values the control is referred back to the main program where the output values are again printed of MODE, T, TA, TDIFF, R', R, DOP, D/E, PHI, PRATIO, and TL.

The main control program now automatically goes to the subroutine DP.F4 to calculate for direct path propagation. Within DP.F4 the following equations are solved.

**Depression angle**

\[
\sin \theta(t) = \frac{D}{\left[ (X(t) - X_a(t))^2 + y(t) - D^2 \right]^{1/2}}
\]
Azimuthal arrival angle

\[
\cos \phi(t) = \frac{\{X_a(t) - X_e(t)\}}{\sqrt{\left\{\left(X_a(t) - X_e(t)\right)^2 + y_a^2(t)\right\}^{1/2}}}
\]

Lateral range

\[
R(t) = \sqrt{\left\{\left(X_e(t) - X_a(t)\right)^2 + y_a^2(t)\right\}^{1/2}}
\]

Slant range

\[
R(t) = \sqrt{\left\{\left(X_e(t) - X_a(t)\right)^2 + y_a^2(t) + D^2\right\}^{1/2}}
\]

Time of closest approach

\[
\tau_o = \frac{(x_{1a} - x_{1e})(\omega_x - \omega_{ax}) - y_{ia}\omega_{ay}}{\left(\left(\omega_x - \omega_{ax}\right)^2 + \omega_{ay}^2\right)^{1/2}}
\]

Time difference between arrival time and arrival time of direct path from CPA point

\[
\Delta t = \tau - \tau_o + \frac{R(t) - R(t_o)}{c_2}
\]

Time of arrival

\[
\tau_A = \tau + \frac{h}{c_1} + \frac{R(t)}{c_2}
\]

Auxiliary variable:

\[
\cos \psi(t) = \frac{\omega_{ax}\left\{X_e(t) - X_a(t)\right\} - \omega_{ay}y_a(t)}{\omega_a R(t)}
\]
Doppler:

\[
\frac{S_B}{S_S} = \left[ 1 - \frac{2n_a}{c_2} \cos \psi(t) \right]^{1/2} \left[ 1 - \frac{2n_a}{c_2} \left\{ \frac{X_S(t) - X_a(t)}{R(t)} \right\} \right]^{1/2}
\]

Squared Pressure:

\[
\frac{P_{dp}(t)}{P_a^2} = \left( \frac{2mD}{\left[ \{X_S(t) - X_a(t)\}^2 + y_a^2(t) + D^2 \right]} \right)^2
\]

Transmission Loss:

\[
TL = 10 \log_{10} \left[ \frac{P_{dp}(t)}{P_a^2} \right] - AR(t)
\]

The control is now sent to the main program where the values outputted are printed MODE, T, TA, TDI, R', R, DOP, D/E, PHI, PRATIO, and TL. Now the final main loop is entered which is the calling of the subroutine DPR.F4 which does the calculations for the rough surface case of direct path propagation.

The initial rough surface formula for the statistics of the rough sea surface are the same; that is

\[
\xi, \sigma^2, \sigma_w, \bar{\psi}, \bar{\eta}^2, m, \bar{N}
\]

are the same exact equations only

\[
\cos \phi_w(t_a)
\]

is different as shown below

\[
\cos \phi_w(t_a) = \frac{n_{sx} \{X_S(t) - X_a(t)\} - n_{sy} y_a(t)}{\sqrt{\{X_S(t) - X_a(t)\}^2 + y_a^2(t)}}
\]
Therefore the modified mean square pressure and TL for the direct path rough surface case are:

\[
\frac{P_{DP}^2}{P_a^2} = \left\{ \frac{2m}{R(t)} \right\}^2 \left[ \frac{1}{2} \left\{ 1 + \text{erf} \left( \frac{\tan \theta + \psi \cos \phi_w}{\sqrt{2} \sigma_w} \right) \right\} \cos^2 \theta \left[ \sigma_w^2 + (\tan \theta + \psi \cos \phi_w)^2 \right] \\
+ \frac{\sigma_w}{\sqrt{2 \pi}} \cos^2 \theta \left( \tan \theta + \psi \cos \phi_w \right) e^{-\frac{(\tan \theta + \psi \cos \phi_w)^2}{2\sigma_w^2}} \right\} \cdot N \cdot 10
\]

\[
TL = 10 \log_{10} \left[ \frac{P_{DP}^2}{P_a^2} \right]
\]

A correction for air attenuation is then added to the TL.*

The final values are returned to the main program where they are printed out in the specified format, the values printed at MODE, T, TA, TDIF, R', R, DOP, D/E PHI, PRATIO, TL.

The main control program has now completed all of the necessary data reduction; it then asks the user if he would like to change anything and rerun the program; if not, the user can exit and end the calculations.

* See Eq. 64 of Appendix A
B.4 Output Description

The output of the programs differs slightly depending upon (1) whether a surface duct is present or not and (2) if the program is outputting single or double bounce results. Examples of the output for each of the four possible cases are given and described.
CASE 1, OUTPUT (A): NO SURFACE DUCT, DOUBLE BOUNCE OUTPUT

TRYING A DOUBLE BOUNCE SOLUTION

<table>
<thead>
<tr>
<th>Mode</th>
<th>T</th>
<th>TA</th>
<th>TDIF</th>
<th>K</th>
<th>R</th>
<th>DOP</th>
<th>DE</th>
<th>PHI</th>
<th>PRATIO</th>
<th>TL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-900</td>
<td>-832</td>
<td>-846</td>
<td>324478</td>
<td>324479</td>
<td>1.076</td>
<td>5.37176</td>
<td>8.353E-14</td>
<td>-130.8</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.525E+01</td>
<td>0.000E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>THEO= 0.537E+01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For each particular mode of propagation, two main lines of output are printed. Numbers 1-14 are the main output values. They are:

1. Mode of propagation
2. Running time parameter t
3. Time of arrival (secs) t_A
4. Difference in time between arrival time of mode t_A and time of arrival of direct path from CPA (secs)
5. Lateral range between source and receiver (ft) at time signal leaves source (t)
(6) Slant range between source and receiver (ft) at time signal leaves source \( t \)

(7) Doppler shift at arrival time \( t_A \)

(8) D/E arrival angle at receiver (deg) at \( t_A \)

(9) Azimuthal arrival angle (deg) at \( t_A \)

(10) Mean square pressure (includes bottom loss, water volumetric attenuation, no air losses) at \( t_A \)

(11) Transmission loss re 1 ft (dB) at \( t_A \)

(12) Lateral range (kiloyards), at time signal leaves source \( t \)

(13) Lateral range at time of arrival (kiloyards) at \( t_A \)

(14) Transmission loss re 1 yd (dB), includes air attenuation, at \( t_A \)

Also outputted are some supporting data used as checks, they are; (15-27):

(15) Grazing angle at bottom (deg)

(16) Bottom loss from LRAPP data for this grazing angle

(17) D/E angle at receiver

(18) \( \xi = \left( \frac{8 h \lambda_{max} g^2}{\pi^2 U^4} \right)^{\frac{1}{4}} \)

(19) \( \text{ERF}(\xi) \)

(20) \( \sigma^2 \) (mean square sea slope (fully arisen sea))

(21) Most probable slope estimate (radians)

(22) \( \bar{\eta}^2 = 2.3 \times 10^4 \frac{\sqrt{2}}{g U^3} \frac{4}{3} \left[ \sqrt{\pi} (1 - \text{erf}(\xi)) + \frac{e^{-\xi^2}}{\sqrt{2\pi}} \left\{ \frac{1}{2} \xi - 1 \right\} \right] \)

(23) \( \bar{\eta} = h \sqrt{\bar{\eta}^2} \)
First part of \( N \) (Average number of refracting paths)

\[
\frac{m^2}{4n} \left[ e^{-\frac{m^2}{2n}} + \frac{\sqrt{n}}{2m} \left( 1 + \text{erf} \left( \frac{m}{\sqrt{n}} \right) \right) \right]^2 2
\]
CASE 1, OUTPUT B: NO SURFACE DUCT — SINGLE BOUNCE OUTPUT

THERE IS A SINGLE BOUNCE

<table>
<thead>
<tr>
<th>BS</th>
<th>-600.</th>
<th>-555.</th>
<th>-569.</th>
<th>217057.217058.1.076</th>
<th>1.28174.</th>
<th>2.258E-14-136.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>72.352</td>
<td>66.967</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DSR</th>
<th>-600.</th>
<th>-555.</th>
<th>-569.</th>
<th>217057.217058.1.076</th>
<th>1.28174.</th>
<th>5.983E-14-132.2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>72.352</td>
<td>66.967</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DP</th>
<th>-600.</th>
<th>-547.</th>
<th>-561.</th>
<th>217057.217058.1.081</th>
<th>0.11174.</th>
<th>1.340E-17-168.7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>72.352</td>
<td>66.102</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DPR</th>
<th>-600.</th>
<th>-547.</th>
<th>-561.</th>
<th>217057.217058.1.081</th>
<th>0.11174.</th>
<th>1.005E-15-150.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>72.352</td>
<td>66.102</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For Case 1, Output B, the format is exactly as in Output A
CASE 1. NO SURFACE DUCT

INPUT USED IN EXAMPLE

ENTER THE APPROPRIATE PARAMETERS IN THE DIMENSIONS INDICATED.

VELOCITY VECTOR X-DIRECTION (KTS) = 220.0
VELOCITY VECTOR Y-DIRECTION (KTS) = 0.0
INITIAL X-COORDINATE OF AIRCRAFT (FT) = 0.0
INITIAL Y-COORDINATE OF AIRCRAFT (FT) = 24000.0
HEIGHT OF AIRCRAFT FROM SEA SURFACE (FT) = 10000.0
FREQ. OF AIRCRAFT RADIATION (HZ) = 150.0
VELOCITY VECTOR X-DIRECTION SUB (KTS) = 7.0
INITIAL X-COORDINATE OF SUB (FT) = 0.0
DEPTH OF SUB (FT) = 400.0
IS THE AIR TEMP. LESS THAN 50F? NO
WIND SPEED (KTS) = 8.8
X-DIRECTION COSINE = .707
Y-DIRECTION COSINE = .707
BOTTOM TYPE EITHER 3.0 OR 5.0 = 3.0
INITIAL TIME = -1000.0
FINAL TIME = 1000.0
TIME INCREMENTS = 100.0
IS THERE A SURFACE DUCT? NO
SURFACE SOUND SPEED (FT/SEC) = 5052.0
BOTTOM SOUND SPEED (FT/SEC) = 5053.0
DEPTH FOR BOTTOM SOUND SPEED (FT) = 15660.0
MIN. SOUND SPEED (FT/SEC) = 4875.0
DEPTH AT MIN. SOUND SPEED (FT) = 3440.0
For each particular mode of propagation, two main lines of output are printed. Numbers 1-14 are the main output values. They are:

(1) Mode of propagation
(2) Running time parameter $t$
(3) Time of arrival (secs) $t_A$
(4) Difference in time between arrival time of mode $t_A$ and time of arrival of direct path from CPA (secs)
(5) Lateral range between source and receiver (ft) at time signal leaves source ($t$)
(6) Slant range between source and receiver (ft) \( t_A \) at time signal leaves source \( t \)

(7) Doppler shift at arrival time \( t_A \)

(8) D/E arrival angle at receiver (deg) at \( t_A \)

(9) Azimuthal arrival angle (deg) at \( t_A \)

(10) Mean square pressure (includes bottom loss, water volumetric attenuation, no air losses) at \( t_A \)

(11) Transmission loss re 1 ft (dB) at \( t_A \)

(12) Lateral range (kiloyards), at time signal leaves source \( t \)

(13) Lateral range at time of arrival (kiloyards) at \( t_A \)

(14) Transmission loss re 1 yd (dB), includes air attenuation, at \( t_A \)

Also outputted are some supporting data used as checks, they are; (15-28):

(15) Grazing angle at bottom (deg)

(16) Bottom loss from LRAPP for this grazing angle

(17) D/E angle at receiver

(18) \[ \xi = \left( \frac{3h \lambda \alpha_0 g^2}{\pi^2 U^4} \right)^{1/3} \]

(19) \[ \text{ERF}(\xi) \]

(20) \( \sigma^2 \) (mean square sea slope (fully arisen sea))

(21) Most probable slope estimate (radians)

(22) \[ \bar{\eta}^2 = 2.3 \times 10^4 \frac{g^2 \sqrt{\pi}}{3U^3} \left[ \text{erf}(\eta) + \frac{e^{-\eta^2}}{2\eta} \right] \]

(23) \[ \eta = \eta \sqrt{\eta} \]

B-29
(24) First part of $N$ (average number of refracting paths)
\[ N = \frac{m^2}{4\pi} \left[ e^{-\frac{1}{2}m^2} + \sqrt{\pi} \left( 1 + \text{erf} \left( \frac{m}{\sqrt{2}} \right) \right) \right]^2 \]

(25) $\bar{N} \left[ \frac{1}{2} \left( 1 + \text{erf} \left( \frac{\tan \theta + \Psi \cos \phi_w}{\sqrt{2} \sigma_w} \right) \right) \cos \theta \sigma_w^2 + \left( \tan \theta + \Psi \cos \phi_w \right) \right] + \frac{\sigma_w \cos \theta}{\sqrt{2}} \left( \tan \theta + \Psi \cos \phi_w \right)
\exp \left[ - \left( \tan \theta + \Psi \cos \phi_w \right) / 2 \sigma_w^2 \right] \}

(26) Grazing angle to bottom

(27) Bottom loss from LRAPP

(28) Path length
For each particular mode of propagation, two main lines of output are printed. Numbers 1-1" are the main output values. They are:

(1) Mode of propagation

(2) Running time parameter \( t \)

(3) Time of arrival (secs) \( t_A \)

(4) Difference in time between arrival time of mode \( t_A \) and time of arrival of direct path from CPA (secs)

(5) Lateral range between source and receiver (ft) at time signal leaves source (\( t \))
(6) Slant range between source and receiver (ft) at time signal leaves source (t)
(7) Doppler shift at arrival time $t_A$
(8) D/E arrival angle at receiver (deg) at $t_A$
(9) Azimuthal arrival angle (deg) at $t_A$
(10) Mean square pressure (includes bottom loss, water volumetric attenuation, no air losses) at $t_A$
(11) Transmission loss re 1 ft (dB) at $t_A$
(12) Lateral range (kiloyards), at time signal leaves source (t)
(13) Lateral range at time of arrival (kiloyards) at $t_A$
(14) Transmission loss re 1 yd (dB), includes air attenuation, at $t_A$

Also outputted are some supporting data used as checks, they are:

(15) $B(\theta_0)$ - bounce distance
(16) $\partial B/\partial \theta_0$
(17) $G_1$
(18) $G_S$
(19) $G_B$
(20) $C_1$ - air sound speed
(21) $C_S$ - surface sound speed
(22) $C_{MAX}$ - max. sound speed
(23) $C_{MIN}$ - min. sound speed
(24) $C_B$ - bottom sound speed
(25) Grazing angle to bottom
(26) Bottom loss from LRAPP
(27) D/E angle at receiver

(28) $\xi$

(29) $\text{erf}(\xi)$

(30) Mean square sea slope (fully arisen sea)

(31) Most probable slope estimate (radians)

(32) $\overline{\eta}$

(33) $m$

(34) First part of $\overline{N}$ (average number of refracting paths)

\[
\overline{N} \left\{ \frac{1}{2} \left[ 1 + \text{erf} \left( \tan^{-1} \left( \frac{\tan \theta + \psi \cos \phi}{\xi_0} \right) \right) \right] \cos^2 \phi \left[ \cos^2 \left( \tan \theta + \psi \cos \phi \right) \right] + \frac{\xi_0}{\psi} \cos^2 \phi \left[ \tan \theta + \psi \cos \phi \right] \right] \\
\exp \left[ \frac{-\left( \tan \theta + \psi \cos \phi \right)^2}{2 \xi_0^2} \right]
\]

(36) Grazing angle to bottom

(37) Bottom loss from FACT

(38) Path length ($S(\phi)$)
CASE 2, WITH SURFACE DUCT

INPUT USED IN EXAMPLE

.LOAD WATER. REL, BO. REL, SB0. REL, BT3. REL, BT5. REL, ERFS. REL
LOADING

LOADER 11K CORE
13+3K MAX 84 WORDS FREE

EXIT

.ST

ENTER THE APPROPRIATE PARAMETERS IN
THE DIMENSIONS INDICATED.

VELOCITY VECTOR X-DIRECTION(KTS)= 220.0

VELOCITY VECTOR Y-DIRECTION(KTS)= 0.0

INITIAL X-COORDINATE OF AIRCRAFT(FT)= 0.0

INITIAL Y-COORDINATE OF AIRCRAFT(FT)= 12000.0

HEIGHT OF AIRCRAFT FROM SEA SURFACE(FT)= 10000.0

FREQ. OF AIRCRAFT RADIATION(HZ)= 150.0

VELOCITY VECTOR X-DIRECTION SUB(KTS)= 7.0

INITIAL X-COORDINATE OF SUB(FT)= 0.0

DEPTH OF SUB(FT)= 400.0

IS THE AIR TEMP. LESS THAN 50F? YES

WIND SPEED(KTS)= 8.8

X-DIRECTION COSINE= .707

Y-DIRECTION COSINE= .707

BOTTOM TYPE EITHER 3.0 OR 5.0= 3.0

INITIAL TIME= 600.0

FINAL TIME= 700.0
Case 2 continued.

TIME INCREMENTS  =  20.0
IS THERE A SURFACE DUCT  YES
SURFACE SOUND SPEED(FT/SEC) =  4990.4
BOTTOM SOUND SPEED(FT/SEC) =  5050.0
DEPTH FOR BOTTOM SOUND SPEED(FT) =  15540.0
MIN. SOUND SPEED(FT/SEC) =  4886.5
DEPTH AT MIN. SOUND SPEED(FT) =  3720.0
MAX. SOUND SPEED(FT/SEC) =  4998.5
DEPTH AT MAX. SOUND SPEED(FT) =  420.0
**TYPE WATER: F4**

```plaintext
COMMON /IN/ VAX, VAY, XIA, YIA, H, F, VS, XIS, D, AXN, BT, U, MSX, WSY
COMMON /OUT/ T, R, RP, TA, TDIF, TL, DOP, DT, PH, PPSO, A, MODE
COMMON /CC/ C1, C2, CS, CB, DB, DS, CMIN, CMAX, DMIN, ETA, GS, GB, RD, TD, PI,
1 G1
M = 0.0

**TYPE 1**

1 FORMAT ('H', 'ENTER THE APPROPRIATE PARAMETERS IN
1 THE DIMENSIONS INDICATED.')</n
CALL AIRV(VAX, VAY)
CALL AIRH(XIA, YIA)
CALL AIRV(H)
CALL AIRF(F)
CALL SUBV(VS)
CALL SUBC(XIS)
CALL SUBCD(D)

**TYPE 6666**

6666 FORMAT ('H', 'IS THE AIR TEMP. LESS THAN 50F?')
ACCEPT 6665, LANS

6665 FORMAT (A3)
IF (LANS.EQ. 'YES') GO TO 6667
AL = 0.000000074*(F+2)*H/(50.0*3.281)
GO TO 6664

6667 AL = 0.0125*F*H/1000.0
CONTINUE

CALL WIND(U, MSX, MSY)
CALL BOT(DT)
CALL TINT('!', TF, DT)

**TYPE 2**

2 FORMAT ('H', 'IS THERE A SURFACE DUCT?')
ACCEPT 6665, LANS

3 FORMAT (A3)
IF (LANS.EQ. 'YES') GO TO 4

CALL ENVS(CS)
CALL ENVC(CB, DB)
CALL ENVD(CMIN, DMIN)
GO TO 5

4 CALL ENVS(CS)
CALL ENVC(CB, DB)
CALL ENVD(CMIN, DMIN)
CALL ENVF(DMIN, DS)
CONTINUE

5 CONTINUE
30 CONTINUE

T = T1

911 CONTINUE

GS = (CS - CMIN)/DMIN
GB = (CB - CMIN)/(DB - DMIN)
C1 = 1100.0
AXN = 42
C2 = 5000.0
PI = 3.14159265
TX1 = (XIA - XIS)
TX2 = (VS - VAX)
TX3 = (YIA + VAY)
TX4 = (TX1 + TX2) - TX3
TX5 = TX2 + 2
TX6 = TX5 + (VAY + 2)
TD = TX4/TX6
RX1 = (VS*TD) + XIS
RX2 = (VAX*TD) + XIA
RX3 = (RX1 - RX2) + 2
RX4 = ((VAY*TD) + YIA) + 2
RD = SQRT(RX3 + RX4 + (D + 2))
```
THE0 = 0.0

AX1 = (F/1000.0)**2
AX2 = (0.1+AX1)/(1.0+AX1)
AX3 = (40.0+AX1)/(4100.0+AX1)
A = (AX2+AX3)/3000.0

RY1 = (VS+T)*XIS
RY2 = (VAX+T)*XIA
RY3 = (RY1+RY2)**2

RY4 = ((VAY+T)+YIA)**2
RP = SQRT(RY3+RY4)

R = SQRT(RY3+RY4+(D+2))
IF (LANS.EQ. 'YES') GO TO 3333
CALL BOUN(THE0)
GO TO 334

3333
G1 = (C-MAX-CMIN)/(DMIN-D)
GS = (C-MAX-CS)/D
CALL $BOUN(THE0)

334
TYPE 31 THE0

31
FORMAT (IH, 'THE0=', 'E10.3')

TYPE 703

703
FORMAT (IH, 'MODE T TA TDIF R'' R DOP D/E' 1 ' PHI P RATIO TL')

705
FORMAT (IH, 'TA', 'TA', 'TDIF', 'R''', 'R DOP D/E' 1 ' PHI P RATIO TL')

706
FORMAT (IH, 'R''', 'R DOP D/E' 1 ' PHI P RATIO TL')

4444
CALL BSR(THE0)

4446
CALL IDR(THE0)
TL1 = TL + (10.0 + ALDG10(9.0)) - AL
TYPE 706, RPK, RPT, TL1
GO TO 4445

4447
CALL BSSR(THE0)
TYPE 705, MODE, T, TA, TDIF, RP, R, DOP, DE, PHI, PRSD, TL
RPK = RP / 3000.0
R1 = (VS + TA) + XIS
R2 = (VAX + TA) + XIA
R3 = (RY1 - RY2) * 2
R4 = (VAY + TA) + YIA * 2
RPA = SQRT(RY3 + RY4)
RPT = RPA / 3000.0
TL1 = TL + (10.0 + ALDG10(9.0)) - AL
TYPE 706, RPK, RPT, TL1
GO TO 4445

4448
CALL BDSR(THE0)
TYPE 705, MODE, T, TA, TDIF, RP, R, DOP, DE, PHI, PRSD, TL
RPK = RP / 3000.0
R1 = (VS + TA) + XIS
R2 = (VAX + TA) + XIA
R3 = (RY1 - RY2) * 2
R4 = (VAY + TA) + YIA * 2
RPA = SQRT(RY3 + RY4)
RPT = RPA / 3000.0
TL1 = TL + (10.0 + ALDG10(9.0)) - AL
TYPE 706, RPK, RPT, TL1
CALL DPR(THE0)
TYPE 705, MODE, T, TA, TDIF, RP, R, DOP, DE, PHI, PRSD, TL
RPK = RP / 3000.0
R1 = (VS + TA) + XIS
R2 = (VAX + TA) + XIA
R3 = (RY1 - RY2) * 2
R4 = (VAY + TA) + YIA * 2
RPA = SQRT(RY3 + RY4)
RPT = RPA / 3000.0
TL1 = TL + (10.0 + ALDG10(9.0)) - AL
TYPE 706, RPK, RPT, TL1
T = T + DT
IF (T < TL, TF) GO TO 911
M = M + 1
IF (M GT, 1) GO TO 3000
TYPE 2000

2000 FORMAT(1H, 'TO CHANGE A RUN PARAMETER, ENTER THE'
1 'APPROPRIATE NUMBER:
1 1 AIRCRAFT VEL. VECTORS
1 2 AIRCRAFT INITIAL POSITION
1 3 AIRCRAFT HEIGHT

B-39
1  4  AIRCRAFT RADIATED FREQ.  
1  5  SUB VEL. VECTOR  
1  6  SUB INITIAL POSITION  
1  7  SUB DEPTH  
1  8  SURFACE SOUND SPEED  
1  9  BOTTOM SOUND SPEED AND DEPTH  
1 10  MINIMUM SOUND SPEED AND DEPTH  
1 11  WIND SPEED AND DIRECTION COSINES  
1 12  BOTTOM TYPE  
1 13  MAX. SOUND SPEED AND DEPTH  
1 14  TIME OF EVENTS  
1 15  RUN  
1 16  STOP  
1 ' ENTER THE APPROPRIATE NUMBER = ' $ 
ACCEPT 2001, K3

2001 FORMAT(I)
GO TO (47, 48, 49, 10, 11, 12, 13, 14, 15, 16, 50, 17, 19, 20, 30, 18) K3

3000 TYPE 3001

3001 FORMAT(IH, 'ENTER THE NUMBER OF THE PARAMETER' 
1  ' THAT YOU WISH TO CHANGE = ', $)
ACCEPT 3002, K4

3002 FORMAT(I)
GO TO (47, 48, 49, 10, 11, 12, 13, 14, 15, 16, 50, 17, 19, 20, 30, 18) K4

47 CALL AIRV(VAX, VAY)
GO TO 3000

48 CALL AIRC(XIA, YIA)
GO TO 3000

49 CALL AIRH(H)
GO TO 3000

10 CALL AIRF(F)
GO TO 3000

11 CALL SUBV(VS)
GO TO 3000

12 CALL SUBC(XIS)
GO TO 3000

13 CALL SUBD(D)
GO TO 3000

14 CALL ENVS(VS)
GO TO 3000

15 CALL ENVC(L, DB)
GO TO 3000

16 CALL ENVD(CMIN, DMIN)
GO TO 3000

50 CALL WIND(U, MSX, MSY)
GO TO 3000

17 CALL BOT(BT)
GO TO 3000

19 CALL ENVA(CMAX, DS)
GO TO 3000

20 CALL TIM(TI, TF, DT)
GO TO 3000

18 CALL EXIT
END

SUBROUTINE AIRV(VAX, VAY)
TYPE 200

200 FORMAT(IH, 'VELOCITY VECTOR X-DIRECTION(KTS) = ', 2X, $)
ACCEPT 201, VAXK

201 FORMAT(F)
VAX = 1.688*VAXK
TYPE 202

202 FORMAT(IH, 'VELOCITY VECTOR Y-DIRECTION(KTS) = ', 2X, $)
SUBROUTINE AIRC(XIA, VIA)
  TYPE 200
  FORMAT(1X, 'INITIAL X-COORDINATE OF AIRCRAFT (FT)=',2X,S1)
  ACCEPT 201, XIA
  TYPE 202
  FORMAT(1X, 'INITIAL Y-COORDINATE OF AIRCRAFT (FT)=',2X,S1)
  ACCEPT 201, VIA
  RETURN
END
SUBROUTINE AIRH(H)
  TYPE 200
  FORMAT(1X, 'HEIGHT OF AIRCRAFT FROM SEA SURFACE (FT)=',2X,S1)
  ACCEPT 201, H
  RETURN
END
SUBROUTINE AIRF(F)
  TYPE 200
  FORMAT(1X, 'FREQUENCY OF AIRCRAFT RADIATION (HZ)=',2X,S1)
  ACCEPT 201, F
  RETURN
END
SUBROUTINE SUBV(VSKT)
  TYPE 200
  FORMAT(1X, 'VELOCITY VECTOR X-DIRECTION KTS=',2X,S1)
  ACCEPT 201, VSKT
  RETURN
END
SUBROUTINE SUBC(XIS)
  TYPE 200
  FORMAT(1X, 'INITIAL X-COORDINATE OF SUB (FT)=',2X,S1)
  ACCEPT 201, XIS
  RETURN
END
SUBROUTINE SUBD(D)
  TYPE 200
  FORMAT(1X, 'DEPTH OF SUB (FT)=',2X,S1)
  ACCEPT 201, D
  RETURN
END
SUBROUTINE ENVS(CS)
  TYPE 200
  FORMAT(1X, 'SURFACE SOUND SPEED (FT/SEC)=',2X,S1)
  ACCEPT 201, CS
  RETURN
END
SUBROUTINE ENVC(DB)
  TYPE 200
  FORMAT(1X, 'BOTTOM SOUND SPEED (FT/SEC)=',2X,S1)
```
ACCEPT 201,CB
FORMAT(F)
TYPE 202
END
SUBROUTINE ENV1(CMIN,DMIN)
TYPE 200
FORMAT(1H,'MIN. SOUND SPEED(FT/SEC)='2X,$)
ACCEPT 201,DMIN
RETURN
END
SUBROUTINE ENV2(CMAX,DS)
TYPE 200
FORMAT(1H,'MAX. SOUND SPEED(FT/SEC)='2X,$)
ACCEPT 201,CMAX
RETURN
END
SUBROUTINE TIM(TI,TF,DT)
TYPE 200
FORMAT(1H,'INITIAL TIME='2X,$)
ACCEPT 201,TI
RETURN
END
SUBROUTINE BOT(BT)
TYPE 200
FORMAT(1H,'BOTTOM TYPE EITHER 3.0 OR 5.0='2X,$)
ACCEPT 201,BT
RETURN
END
SUBROUTINE WIND(U,MSX,WSY)
TYPE 200
FORMAT(1H,'WIND SPEED(KTS)='2X,$)
ACCEPT 201,U1
FORMAT(F)
U=U1*1.688
RETURN
END
```

SUBROUTINE DP (THEO)
COMMON /IN/ VAR, VAY, XIA, YIA, H, F, VS, XIS, D, RXN, BT, U, WSY, WSY
COMMON /OUT/ T, R, PR, TA, TDIF, TL, DOP, DE, PHI, FRSQ, H, MODE
COMMON /CC/ CI, C2, CS, CB, DB, DS, CHIN, CMAX,DMIN, ETA, 65, 66, RO, TD, FI,
1 GI
X1 = (VS + T) + XIS
X2 = (VAR + T) + XIA
X3 = (X1 + X2) + 2
X4 = (VAR + T) + YIA
X5 = X4 + 2
SIN = D / SORT (X3 + X5 + D) + 2
DE = ASIN (SIN)
DE = (DE + 180.0) / PI
MODE = "DP"
CD1 = (VAR + T) + XIA
CD2 = X1
CD3 = (CD1 - X1) / SORT (X3 + X5)
CD4 = ASIN (CD3)
PH1 = (CD4 + 180.0) / PI
TDIF = T - TD + (R - RO) / C2
TA = T + (H / CI) + (R / C2)
ZK = (VS + T) + XIS
ZK1 = (VAR + T) + XIA
ZK2 = (VAR + T) + YIA
ZK3 = (VAR + (2X - ZK1) - (VAR + ZK2)
ZK4 = SORT (VAR + ZK4 + VAY + 2)
ZK5 = ZK3 / ZK4 + R
D01 = (2.0 * ZK4 * ZK5) / C2
D02 = SORT (1.0 - D01)
D03 = 1.0 * D02
D05 = (2X - ZK1) / P
D06 = (2.0 * VS + D05) / C2
D07 = SORT (1.0 - D06)
DOP = (D03 * D07)
QZ1 = 4.0 * (AKH + 2) + (DI + 2) * D02
PRSQ = PR1 / (R + 2)
ETA = 1.0 * (10.0 + 20)
IF (PRSQ LT ETA) GO TO 999
GO TO 999
999
RETURN
END
SUBROUTINE DFR (THEO)
COMMON /IN/ VAR, VAY, XIA, YIA, H, F, VS, XIS, D, RXN, BT, U, WSY, WSY
COMMON /OUT/ T, R, PR, TA, TDIF, TL, DOP, DE, PHI, FRSQ, H, MODE
COMMON /CC/ CI, C2, CS, CB, DB, DS, CHIN, CMAX,DMIN, ETA, 65, 66, RO, TD, FI,
1 GI
MODE = "DFR"
CD1 = (VS + T) + XIS
CD2 = (VAR + T) + XIA
CD3 = (VAR + T) + YIA
CD4 = (WSY + CD1 - CD2) / (WSY + CD3) / PP
ZET = CI / F
ZET1 = (8.0 * H * ZET + (32, 174 + 2)) / ((PI + 2) + (U + 4))
ZET2 = SORT (ZET1)
ZET3 = SORT (ZET2)
TYPE 1111,ZET3
1111

FORMAT(1H,E10.4)
ZET4=U+30.4S
CALL ERFS,ZET3,ERF

TYPE 1111,EPF
SIG=(11500.0*ZET4)/(981.5*3)
SIG1=SORU(P1/2,0)
SIG2=(SIG+SIG1)*(1.0-ERF)
TYPE 1111,EPF
SIGW=SIG/SHRT(2.0)
CY=(2.86*ZET4)/1000.0
CY1=(CY+PI)/180.0
TYPE 1111,EPF
AT1=(23000.0*(SHRT(2.0)*4.0)/(081.5*(ZET4*3)+3.0)
AT2=(SHRT(P1)*1.0-ERF)
AT3=ZET3*2
AT4=EXP(-AT3)/ZET3
AT5=(1.0/(2.0*AT3))/-1.0
ATN=AT1*(AT2+(AT4+AT5))
TYPE 1111,ATN
HX=H*30.4S
EM=HX*SHRT(ATN)
TYPE 1111,EM
EM1=1.0/EM
CALL ERFS(EM1,ERF)
BN1=((1.0-ERF)*SHRT(P1))/(2.0*EM)
BN2=(1.0/(EM**2))
BN3=(EXP(-BN2)+BN1)**2*2.0
BN4=(EM**2)+BN3/(4.0*PI)
TYPE 1111,BN4
SXN=R
SXN1=ASIN(SXN)
SXN2=(180.0*SXN1)/PI
SXN3=SHRT(SXN2)/COSD(SXN3)
SXN4=SXN3/(SHRT(2.0)*SIMP)
CALL ERFS(SXN4,ERF)
SXN5=ERF
SXN6=(1.0-SIND(SXN2))/COSD(SXN2)
SXN7=SXN6/(SHRT(2.0)*SIMP)
CALL ERFS(SXN7,ERF)
BN5=SXN5*ERF
BN6=BN5*BN4
FD1=SXN4*(CY1*COS4)
FD2=FD1/(SHRT(2.0)*SIMP)
CALL ERFS(FD2,ERF)
FD3=(1.0+ERF)/2.0
FD4=((SIG**2)+FD1**2)+(COSD(SXN2))**2
FD5=(FD4+FD3)
FD6=(FD1+FD1)/(2.0*(SIG**2))
FD7=EXP(-FD6)
FD8=((COSD(SXN2))**2+FD7**2)*FD7**2)/SHRT(2.0+PI)
FD9=(FD5+FD8)*BN6
FD10=(A*R)/10.0
FD11=(10.0+FD10)
FD12=FD0+FD11
PO1=(2.0*A2)+R
PO50=(FD1**2)+FD12
ETA=1.0/(10.0**2)
IF(PO50.LT.ETA) GO TO 999

B-44
GO TO 998
998 TL=999.0
GO TO 100
100 RETURN
END
SUBROUTINE BSR(THE0)
COMMON /IN/ VAX, VAY, XIA, YIA, H, F, VS, XIS, D, AXN, BT, U, MSX, MSY
COMMON /OUT/ TR, RP, TA, TDIF, TL, DOP, DE, PHI, PSQ, A, MODE
COMMON /C/ C1, C2, CS, CB, DB, DS, CMIN, CMAX, DMIN, ETA, GS, GE, FD, TD, PI
1 G1
B1=<<CMIN/CS)>>2 * (COSD(THE0))**2
B2=<SORT(1.0-B1)> - SIND(THE0)
B3=<CB/CS>>2 * (COSD(THE0))**2
B4=<<SORT(1.0-B3)
B5=<SORT(1.0-B1)> - B4
B6<(B2/GS)> + B5/GB)
B7=(2.0-CB>>2) * (COSD(THE0)
T1=CS/CMIN+CORD(THE0)
T2=ALOG(T1+SORT(T1**2)-1.0))
T3=1.0/COSD(THE0)
T4=ALOG(T3+SORT(T3**2)-1.0))
T5=(T2-T4)**2.0/GS
T6<CS/CE+COSD(THE0)
T7=ALOG(T6+SORT(T6**2)-1.0))
T8=(T7-T2)**2.0/G6
T9=T5-T8
MODE='BSR'
P1=B7*<SIND(THE0)> / (COSD(THE0))
P2<CMIN/CS>>2 + SIND(THE0)
P3<CMIN/CS>>2 + (COSD(THE0))**2
P4=<<SORT(1.0-P3)
P5=<P2-P4>-1.0>/66
P6<CB/CS>>2
P7<P6*(COSD(THE0))**2
P8<<SORT(1.0-P7)
P9=P6/P8
P10=(<CMIN/CS>>2)/P4-P9
P11=(<SIND(THE0)> /P10)/GB
P12=(P5+P11)*2.0/C5+P1
COS1=(VS-T*T9)>>X1S
COS2=(VAY-T*(H-C1)>>2)*XIA
COS3=(VAY-T*(H-C1)>>2+Y1A
COS4=(MSX+CORD1-CORD2)- (MSY+CDSS)>>P7
ZET=C1/F
ZET1=(8.0+ZET**2 + (32.174**2))/<(F1**2)+ (U**4))
ZET2=<<SORT(ZET1)
ZET3<<SORT(ZET2)
TYPE 1111+ZET3
1111 FORMAT(1H,E10.4)
ZET4=U*30.48
CALL ERF3(ZET3,ERF)
TYPE 1111+ERF
SIG=(11500.0+ZET4) / (981.5**3)
SIG3=<<SORT(P1>2.0)
SIG2=(SIG+SIG1)**1.0-ERF
TYPE 1111+SIG2
SIG4=<<SORT(SIG2) / (SORT(2.0)
CY=(2.86+ZET4) / 1000.0
CY1=(CY+PI) / 180.0
TYPE 1111+CY
AT1=(23000.0*(SORT(2.0)+4.0)/(981.5*(ZET4+3)+3.0))
AT2=(SORT(PI)+(1.0-ERF))
AT3=ZET3+2
AT4=EXP(-AT3)/ZET3
AT5=(1.0-(2.0*AT3))-1.0
AT6=AT1*(AT2+AT4+AT5)
TYPE 1111=AT6
HX=(H+30.4E)
EM=HX*SORT(ATM)
TYPE 1111=EM
EM1=(1.0-EM)
CALL ERSF(EM1,ERF)
BM1=(1.0-EM)*SORT(PI)/(2.0*EM)
BM2=(1.0/(EM*2))
BM3=(EXP(-BM2)+BM1)*2+2.0
BM4=(EM*2)+BM3/(4.0+PI)
TYPE 1111=BN4
SNM3=SIND(TH0)/COSD(TH0)
SNM4=SNM3/(SORT(2.0)*SIGM)
CALL ERSF(SNM4,ERF)
SNM5=ERF
SNM6=(1.0-SIND(TH0))/COSD(TH0)
SNM7=SNM6/(SORT(2.0)*SIGM)
CALL ERSF(SNM7,ERF)
BN5=SNM5+ERF
BN6=BN5*BN4
FD1=SNM3+CY1*COS4
FD2=FO1/(SORT(2.0)*SIGM)
CALL ERSF(FD2,ERF)
FD3=(1.0-ERF)/2.0
FD4=(SIGM*2)+(FO1*2)*(COSD(TH0))*2
FD5=(FD4+FO3)
FD6=(FO1*2)/(2.0*(SIGM*2))
FD7=EXP(-FD6)
FD8=(COSD(TH0)*2)*FD7*SIGM/SORT(2.0+PI)
FD9=(FD4+FO3)*BN6
TB=CB*COSD(TH0)/CS
TB1=ACOS(TB)
TB2=(TB+180.0)/PI
IF(BT.EQ.3.0) GO TO 9998
IF(BT.EQ.5.0) GO TO 9999
TYPE 9996
FORMAT(1H 'DID NOT HAVE A VALUE FOR BT')
9998 CALL BT3(TB2,REL)
GO TO 9997
9999 CALL BT5(TB2,REL)
9997 TYPE 9995=TB2,RLA
9995 FORMAT(1H ,2E10.4)
S1=(2.0*CS)/COSD(TH0)
S2=(1.0*CS)+(1.0/GB)
S3=ASIN(COSD(TH0))/CS
S4=ASIN(COSD(TH0))/CS
S5=(PI+2.0)-(TH0+PI)/180.0)/6S
S6=(-S2+S3)+(S4/GB)+S5)*S
FD10=-(REL+ASIN)/10.0
FD11=(10.0*FD10)
FD12=FO4+FD11
FD13=(8.0*(ASIN+2))*COSD(TH0)/(COSD(TH0)*7.0*ERF(P12))
PRSO=FD1*FD12
TL=10.0*ALOG10(PRSO)
RETURN
END
SUBROUTINE BDR (THEO)

COMMON /IN/ VAX, VAY, VIA, VIA, H, F, VS, XIS, D, AXN, BT, U, MX, MSY
COMMON /OUT/ TR, FP, TH, TIF, TL, DOP, VE, PHI, FSM, R, NODE
COMMON /C/ C1, C2, CS, CI, DB, DS, CMIN, VMAX, DMIN, ETA, GS, GB, FD, TD, PI,
1 61
B1=((CMIN/CS)+2)*CosD(THEO)**2
B2=(SORT (1.0-B1))-SIND (THEO)
B3=((CS/CSI)**2)*CosD (THEO)**2
B4=SORT (1.0-B3)
B5=(SORT (1.0-B1)),-B4
B6=(B2/CSI)+(B5-B6)
B7=(2.0*CS-B6)*CosD (THEO)
T1=CS/CMIN*COSD (THEO))
T2=ALOS (T1+SORT (T1**2)-1.0)
T3=1.0/COSD (THEO)
T4=ALOS (T3+SORT (T3**2)-1.0)
T5=(T2-T4)**2.0/GB
T6=CSI/CSI+COSD (THEO))
T7=ALOS (T6+SORT (T6**2)-1.0)
T8=(T7-T2)**2.0/GB
T9=T5-T8
MODE='BDR'

P1=B7* (SIND (THEO))/COSD (THEO))
P2=((CMIN/CSI)**2)*SIND (THEO))
P3=((CMIN/CSI)**2)*SIND (THEO))**2
P4=SORT (1.0-P3)
P5=((P2/P4)-1.0)/6S
P6=(C5/CSI)**2
P7=(P2+COSD (THEO))**2
P8=SORT (1.0-P7)
P9=P6/P8
P11=((CMIN/CSI)**2)/P4-P9
P11=(SIND (THEO))*/P40/GB
P12=(P5+P11)**2.0/CSI+P1
COSD=VS+ (T+(2.0*TS))+XIS
COSD=VAY+(T-(H-H1)))+XIR
COSD=(4CSX+COSD2))/(WS*COSD3)/(2.0*B7)
ZET=CI+F
ZET= (8.0*H)+ZET+ (32.174**2)/(PI**2)*/(U**4)
ZET2=SORT (ZET1)
ZET3=SORT (ZET2)
ZET4=TYPE 1111*ZET3

TYPE 1111*ZET3

ZET4=U*30.48
CALL EPS (ZET3, EPF)

ZET4=U*30.48
CALL EPS (ZET3, EPF)

ZET4=U*30.48
CALL EPS (ZET3, EPF)

ZET4=U*30.48
CALL EPS (ZET3, EPF)

ZET4=U*30.48
CALL EPS (ZET3, EPF)

ZET4=U*30.48
CALL EPS (ZET3, EPF)
TYPE 1111.ATN
HN=H+30.48
EM=H*SORT(ATN)
TYPE 1111.EM
EM1=(1,0/EM)
CALL ERFS(EM1,EMF)
BN1=(1,0+EMF)*SORT(PI)/(2,0*EM)
BN2=(1,0/EMF)**2)
BN3=(EXP(-BN2)+BN1)**2)+2,0
BN4=(EMF**2)*BN3/(4,0+PI)
TYPE 1111.BN4
SXN3=SIND(THD)/COSD(THD)
SXN4=SXN3/(SORT(2,0)*SIGW)
CALL ERFS(SXN4,EMF)
SXN5=EMF
SXN6=(1,0-SIND(THD))/COSD(THD)
SXN7=SXN6/(SORT(2,0)*SIGW)
CALL ERFS(SXN7,EMF)
DN5=(SXN5+EF)
BN6=BN5+BN4
FO1=SXN3+(C1+COS4)
FO2=FO1/(SORT(2,0)+SIGW)
CALL ERFS(FO2,EMF)
FO3=(1,0+EMF)**2)
FO4=((SIGW**2)+(FO1**2)+COSD(THD)**2)
FO5=(FO4+FO3)
FO6=(FO1)**2)+(2,0)*(SIGW**2))
FO7=EXP(-FO6)
FO8=((COSD(THD)**2)+FO7+SIGM)/SORT(2,0+PI)
FO9=(FO5+FO8)*BNE
TR=(C1+CSTMN(THD))/C3
TBI=ACOS(TB)
TB2=TD1*180,0)*PI
IF<DT.E0,2,0)GO TO 9999
IF<DT.E0,5,0)GO TO 9999
TYPE 9996
9996 FORMAT (1H 'DID NOT HAVE A VALUE FOR BT')
9997 CALL BT3<TB2+PI)
GO TO 9997
9999 CALL BT5<TB2+PI)
9997 TYPE 9995<TR2+PI)
9995 FORMAT (1H 'BE10.4')
S1=(2.0*CD)/COSD(THD)
S2=(1.0-60)+(1,0/68)
S3=ASIN(CMIN+COSD(THD))/C3)
S4=ASIN(C1+COSD(THD))/C3)
S5=(PI/2,0)/(THE0+PI)/180,0))/61
S6=(<S2)+C14,6B)+55)**1
FO10=<2,0*PBL)+(2,0*A+36)+10,0
FO11=(10,0*FO10)
FO12=FO3*FO11
FO1=S(0,0+A/H)+62*COSD(THD))/4,0*SIND(THD)+BT+ARC(P12)
PR5=FO1*FO12
TL=10,0+ALO310+PPI)
RETURN
END
SUBROUTINE BCP(THD)
COMMON /TH*/IN,ICANG,TA,TV,TAIP,TVIP,VIC,T,THM,TAR,THM,THF
COMMON /OUT*/T,P,P,P,TADIF,TL,TOF,DEP,EPI,PI,P10,HI,MDF
COMMON /CM/CI,C1,C2,CC,CB,DD,DC,CMN,CMHS,CMIN,ETC,BC,GE,FO,TF,PI
1 61
B1=(2,0/CD)/COSD(THD)
B2=CMAN/CD)**2)+COSD(THD)**2)
B3=SOFI(1,0-B2)
B4 = (SIND(THEO) - .63) / GS
B5 = (COS(THEO) * .82) * COSD(THE0) * .82
B6 = SORT(1.0 - E5)
B7 = (B6 - B3) / 61
B8 = (COS(THE0) * .82) * COSD(THE0) * .82
B9 = SORT(1.0 - E8)
B10 = (B6 - B9) / GB
B11 = (B4 + B7 + B10) * F1
TH = ALG5 ((1.0 - COSD(THE0)) * SORT ((1.0 - COSD(THE0)) * .82) - 1.0)
TH1 = CS / (CNX * COID(THE0))
TH2 = ALG5 ((TH1 + 20) * (TH1 + 2) - 1.0)
TH3 = (TH - TH2) + 2.0 / GS
TH4 = CS / (CNX * COSD(THE0))
TH5 = ALG5 (TH4 + 20) * (TH4 + 2) - 1.0
TH6 = (TH - TH5) + 2.0 / GS
TH7 = CS / (CNX * COSD(THE0))
TH8 = ALG5 (TH7 + 20) * (TH7 + 2) - 1.0
TH9 = (TH5 + TH8) + 2.0 / GB
TH10 = TH3 + TH6 + TH9
MODE = 'SSR'
P1 = B11 * (SIND(THE0) * COSD(THE0))
P2 = (CNX * CS) * .82 * (SIND(THE0))
P3 = (CNX * CS) * .82 * (SIND(THE0)) * .82
P4 = SORT(1.0 - F3)
P5 = (1.0 - (P2 + P4)) / GS
P6 = (CNX * CS) * .82
P7 = SORT(1.0 - (P6 * (COSD(THE0) * .82)))
P8 = P6 / P7
P9 = (CNX * CS) * .82
P10 = P9 / P1
P11 = (PS - P14) * SIND(THE0) / GS
P12 = (CB - CS) * .82
P13 = SORT(1.0 - (P12 * (COSD(THE0) * .82))
P14 = P12 / P13
P16 = (PS - P14) * SIND(THE0) / GS
P15 = (P5 + P11 + P16) + 2.0 * CS + P1
COS1 = VS * (1 + TH10) + K1
COS2 = (VAR + (1 - CS)) + VIA
COS3 = (VAR + (1 - CS)) + VIA
COS4 = (COS1 - COS1) + 20)
ZET1 = C1 + F
ZET1 = (8.0 + ZET1) * (6.174 * 20) / (PI * 2 * 2 * 3.0)
ZET2 = SORT(ZET1)
ZET3 = SORT(ZET2)
TYPE 1111, ZET3
FDPMAT (1H + E10.4)
ZET4 = 0 + 30.48
CALL EPFS (ZET3, EFF)
TYPE 1111, EFF
SIG = (1500.0 * ZET4) / (981.5 * 3)
SIG1 = SORT(PI + 2.0)
SIG2 = SIG1 + SIG1 + (1.0 - EFF)
TYPE 1111, EFF
SIG4 = SORT(SIG2) + SORT(2.0)
CY = 2.86 + ZET4 + 1000.0
CY1 = CY + PI + 180.0
TYPE 1111, CY1
AT1 = (3000.0 + SORT(2.0 + 4.0) / (981.5 * ZET4 * 3.0)
AT2 = (SORT(PI) + (1.0 - EFF)
AT3 = ZET3 + 2
AT4 = EXP(-AT3) / ZET3
Bolt Beranek and Newman Inc.

ATS = (1.0 - (2.0*AT3)) - 1.0
ATH = AT1 + (AT2 + (AT4 + AT5))
TYPE 1111,ATN
HN = 30.48
EM = HN*SORT(ATH)
TYPE 1111,EN
EM1 = (1.0/EM)

CALL ERFS(EM1, EFF)
EM2 = (1.0 - (EM/2.0))
EM3 = (1.0 - (EM/2.0))
EM4 = (EM/2.0)
TYPE 1111,BHN
SXH3 = SIND(THE0) + COSD(THE0)
SXH4 = SXH3*SORT(2.0) + SIGM
CALL ERFS(SXH4,EFF)
SXN5 = EFF
SXN6 = (1.0 - SIND(THE0)) + COSD(THE0)
SXN7 = SXN6*SORT(2.0) + SIGM
CALL ERFS(SXN7,EFF)
BN5 = SXN5*EFF
BN6 = EM5*BN4
FD1 = SXN5 + (COS*EM5)
FD2 = FD1/SORT(2.0) + SIGM
CALL ERFS(FD2,EFF)
FD3 = (1.0 + EFF)/2.0
FD4 = ((SIGM**2) + (FD1**2)) + COSD(THE0)**2
FD5 = FD4/FD3
FD6 = (FD1**2) + (2.0*SIGM**2)
FD7 = EFF + FD6
FD8 = FD7/SORT(2.0) + SIGM
FD9 = FD5*FD8 + BN6
TYPE 3344,FD9

FORMAT(1H + E10.4)
TB = (CB*COSD(THE0))/CS
TB1 = ACOS(TB)
TB2 = TB1 + 180.0 + PI
IF (BT.EQ.3.0) GO TO 9999
IF (BT.EQ.5.0) GO TO 9999
3344

FORMAT(1H + (DID NOT HAVE A VALUE FOR BT) + E10.4)
9996

CALL BT*(TB1 + REB)
GO TO 9997
9999
CALL BT*(TB2 + REB)
9997

TYPE 9995, TB1 + REB
9995

FORMAT(1H + E10.4)
S1 = (2.0*CM) + COSD(THE0)
S2 = ACOS((CM + COSD(THE0))/CS)
S3 = (S1 + PI) - 180.0 - 10.0 - 63
S4 = ACOS((CM + COSD(THE0))/CS)
S5 = (S4 - 2.0)/61
S6 = ACOS((CM + COSD(THE0))/CS)
S7 = (S4 - 2.0)/65
S8 = (S4 + S7)/17.0 + 31
TYPE 6543, 18

FORMAT(1H + E10.4)
FD10 = -(REB + (CM*2)) - 10.0
FD11 = -10.0*FD10
FD12 = FD10*FD11
FD1 = (2.0*ACOS*CM + COSD(THE0) + SIND(THE0) + R11 + R02 + (P15))
PR20 = FD1*FD12
TL = 10.0*ALG10(REB)
RETURN
END
SUBROUTINE BDSR(THE0)
COMMON /IN/ VAX, VAY, XIA, YIA, H, F, VS, XIS, DAXH, BTU, MSX, MSY
COMMON /OUT/ T, R, FP, TH, TDF, TL, DOP, DFH1, FRS0, ANODE
COMMON /C/ C1, C2, CS, CB, DB, DS, CMIN, CMAX, DMIN, ETA, GS, GB, RO, TO, PI,
1 G1
B1=(2.0+CS)/COSD(THE0)
B2=((CMAX/CS)**2) *COSD(THE0)**2
B3=SOR.T(1.0-B2)
B4=(SIND(THE0)-B3)/6S
B5=((CMIN/CS)**2) *COSD(THE0)**2
B6=SOR.T(1.0-B5)
B7=(B6-B3)/61
B8=((CB/CS)**2) *COSD(THE0)**2
B9=SOR.T(1.0-B8)
B10=(B6-B9)/6B
B11=(B4+B7+B10+B1)
TH=ALOG((1.0/COSD(THE0)) +SOR.T(((1.0/COSD(THE0)**2)-1.0))
TH1=CS/(CMAX*COSD(THE0))
TH2=ALOG(TH+SOR.T((TH**2)-1.0))
TH3=((TH-TH2)**2)/6S
TH4=CS/(CMIN*COSD(THE0))
TH5=ALOG(TH4+SOR.T((TH4**2)-1.0))
TH6=((TH5-TH2)**2)/61
TH7=CS/(CB*COSD(THE0))
TH8=ALOG(TH7+SOR.T((TH7**2)-1.0))
TH9=((TH5-TH8)**2)/6B
TH10=TH3+TH6+TH9
MODE='BDSR/
P1=P11*SIND(THE0)/COSD(THE0))
P2=((CMAX/CS)**2) *SIND(THE0)
P3=((CMAX/CS)**2) *COSD(THE0)**2
P4=SOR.T(1.0-P3)
P5=(1.0-(P4/P4))/6S
P6=(CMIN/CS)**2
P7=SOR.T(1.0-(P6*(COSD(THE0)**2)))
P8=P6/P7
P9=((CMAX/CS)**2)
P10=P9-P4
P11=((P8-P10) *SIND(THE0))/61
P12=(CB/CS)**2
P13=SOR.T(1.0-(P12*(COSD(THE0)**2)))
P14=P12/P13
P15=((P8-P14) *SIND(THE0))/6B
P16=(P5+P11+P16)/6S+P1
COS1=(VS+T+(2.0+TH10)) *XIS
COS2=(VAX+T-(H-C1)) *XIA
COS3=(VAY+T-(H-C1)) *TH
COS4=(MSX*(COS1-COS2)-(MSY*COS3))/2.0+B11
ZET=C1/F
ZET1=(2.0+E2*ZET+(32.174**2)) /((PI**2)**0**4))
ZET2=SOR.T(ZET1)
ZET3=SOR.T(ZET2)
TYPE 1111*ZET3
1111 FORMAT(1H,+E10.4)
ZET4=U*30.48
CALL EF5S(ZET3,ERF)
TYPE 1111*EF5
SIG=(11500.0*ZET4)/(981.5**3)
SIG1=*(P1/2.0)
SIG2=(SIG*SIG1)*1.0-ERF
TYPE 1111*SIG2
SIG6=SOR.T(SIG62)/SOR.T(2.0)

B-51
Bolt Beranek and Newman Inc.

CY=(2.66*ZET4)/1000.0
CY1=(CY*PI)/180.0
TYPE 1111;CY1
AT1=(23000.0*SORT(2.0)*4.0)/(981.5*(ZET4**3)*3.0)
AT2=(SORT(P1))**(1.0-ERF)
AT3=ZET3*2
AT4=EXP(-AT3)-ZET3
AT5=(1.0/(2.0*AT3))-1.0
ATN=AT1*(AT2+(AT4+AT5))
TYPE 1111;ATN
HX=H*30.48
EM=HX*SORT(ATN)
TYPE 1111;EM
EM1=(1.0/EM)
CALL ERF$EM1,ERF)
EM2=(1.0/EM**2)
EM3=EXP(-EM1*EM2)*2.0
EM4=(EM**2)*EM3/(4.0*PI)
TYPE 1111;EM4
SN3=SIND(THE0)/COSD(THE0)
SN4=SN3/SORT(2.0)*SIGW)
CALL ERF$SN4,ERF)
SN5=ERF
SN6=SN5*EM)
SN7=SN6/SORT(2.0)*SIGW)
CALL ERF$SN7,ERF)
BN5=(SN5*ERF)
BN4=BN5*BN4
FD0=FO1/SORT(2.0)*SIGW)
FD1=FO1/SORT(2.0)*SIGW)
FD2=FD0/SORT(2.0)*SIGW)
FD3=FD0/SORT(2.0)*SIGW)
FD4=FD0/SORT(2.0)*SIGW)
FD5=FD0/SORT(2.0)*SIGW)
FD6=FD0/SORT(2.0)*SIGW)
FD7=FD0/SORT(2.0)*SIGW)
FD8=FD0/SORT(2.0)*SIGW)
FD9=FD0/SORT(2.0)*SIGW)
TYPE 3331;FD9
3331 FORMAT(IH,E16.4)
TB=(CE*COSD(THE0))/CS
TB1=ACOS(TB)
TB2=(TB1*180.0)/PI
IF(IET.EQ.3.0) GO TO 9998
IF(IET.EQ.5.0) GO TO 9999
TYPE 9996
9996 FORMAT(IH,DID NOT HAVE A VALUE FOR IT)
9998 CALL ET3(TBE+PBL)
GO TO 9997
9997 CALL ET5(TBE+PBL)
9995 TYPE 9995;TB2+PBL
9995 FORMAT(IH,2E16.4)
S1=(2.0*CS)/COSD(THE0)
S2=ACOS((CE+CS)/COSD(THE0))/CS)
S3=(THE0*PI)/180.0-S2/60
S4=ACOS((CE+MIN(COSD(THE0))/CS)
S5=(I4-32)/61
S6=ACOS((CE*COSD(THE0))/CS)
S7=(I4-32)/68
S8=(S3+S5+S7)*61

B-52
TYPE 3332
FORMAT (H, E10.4)
FD10 = -(2.0*PB+2.0*A*S)/10.0
FD11 = (10.0*FD10)
FD12 = FD9*FD11
PO1 = (8.0*(AXN+2)*COSD(THE))/((4.0*SIND(THE)*B11+ABS(P15))
PRSO = PO1*FD12
TL = 10.0*ALOG10(PRSO)
RETURN
END
SUBROUTINE BOUN(THED)

COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,BT
COMMON /OUT/ TR,FP,TA,TDF,TL,DOP,DE,PHI,PSO,A,MODE
COMMON /CC/ C1,C2,CS,CB,DB,DS,CMIN,CMAX,DMIN,ETA,GC,GB,RO,TO,PI,
1 G1
X1=(VAX-(T-(H/C1)))+XIA
X2=(VS+T)+XIS
X3=(VAY-(T-(H/C1)))+YIA
X4=(X1-X2)**2+X3**2
X5=SQRT(X4)
R1=(CMIN-CS)**2
R2=SQRT(1.0-P1)
R3=SQRT((1.0-(CS/CS)**2))
R4=(R2-R3)/GS
R5=R2/GS
R6=(R4+R5)*2.0*CS
IF(X5.GE.R6) GO TO 100

TYPE 1

1 FORMAT('THERE IS A SINGLE BOUNCE')
THETAS=-AC(S/CSC/CS)
THETE=(C180.0*THETAS)/3.14159265
ZN=0.01
11 DO 10 I=1,9000,1
Z2=FLOAT(I)
Z=THETE+(Z2-1.0)/100.0+.01
IF(Z.GT.90.0) GO TO 400
TH=CS/(CMIN*COSD(Z))
TH1=HLOG(TH,THETE-1.0)
TH2=1.0/COSD(Z)
TH3=HLOG(TH2+SORT((TH2**2)-1.0))
TH4=CS/(CS*COSD(Z))
TH5=F*DS/(TH4+SORT((TH4**2)-1.0))
TH6=1.0*(TH1-TH3)/GS
TH7=(2.0*TH5-TH1)/GB
TH8=TH6-TH7
X1=(VAX-(T-(H/C1)))+XIA
X2=(VS+T+TH8)+XIS
X3=(VAY-(T-(H/C1)))+YIA
X4=(X1-X2)**2+X3**2
X5=SQRT(X4)
R1=(CMIN-CS)**2*(COSD(Z)**2)
R2=SQRT(1.0-P1)
R3=(R2-R3)/GS
R4=(R4+R5)*2.0*CS
R5=SQRT((R3-R5))
R6=(R3+R5)*2.0*CS
R7=(R3+R5)*2.0*CS
R8=(R3+R5)*2.0*CS
IF(R8.LE.ZN) GO TO 41

CONTINUE
ZN=ZN+.01
0=.1
IF(ZN.LE.0) GO TO 11

TYPE 40

40 FORMAT('NO SOLUTION TO SINGLE BOUNCE')
GO TO 100

41 THEE=2
CALL BS(THEE)
GO TO 42
TYPE 43

FORMAT ('TRACING A DOUBLE BOUNCE SOLUTION')
THE1 = (180.0 + THE9) / 3.14159265
Z = 0.1

DO 99 I = 1, 9000, 1
Z2 = FLOOR (Z)
Z1 = THE1! ((Z2 - 1.0) / 100.0) + 0.1
IF (Z1 > 90.0) GO TO 98
THE8 = CS / (CMIN + COSD (Z1))
THE8 = 1.0 / COSD (Z1)
THE3 = ALDG (THE8 + SORT ((THE8**2) - 1.0))
THE4 = CS / (CB + COSD (Z1))
THE5 = ALDG (THE4 + SORT ((THE4**2) - 1.0))
THE6 = (2.0** (THE3 - THE8)) / CS
THE7 = (2.0** (THE5 - THE8)) / CS
THE8 = THE6 - THE7

XB1 = (VAX** (T - (H/C1))) + XIA
XB2 = (VNS** (T + (2.0** THE8))) + XIS
XB3 = (VAX** (T - (H/C1))) + YIA
XB4 = (XB1 - XB2**2) + XB3**2
XB5 = SORT (XB4)
BB1 = (CMIN + CS**2) / (COSD (Z1)**2)
BB2 = SORT (1.0 - BB1)
BB3 = (BB2 - SIN (Z1)) / CS
BB4 = (CB + CS**2) / (COSD (Z1)**2)
BB5 = SORT (1.0 - BB4)
BB6 = (BB2 - BB3) / GS
BB7 = ((BB3 + BB6) * 4.0** CS) / (COSD (Z1))
BB8 = (BB7 - BB6) / BB7
BB9 = ABS (BB9)
IF (BB9 .LE. ZN1) GO TO 97

CONTINUE
ZN1 = ZN1 + 0.1
Q1 = .1
IF (ZN1 .LE. 0.1) GO TO 12

TYPE 98

FORMAT ('NO SOLUTION TO DOUBLE BOUNCE')
GO TO 42

THE0 = Z1
CALL BE1 (THE0)
RETURN

END
CALL BT5(TE2,RBL)
TYPE 9995,TE2,RBL
FORMAT(1H :2E10.3)
S1=(2.0*CS)/COSD(THE0)
S2=(1.0/6S)+(1.0/6G)
S3=ASIN((CMIN*COSD(THE0))/CS)
S4=ASIN((CB*COSD(THE0))/CS)
S5=(PI/2.0)-(THE0+PI)/180.0)/6S
S6=(-S2*S3)+(S4/6G)+S5+S1
Q21=-((A+S6)+RBL)/10.0
Q22=(-(10.0+Q21))
PO1=(S.0+AXN*2)*SIND(THE0)+COSD(THE0)+Q22
PRSO=PO1/(B7*ABS(P12))
TL=(10.0*ALOG10(PR5O))
RETURN
END
SUBROUTINE BT(THE0)
COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,D,AXN,BT
COMMON /GUT/ T,TR,TP,TA,TDIF,TL,DP,DE,FH1,PRSO,A,MODE
COMMON /C1/ C1,C2,CS,CE,DP,DS,CMIN,CMAX,DMIN,ETA,GS,GB,RF,TO,PI,
*                                      1G1
B1=(CMIN/CS)**2+1(COSD(THE0))**2
B2=(SORT(1.0-B1))=SIND(THE0)
B3=(CB/CS)**2+1(COSD(THE0))**2
B4=1.0-SORT(1.0-B3)
B5=SOR(1.0-B1)-B4
B6=(B2/6S)+1(B5/6G)
B7=(2.0*CS*6G)/COSD(THE0)
T1=CS/(CMIN*COSD(THE0))
T2=ALOG(T1+SORT(T1**2)-1.0)
T3=1.0*COSD(THE0)
T4=ALOG(T3+SORT(T3**2)-1.0)
T5=(T2-T4)**2+1/6S
T6=CS/(C1*COSD(THE0))
T7=ALOG(T6+SORT(T6**2)-1.0)
T8=(T7-T2)**2+1/6G
T9=T5-T8
DE=THE0
TA=1+(2.0*9)
TDIF=T+2.0*9)-180*H/C1-1)(RD/C2)
F1=B7*(SIND(THE0))/COSD(THE0))
P0=(CMIN/CS)**2+1(SIND(THE0))
P3=(CMIN/CS)**2+2(COSD(THE0))**2
P4=1.0+SORT(1.0-P3)
P5=(P2/P4)**1.0/6S
P6=(CB/CS)**2
P7=P6+1(COSD(THE0))**2
P8=SOR(1.0-P7)
P9=P8/P9
P10=(CMIN/CS)**2-P4-P9
P11=(SIND(THE0))*P10)/6S
P12=(P5+P11)**2.0*CS)+P1
X1=(VAX+1(T-H/C1)+XIA
X2=(VS+(T-2.0*9)+XIS
X3=(X1-X2)**2
X4=(VAY+(T-H/C1))+YIA
X5=(X1-X2)/SORT(X3+(X4**2))
X6=ACOS(0.5)
PHI=(180.0+X6)/PI

B-57
CO1=VS*(T+(2.0*T9))/XIS
CO2=VAX*(T-(H-C1))/XIA
CO3=VAY*(CO1-CO2)
CO4=-(VAY*T-(H-C1))+YIA
CO5=COE+(VAY-CO4)
CO6=SOFT(VAY**2+VAY**2)
CO7=ABS(CO6)
CO8=CO2*(CO4**2)
CO9=CO8*(CO4**2)+(4.0*DB)-D**2
CO10=CD7*SOFT(CO9)
CD11=CO5-CO10
DO1=(CO6**2)/(C2**2)
DO2=(2.0*CO7)+CO11/C2
DO3=1.0+DO1-DO2
DO4=SOFT(DO3)
DO5=1.0/DO4
DO6=(SORT(CO8)+2.0*VS)/C2
DO7=1.0+DO5-DO6/SORT(CO3))
DO8=DO3*SOFT(DO7)
TB=CB*COSD(THEO))/CS
TB1=ACD*(TD)
TB2=(TB1+180.0)/PI
IF(BT.EQ.0.0) GO TO 9998
IF(BT.EQ.5.0) GO TO 9999
TYPE 9996
9996 FORMAT(1H + DID NOT HAVE A VALUE FOR BT(0))
9998 CALL BTK(TBE,REL)
9999 GO TO 9907
9997 TYPE 9995;TE1,REL
995 FORMAT(1H + ZE10.3)
S1=(2.0*CS)/COSD(THE0)
S2=(1.0/GS)+(1.0/GB)
S3=ASIN(COSD(THE0))/CS
S4=ASIN(COSD(THE0))/CS
S5=(PI/2.0)-((THE0+PI)-180.0))/GS
S6=-(S2*32)+(S4/GS)+S5*81
O211=-(COSM**2)+(2.0*REL))/10.0
O222=(10.0*O221)
MODE='E0'
PO1=8.0*(A/N**2)*SIND(THE0)*COSD(THE0)*O222
PNO1=PO1*(8.0*REL)/(1.0*REL)
POL1=10.0*ALDS10*(S61)
RETURN
END

B-58
SUBROUTINE S88HUM(THEO)
COMMON /IN/ VAX,VA,PIA,YIA,H,F,VS,XIS,B,AAXN,BT
COMMON /OUT/ TR,RP,TA,TDIF,TL,DOP,DEPHI,PSQ,A,MODE
COMMON /C/ C1,C2,C3,CB,DB,DS,CMIN,C1M4X,DMIN,ETA,GB,RO,TO,PI,
     1 G1
X1=(VAX+(T-(H/C1)))/XIA
X2=(VS+T)/XIS
X3=(VAY+(T-(H/C1)))/YIA
X4=(X1+X2)/X32
X5=SQRT(X4)
R1=SQRT((1.0-(COS(C1)**2)))
R2=SQRT((1.0-(COS(CB)**2)))
R3=(R1-R2)/GB
R4=SQRT((1.0-(COS(CE)**2)))
R5=(R4-R2)/GB
R7=(R3+R5+R6)*2.0*CE

IF(X5.GE.R7) GO TO 100

TYPE 1

FORMAT(1H,'SURFACE DUCT WITH SINGLE BOUNCE SOLUTION'/)
THEB=ACOS(CS/CB)
THE=(180.0+THEB)/PI
ZB=.01

11
DC 10 I=1,9000+1
Z2=FLOAT(I)
Z=THE+(Z2-1.0)/100.0+.01
IF(Z.GT.90.0) GO TO 400
TH=ALOG((1.0-COS(Z)))+SORT(((1.0-COS(Z))**2)-1.0))
TH1=CS/(CMIN+CS)
TH2=ALOG(TH1)+SORT((TH1**2)-1.0))
TH3=(TH/TH2)*2.0/GB
TH4=CS/(CM1+CS)
TH5=ALOG(TH4)+SORT((TH4**2)-1.0))
TH6=((TH5-TH2)*2.0)/61
TH7=CS/(CB+CS)
TH8=ALOG(TH7)+SORT((TH7**2)-1.0))
TH9=(TH5-TH8)*2.0/GB
TH10=TH3+TH6+TH9
X1=(VAX+(T-(H/C1)))/XIA
X2=(VS+(T+TH10))/XIS
X3=(VAY+(T-(H/C1)))/YIA
X4=(X1+X2)/X32
X5=SQRT(X4)
B1=(2.0*CS)/COS(Z)
B2=((COS(Z)**2)+(COS(Z)))*GB
B3=SQRT((1.0-B2)
B4=CS/D2(B3)/GB
B5=(COS(Z)**2)+(COS(Z))**2
B6=SQRT((1.0-B5)
B7=(B6-B3)/GB
B8=(CB/CS)**2)+(COS(Z))**2
B9=SQRT((1.0-B8)
B10=(B2-B9)/GB
B11=(B4+B7+B10)+B1
B12=(B11-X5)+B1
B13=ABS(B12)

IF(B13.LE.ZN) GO TO 41
CONTINUE
ZB=ZN+.01
Q=1

B-60
IF(ZN.LE.0) GO TO 11
400 TYPE 40
10 FORMAT(1H1, 'NO SOLUTION OF SURFACE DUCT WITH SINGLE BOUNCE'/)
GO TO 100
41 THEO=2
CALL BSS(THEO)
GO TO 42
100 TYPE 43
43 FORMAT(1H1, 'TRYING A SURFACE DUCT WITH DOUBLE BOUNCE SOLUTION'/)
THE9=ACOS(CS/CE)
THE1=(180.0+THE9)/PI
ZTH=0.01
12 DO 99 I=1,9900+1
23=FLOAT(I)
21=THE1+((Z2-1.0)/100.0)+.01
IF(Z2.GT.90.0) GO TO 988
THB=ALOG((1.0*COSD(Z2))+SORT(((1.0*COSD(Z2))**2)-1.0))
THB1=CS/COSD(Z2)
THB2=ALOG((THEB+SORT((THEB**2)-1.0))
THB3=(THEB-COSD(Z2))
THB4=CS/COSD(Z2)
THB5=ALOG((THB4+SORT((THB4**2)-1.0))
THB6=(THEB5-COSD(Z2))
THB7=(THEB5-COSD(Z2))
THB8=ALOG((THB7+SORT((THEB7**2)-1.0))
THB9=(THEB5-COSD(Z2))
THB10=THEB3+THEB6+THEB9
XB1=(VAY* (1-0+(COSD(Z1))))*XIA
XB2=(VY* (1+0+(COSD(Z1))))*XIA
XB3=(VAY* (1-0+(COSD(Z1))))*XIA
XB4=(THEB1-XB2)**2+XB3**2
XB5= SORT (XB4)
BB1=(2.0+CS)/COSD(Z2)
BB2=(COSD(Z2))**2+CS/COSD(Z2)**2
BB3= SORT (1.0-BB2)
BB4= (SIND(Z1)-BB3)/60
BB5=(COSD(Z1))**2+CS/COSD(Z2)**2
BB6= SORT (1.0-BB5)
BB7=(BB6-BB3)/60
BB8=(CS)+(CS)(**2)+COSD(Z1)**2
BB9= SORT (1.0-BB8)
BB10=(BB6-BB9)/60
BB11=(BB4+BB7+BB10)+BB1+2.0
BB12=(BB11-BB5)+BB11
BB13=ABS(BB12)
IF(BB13.LE.2M1) GO TO 97
99 CONTINUE
2M1=2M1+.01
01=.1
IF(Z2M1.LE.0) GO TO 12
998 TYPE 98
98 FORMAT(1H1, 'NO SOLUTION OF SURFACE DUCT WITH DOUBLE BOUNCE'/)
GO TO 42
97 THEO=21
CALL BSS(THEO)
42 RETURN
END
SUBROUTINE ISSOTHit)
    COMMON /IM''V«X» VftV» XIRJ Vlfl.H« F. Vfl. XIS« I=IXN. it
    COMMON 'OUT ' T» P. PP» TR« TDIF« TL» tnF •
    COMMON ' CC'' C£» C£» CSfCt« I'E- Di»
    B1=(2.0+CS) /COSD(THE0)
    B2=((CMAX-CS)++2) *COSD(THE0))++2
    B3=SOFT(1.0-B2)
    B4=(SIND(THE0)-B3)/65
    B5=((CMIN-CS)++2) *COSD(THE0))++2
    B6=SOFT(1.0-B5)
    B7=(B6-B3)/61
    B8=((CB/CS)++2) *COSD(THE0))++2
    B9=SOFT(1.0-B8)
    B10=(B6-B9)/61
    B11=(B4+B7+B10)*61
    TH=ALDG((1.0/COSD(THE0)+SOFT((1.0/COSD(THE0))++2)-1.0))
    TH1=CS/(CMAX/COSD(THE0))
    TH2=ALDG((TH1/SIND(THE0))++2)-1.0))
    TH3=(TH-TH2) *2.0/G6
    TH4=CS/(CMIN/COSD(THE0))
    TH5=ALDG((TH4/SORT((TH4++2)-1.0))
    TH6=(TH5-TH2) *2.0/G1
    TH7=CS/(CB/COSD(THE0))
    TH8=ALDG((TH7/SORT((TH7++2)-1.0))
    TH9=(TH5-TH8) *2.0/GB
    TH10=TH3+TH4+TH9
    DE=THE0
    TA=T+TH10
    P1=B11*(SIND(THE0)/COSD(THE0))
    P2=((CMAX-CS)++2) *SIND(THE0))
    P3=((CMAX-CS)++2) *COSD(THE0))++2
    P4=SOFT(1.0-P2)
    P5=(1.0-P2-P4)/65
    P6=(CMIN/CS)++2
    P7=SOFT(1.0-(P6+COSD(THE0))++2))
    P8=P6/P7
    P9=(CMAX/CS)++2
    P10=P9/P4
    P11=((P8-P10) *SIND(THE0)) /G1
    P12=(CB/CS)++2
    P13=SOFT(1.0-(P12+COSD(THE0))++2))
    P14=P12/P13
    P16=((P8-P14) *SIND(THE0)) /GB
    P15=((P5+P11+P16) *2.0/CS)+P1
    TYPE 3334+B11+F15
    3334 FORMAT(1H-2E10.4)
    TYPE 3335+G1+G5+GB
    3335 FORMAT(1H+3F10.3)
    TYPE 3336+C1+CS+CMAX+CMIN+CB
    3336 FORMAT(1H+5F10.3)
    X1=((VAX*(T-(H-C1)))++XIA
    X2=(VS*(T+TH10))++XIS
    X3=(X1-X2)++2
    X4=((VAY*(T-(H-C1)))++YIA
    X5=(X1-X2)*SORT(X3=(X4++2))
    X6=ACD2(C5)
    PH1=480.0*X6/P1
    CD1=(V5*(T+TH10))++XIS
    CD2=(VAX*(T-(H-C1)))++XIA

B-62
CO3=VAY*(CO1-CO2)
CO4=(VAY+(H-C1)) YIA
CO5=CO3*VAY*CO4
CO6=SOFT(VAY+2+VAY+2)
CO7=ABS(CO6)
CO8=(CO1-CO2)*CO2
CO9=CO8*2+(2.0*DB)-D**2
CO10=CO7*SOFT(CO9)
CO11=CO5/CO10
DD1=(CO5-CO1)/C2
DD2=(C20+DD)/C11/C2
DD3=1.0+DD1-DD2
DD4=SOFT(DD3)
DD5=DD3/DD4
DD6=SOFT(DD5+2.0+V5)/C2
DD7=1.0+DD5-DD6/DD9/DD2
DD8=DD1*SOFT(DD7)
MODE=110
TDIF=1+10-TO-(PD-C2)-(H-C1)
TB=CB+CO3*THE0/C2
TB1=ACOS(1)
TB2=(TB1+180.0)/PI
IF (ET.EQ.3.0) GO TO 9996
IF (ET.EQ.5.0) GO TO 9999
TYPE 9996
FORMAT(1H ' DID NOT HAVE A VALUE FOR RT/)
9996 CALL BT3(TB+PBL)
GO TO 9997
9997 CALL BT5(TO+PBL)
9999 TYPE 9995
9995 FORMAT(1H '2E10.3)
S1=(2.0+C2)/COS(THE0)
S2=ACOS((S1)/COS(THE0))/CS
S3=-(THE0+PI)/180.0+S2)/CS
S4=ACOS((S1+S2)/CS)
S5=(34-22)/61
S6=ACOS((S4+CS)/THE0)/CS)
S7=(S4-22)/CB
S8=(S4+57)+S1
Q21=<(S8+3 Diss+PBL)-10.0
Q22=<(10.0+Q21)
PO1=8.0+<<<IND(THE0)+COS(THE0)*Q22
PRSS=PO1/(PI+1+HED(P15))
TL=(10.0+HLD910+PRSS)
RETURN
END
SUBROUTINE BDS(THE0)
COMMON /IN/ VAY+VAY+YIA+YIA+H+F+VX+XIS+D+H/M/F
COMMON /OUT/ T+FFP+TA+TDIF+TL+DOP+DE+PHI+PRSS+AN+MODE
COMMON /CC/ C1+C2+CS+CB+DB+DC+CMIN+CMAX+DMIN+ETA+G3+GB+PD+TO+PI+1 61
B1=(3.0+CO)/COS(THE0)
B2=(CMAX+CO)**2*COS(THE0)**2
B3=SOFT(1.0+B2)
B4=(SIND-THE0-B3)**G3
B5=(CMIN+CO)**2*COS(THE0)**2
B6=SOFT(1.0+B5)
B7=(B6-B3)/G3
B8=(CB-C)**2+COS(THE0)**2
B9=SOFT(1.0+B8)
B10=(86-89)/GB
B11=(84+87+B10)*B1
TH=ALOG((1,0/COSD(TH0))+SORT((1,0/COSD(TH0))*2)-1,0)
TH1=CS/(CMAX*COSD(TH0))
TH2=ALOG((TH1+SORT((TH1)*2)-1,0))
TH3=((TH-TH2)*2,0)/GS
TH4=CS/(CMIN*COSD(TH0))
TH5=ALOG((TH4+SORT((TH4)*2)-1,0))
TH6=((TH5-TH2)*2,0)/G1
TH7=CS/(CB*COSD(TH0))
TH8=ALOG((TH7+SORT((TH7)*2)-1,0))
TH9=((TH5-TH8)*2,0)/GB
TH10=TH3+TH6+TH9
DE=THE0
TA=T+(2,0+TH10)
P1=B11*(SIND(TH0)/COSD(TH0))
P2=((CMAX/CS)*2+SIND(TH0))
P3=((CMAX/CS)*2+COSD(TH0))*2
P4=SORT(1,0-P3)
P5=(1,0-(P2/P4))/GS
P6=(CMAX/CS)*2
P7=SORT(1,0-(P6+(COSD(TH0)*2)))
P8=P6/P7
P9=(CMAX/CS)*2
P10=P9/P4
P11=(P8-P10)*SIND(TH0)/G1
P12=(CB/CS)*2
P13=SORT(1,0-(P12+(COSD(TH0)*2)))
P14=P12/P13
P15=(P5+P11+P16)*2,0/CS-P1
X1=(VAX*(T-(H/C1)))+XIA
X2=(V5*(T+(2,0+TH10)))+XIA
X3=(X1-X2)*2
X4=(VAY*(T-(H/C1)))+YIA
X5=(X1-X2)/SORT(X3+(X4*2))*2
X6=RADS(5)
P11=(130,0*X6)/PI
CD1=(V5*(T+(2,0+TH10)))+XIA
CD2=(VAX*(T-(H/C1)))+XIA
CD3=VAX*(CD1-CD2)
CD4=(VAY*(T-(H/C1)))+YIA
CD5=CD3+(VAY*CD4)
CD6=SORT((VAX*2)+VAY*2)
CD7=ABS(CD6)
CD8=(CD1-CD2)*2
CD9=(CD8*2)+(4,0*DB)-D*2
CD10=CD7*SORT(CD9)
CD11=CD5+CD10
DO1=(CD6*2)+(C2*2)
DO2=((2,0+CD7)+(CD11)/C2
DO3=1,0+DO1-DO2
DO4=SORT(DO3)
DO5=1,0+DO4
DO6=((SORT+DO3)*2)+2,0+V1/C2
DO7=1,0+DO5-DO6/SORT(CD9)
DO8=(DO1+DO7)*DO7
TE=CD*CD1*(THE0)/CS
TB1=RADS(TI)
TB2=(TB1+130,0)/P1
IF (BT.EQ.3.0) GO TO 9998
IF (BT.EQ.5.0) GO TO 9999
TYPE 9996
9996 FORMAT (1H 'DID NOT HAVE A VALUE FOR BT')
9998 CALL BT3(TB2,PBL)
GO TO 9997
9999 CALL BT5(TB2,PBL)
9997 TYPE 9995,TB2,PBL
9995 FORMAT (1H 'E10.3')
S1 = (2.0*CD)/COS(DHEO)
S2 = ACOS((CMAN-COS(DHEO))/CS)
S3 = ((DHEO+F1)/180.0)-S2/GS
S4 = ACOS((CMIN-COS(DHEO))/CS)
S5 = (S4-S2)/G1
S6 = ACOS((CD-COS(DHEO))/CS)
S7 = (S4-S6)/GB
S8 = (S3+S5+S7)/S1
O2Z1 = ((2.0*R2)+(2.0*RBL))/10.0
O2Z2 = (10.0*O2Z1)
PO1 = 8.0*(A0N+2)*(IND(DHE0)*COS(DHE0)*O2Z2)
PRSO = FO1/(((B11+RE-(PI5))/4.0)
TL = (10.0*ALOGB1)*PRSO
MODE = 'B01'
TDIF = T+(2.0+H1)*TD-(PO/C2)-(H+C1)
RETURN
END
TYPE BT3,F4

SUBROUTINE BT3(TB2,RBL)
COMMON /IN/ VAX,VAY,XIA,YIA,H,F,V,VS,XIS,DIRN,RT
COMMON /OUT/ T,R,RP,TIDIF,TL,DOF,DEPHI,FR50,AMODE
COMMON /CC/ C1,C2,CS,CE,DD,DS,MTHNH,CTRND,MIN,ETA,GS,GB,RO,TO,P1,
1 61
IF(F.LE.300.0) GO TO 1
IF(F.GT.300.0.AND.F.LT.750.0) GO TO 2
IF(F.GE.750.0.AND.F.LE.1500.0) GO TO 3
IF(F.GT.1500.0.AND.F.LE.2700.0) GO TO 4
IF(F.GT.2700.0.AND.F.LE.5000.0) GO TO 5
IF(F.GT.5000.0) GO TO 6

TYPE 556

556 FORMAT(1H1,'OUTSIDE FREO. RANGE OF PROGRAM')
1 IF(TB2.LE.11.0) GO TO 8
2 RBL=0.0
3 GO TO 100
200 IF(TB2.GT.11.0.AND.TB2.LT.50.0) GO TO 9
4 RBL=(10.0+39.0)2.82
5 GO TO 100
300 IF(TB2.GE.50.0) GO TO 10
6 GO TO 500
10 RBL=10.0
11 GO TO 100
2 IF(TB2.LE.13.0) GO TO 11
12 RBL=3.0
13 GO TO 100
400 IF(TB2.GT.13.0.AND.TB2.LT.20.0) GO TO 12
14 GO TO 600
12 RBL=(2.3.TB2+7.0)+1.27
15 GO TO 100
600 IF(TB2.GE.20.0.AND.TB2.LE.35.0) GO TO 13
16 GO TO 700
13 RBL=(3.4.TB2+15.0)+.77
17 GO TO 100
700 IF(TB2.GT.35.0.AND.TB2.LT.52.0) GO TO 14
18 GO TO 701
14 RBL=(1.6.TB2+10.0)+3.1
19 GO TO 100
701 IF(TB2.GE.52.0) GO TO 15
20 GO TO 500
15 RBL=11.0
21 GO TO 100
3 IF(TB2.LE.15.0) GO TO 16
22 GO TO 702
16 RBL=3.0
23 GO TO 100
702 IF(TB2.GT.15.0.AND.TB2.LT.45.0) GO TO 17
24 GO TO 703
17 RBL=(2.0.TB2+30.0)+1.0
25 GO TO 100
703 IF(TB2.GE.45.0) GO TO 18
26 GO TO 500
18 RBL=11.0
27 GO TO 100
4 IF(TB2.LE.13.0) GO TO 19
28 GO TO 704

B-66
19 RBL=3.0
GO TO 100

704 IF(TB2.GT.13.0.AND.TB2.LT.42.0) GO TO 20
GO TO 705

20 RBL=((8.0*TB2)/29.0)-.59
GO TO 100

705 IF(TB2.GE.42.0) GO TO 21
GO TO 500

21 RBL=11.0
GO TO 100

5 IF(TB2.LE.2.5) GO TO 22
GO TO 706

22 RBL=7.0
TYPE 1111
1111 FORMAT(1H , 'RBL HAS ERROR GREATER THAN 2 DB')
GO TO 100

706 IF(TB2.GT.2.5.AND.TB2.LE.12.5) GO TO 23
GO TO 707

23 RBL=((7.0*TB2)/10.0)+5.25
TYPE 1111
GO TO 100

707 IF(TB2.GT.12.5.AND.TB2.LT.32.5) GO TO 24
GO TO 708

24 RBL=((1.5*TB2)/20.0)+13.06
TYPE 1111
GO TO 100

708 IF(TB2.GE.22.5) GO TO 25
GO TO 500

25 RBL=15.5
TYPE 1111
GO TO 100

6 IF(TB2.GE.7.5) GO TO 26
GO TO 709

26 RBL=8.0
GO TO 100

709 IF(TB2.GT.7.5.AND.TB2.LT.15.0) GO TO 27
GO TO 710

27 RBL=((5.0*TB2)/7.5)+3.0
GO TO 100

710 IF(TB2.GE.15.0.AND.TB2.LT.25.0) GO TO 28
GO TO 711

28 RBL=((3.0*TB2)/10.0)+8.5
GO TO 100

711 IF(TB2.GE.25.0) GO TO 29
GO TO 500

29 RBL=16.0
GO TO 100

500 TYPE 555
555 FORMAT(1H , 'CANT FIND REFLECTION LOSS')
100 RETURN

END
TYPE BT5.F4

SUBROUTINE BT5(TB2.RBL)

COMMON /IN/ VAX,VAY,XIA,YIA,H,F,VS,XIS,DIK,AXN,BT
COMMON /OUT/ TR,RP,TA,TDIF,TL,DOP,DE,PHI,PR50,AMODE
COMMON /CC/ CI,C2,CS,CE,DB,DS,CMIN,CMAX,DMIN,ETA,GS,GE,RO,TO,PI,

1       61
IF(F.LE.300.0) GO TO 1
IF(F.GT.300.0.AND.F.LT.750.0) GO TO 2
IF(F.GE.750.0.AND.F.LE.1500.0) GO TO 3
IF(F.GT.1500.0.AND.F.LE.2700.0) GO TO 4
IF(F.GT.2700.0.AND.F.LE.5000.0) GO TO 5
IF(F.GT.5000.0) GO TO 6

TYPE 556

556 FORMAT(1H,'OUTSIDE RANGE OF FREQUENCIES.')

GO TO 100
1       60
IF(TB2.LE.5.0) GO TO 8
GO TO 200
8       40
RBL=4.0
GO TO 100
200     40
IF(TB2.GT.5.0.AND.TB2.LT.21.0) GO TO 9
GO TO 300
9       40
RBL=(<10.0*TB2>/16.0)*.88
GO TO 100
300     40
IF(TB2.GE.21.0) GO TO 10
GO TO 500
10      40
RBL=14.0
GO TO 100
2       60
IF(TB2.LE.3.0) GO TO 11
GO TO 400
11      40
RBL=9.0
GO TO 100
400     40
IF(TB2.GT.3.0.AND.TB2.LT.22.0) GO TO 12
GO TO 500
12      40
RBL=(<8.0*TB2>/19.0)+6.74
GO TO 100
500     40
IF(TB2.GE.22.0) GO TO 13
GO TO 500
13      40
RBL=16.0
GO TO 100
3       60
IF(TB2.LE.2.5) GO TO 16
GO TO 702
16      40
RBL=9.0
GO TO 100
702     40
IF(TB2.GT.2.5.AND.TB2.LT.17.5) GO TO 17
GO TO 703
17      40
RBL=(<9.0*TB2>/15.0)+7.5
GO TO 100
703     40
IF(TB2.GE.17.5) GO TO 18
GO TO 500
18      40
RBL=19.0
GO TO 100
4       60
IF(TB2.LE.2.5) GO TO 19
GO TO 704
19      40
RBL=9.0
GO TO 100
704     40
IF(TB2.GT.2.5.AND.TB2.LT.20.0) GO TO 20
GO TO 705
20      40
RBL=(<9.0*TB2>/17.5)+7.71
GO TO 100
705     40
IF(TB2.GE.20.0) GO TO 21
GO TO 500
21   RBL=10.0
    GO TO 100
5    IF(TB2.LE.2.5) GO TO 22
    GO TO 706
22   RBL=11.0
    GO TO 100
706  IF(TB2.GT.2.5.AND.TB2.LT.12.5) GO TO 23
    GO TO 707
23   RBL=((7.1*TB2)/10.0)+9.23
    GO TO 100
707  IF(TB2.GE.12.5.AND.TB2.LT.22.5) GO TO 24
    GO TO 708
24   RBL=((1.9*TB2)/10.0)+15.73
    GO TO 100
708  IF(TB2.GE.22.5) GO TO 25
    GO TO 500
25   RBL=20.0
    GO TO 100
6    IF(TB2.LE.2.5) GO TO 26
    GO TO 709
26   RBL=10.0
    GO TO 100
709  IF(TB2.GT.2.5.AND.TB2.LT.10.0) GO TO 27
    GO TO 710
27   RBL=((6.0*TB2)/7.5)+8.0
    GO TO 100
710  IF(TB2.GE.10.0.AND.TB2.LT.20.0) GO TO 28
    GO TO 711
28   RBL=((0.9*TB2)/10.0)+13.0
    GO TO 100
711  IF(TB2.GE.20.0) GO TO 29
    GO TO 500
29   RBL=19.0
    GO TO 100
500  TYPE 555
555  FORMAT(1H "CANT FIND REFLECTION LOSS")
100  RETURN
END

TYPE ERF.S.F
SUBROUTINE ERF(ARG,ERF)
ARG=ARG/ABS(ARG)
X=ABS(ARG)
ER=1.0
IF(X.GT.4.8) GO TO 200
E=1.0+.0705220784*X+.0422820123*X*X+.0092705272*(X*X*X)+
  .0001520143*(X*X*X)+.00002765672*(X*X*X*X)+.00000430638*(X*X*X*X)
+ER=1.0-1.0/(E**16)
200  ERF=ER*ARG
RETURN
END

B-69