GAMARAY User's Guide

Evan K. Westwood

October 16, 1996

Contents

List of Figures

1 INTRODUCTION

GAMARAY is a broadband propagation model that uses ray theory to predict the acoustic field in range-invariant, layered-bottom ocean environments. The model may be run for a simple case of propagation from a single point source to a single point receiver, or for the more complicated case of propagation from a source moving horizontally along a series of straight-line segments to an arbitrarily spaced, volumetric array. To run the model, the user must edit two input files: (1) the **.svp** file that contains the geoacoustic profile of the ocean/layered-bottom environment, and (2) the **.opt** file that contains a number of options, including the source-receiver geometry, the ray paths to consider, and the types of outputs desired. GAMARAY was developed on an Alliant FX8 at the Applied Research Laboratories, The University of Texas at Austin, and is written in Fortran77.

The GAMARAY model is based on theory described in References 1 and 2. In modeling the ocean environment, the user may: (1) specify multiple, elastic layers in the ocean bottom, (2) use linear or curved segments to connect the sound velocity profile points in the ocean, (3) use linear or curved (BLUG) profiles in ocean bottom layers, and (4) specify arbitrary geoacoustic parameters for the halfspace above the ocean. In calculating the acoustic field in terms of rays, the user may choose to: (1) correct for ray theory caustics, (2) include beam displacement in the ray trajectories, (3) include surface and/or substrate loss by specifying a certain type of scattering or by supplying a file containing loss vs. angle and frequency, (4) consider only certain ray paths, and (5) ignore eigenrays weaker than a specific threshold.

GAMARAY may be used to address a variety of acoustic propagation problems. For example, it can calculate the transfer function (complex field vs. frequency) or impulse response between a single point source and receiver. Or, it can compute the propagation loss (complex field vs. range) for a given source-receiver range interval. And finally, it can compute the field at an arbitrary volumetric array due to a source moving at constant depth along a series of straight-line segments. Several of the outputs GAMARAY can produce are[1]: (1) a list of eigenrays and their characteristics (travel time, angle, strength, etc.) for each source-receiver configuration chosen; (2) a picture of the eigenrays that travel from source to receiver; (3) a plot and listing of propagation loss vs. range or frequency; (4) a binary file containing the transfer function (FFT of the impulse response) between source and receiver; (5) a binary file containing the impulse response between source and receiver; (6) a plot of the plan view of a chosen receiving array and source track specification; (7) a plot of the R–θ functions for the source and receiver depths chosen and the ray paths specified; (8) a file containing bottom loss vs. grazing angle; and (9) a file containing a list of the eigenray angles and their complex fields from which one can

[1] The plotting capabilities referred to in this manual were originally written for the Cyber 830 computer. The plotting subroutines remain in the code and currently generate ASCII text files of numbers, which can be transferred to a personal computer to be plotted.

apply a beam pattern to the source or receiver.

Section 2 of this user's guide describes the input files required by the GAMARAY model. Section 3 covers the procedure for running the model using a UNIX script. The form and contents of GAMARAY's output files are summarized in Section 4. Finally, some of the theoretical considerations relevant to the ray model are discussed in Section 5.

• Items that have bullets beside them in the left margin are items that have been changed recently. Users should refer to the input files /usr/fnevan/inp/gama.svp and /usr/fnevan/inp/gama.opt for the latest version of the input file formats.

2 GAMARAY **INPUT FILES**

GAMARAY requires two input files: (1) the **.svp** file that contains the geoacoustic parameters of the environment, and (2) the **.opt** file that contains user options. In both of these files, an asterisk as the first character on a line is used to indicate that the next line contains data to be read by the program. An explanation of the data expected by GAMARAY is given by text on and before each asterisk line. When editing input files, the user may *not* delete or add an asterisk line, but *may* insert or modify lines before an asterisk line for additional comments. Except where noted, the program inputs the data with *unformatted read statements*, so numbers can be separated by commas, spaces, or even carriage returns. Separating numbers with several spaces improves readability. The asterisk lines in each file are numbered so that when the program cannot read a line or reads illegal data, the resulting error message indicates the line number.

2.1 The .svp Input File

An example **.svp** file, which can be copied from **/usr/fnevan/inp/gama/gama.svp**, appears in Table 2.1. The input items in the **.svp** file are:

(1) Title of profile (64 characters maximum).

(2) Geoacoustic parameters of medium above ocean:

   (i) compressional velocity, CP.

   (ii) shear velocity, CS.

   (iii) density, RHO.

   (iv) compressional wave attenuation, KP.

   (v) shear wave attenuation, KS.

   (For air, $CP = 343$ m/s, $RHO = .00121$ g/cc, $CS = KP = KS = 0$.)

(3) Number of sound velocity profile points in the ocean ($2 \leq N_{\mathrm{SVP}} \leq 50$). The program will read this number of lines in line (4).

(4) Depth, sound velocity points. The first depth must be 0 (the ocean surface) and the last depth (a positive number) is assumed to be the depth of the ocean. The first line must also include the water column density (typically 1.05 g/cc) and compressional wave attenuation. (This input allows the user to substitute other materials for the ocean itself, such as the atmosphere.)

(5) Number of layers in ocean bottom ($0 \leq N_{\text{LAY}} \leq 42$). The program will read this number of lines in line (6).

(6) Geoacoustic parameters for each bottom layer.

(i) TYPE of compressional velocity profile (1=linear; 2,3,4=BLUG).

(ii) layer thickness, H.

(iii) compressional velocity at top of layer, CP_1.

(iv)
- TYPE=1: compressional velocity at bottom of layer, CP_2.
- TYPE=2: velocity gradient at top of BLUG layer, G.
- TYPE=3,4: compressional velocity at bottom of BLUG layer, CP_2.

(v) shear velocity at top of layer, CS_1.

(vi) shear velocity at bottom of layer, CS_2.

(vii) density at top of layer, RHO_1.

(viii) density at bottom of layer, RHO_2.

(ix) compressional attenuation at top of layer, KP_1.

(x) compressional attenuation at bottom of layer, KP_2.

(xi) shear attenuation at top of layer, KS_1.

(xii) shear attenuation at bottom of layer, KS_2.

(xiii)
- TYPE=1: not read.
- TYPE=2,3: BLUG parameter β, $BETA$.
- TYPE=4: velocity gradient at top of BLUG layer, G.

Note: If $BETA$ is set to -999 the program will choose $BETA$ such that the sound velocity gradient across the top of the layer is continuous. $BETA = -999$ is therefore illegal for the first bottom layer.

(xiv)
- TYPE=1: not read.
- TYPE=2,3,4: angle-independent reflection coefficient at bottom of layer, R_{BLUG}. Enter 0 to use usual plane wave reflection coefficient. Non-zero R_{BLUG} may only be specified for the deepest layer.

(7) Geoacoustic parameters for substrate [same format as line (2)].

The BLUG profile in the ocean bottom is a three-parameter curve given by

$$c(z) = \sqrt{c_1^2(1+\beta)^2 + 2gc_1(1+\beta)z} - \beta c_1 \quad , \tag{2.1}$$

where the depth z varies from 0 to the layer thickness H, c_1 (CP_1) and c_2 (CP_2) are the sound speeds at the top and bottom of the layer, g is the sound velocity gradient at the top of the layer (G), and β is the curvature parameter ($BETA$). By setting TYPE=2 in item (6)(i), the user may choose to specify c_1, g, and β, which are the parameters usually specified in databases. Alternatively, the user may specify the BLUG curve by c_1, c_2 , and β (TYPE=3) or by c_1, c_2 , and g (TYPE=4). In summary, the three different TYPE options for the BLUG profile are designed to allow the user to specify the three BLUG parameters most readily available. The relationships between the various BLUG parameters are given by the following expressions:

$$c_2(c_1,\beta,g,H) = \sqrt{c_1^2(1+\beta)^2 + 2gc_1(1+\beta)H} - \beta c_1 \tag{2.2}$$

$$g(c_1,c_2,\beta,H) = \frac{c_2-c_1}{H}\left(1+\frac{c_2-c_1}{2c_1(1+\beta)}\right) \tag{2.3}$$

$$\beta(c_1,c_2,g,H) = \frac{(c_2-c_1)^2}{2c_1(c_1-c_2+gH)} - 1 \tag{2.4}$$

The standard units assumed by GAMARAY in the **.svp** file are: distance in m, velocity in m/s, density in kg/m^2, attenuation in dB/m-kHz, velocity gradient in s^{-1}, and frequency in Hz. Parameter values in different units may be used as long as they are consistent with each other and the user keeps in mind that labels in the outputs will remain in the standard units.

2.2 The .opt Input File

An example **.opt** file, which can be copied from **/usr/fnevan/inp/gama/gama.opt**, appears in Table 2.5. The input items in the **.opt** file are:

(1) USER OPTIONS

(i) $FREQ$ is the frequency at which the field calculations are done for (1) the eigenray picture [if the PIC option is chosen in line (2)], (2) the bar graph [if the BAR option is chosen in line (2)], and (3) the eigenray list [if the EIG option is chosen in line (2)].

(ii) $CUT_{\rm DB}$ determines the cutoff level for weak eigenrays: any eigenray that is at least $CUT_{\rm DB}$ dB weaker than the *strongest* eigenray is not included in the calculations. For example, if the strongest eigenray at a particular receiver has a strength of -50 dB and $CUT_{\rm DB}$ is chosen as 40 dB, then any eigenrays that are weaker than -90 dB will be ignored. It is usually quite safe to ignore rays that are 1/50 the amplitude of the strongest ray, in which case $CUT_{\rm DB}$ would be set to $20\log(1/50) = 34$ dB. For very precise propagation loss calculations (benchmarks problems, for example) one might set $CUT_{\rm DB}$ as high as 50 dB. Two more points should be made: (1) Eigenray magnitudes are compared to

Table 1: Example **.svp** file

```
$$$ SVP FILE FOR GAMARAY PROGRAM ON ALLIANT $$$
$$$ NOTE: program reads lines following starred lines.  do not delete
          any starred lines. do not omit data, even if not needed. $$$
-----------------------------------------------------------------------
*(1) TITLE (up to 64 characters):
gama svp
-----------------------------------------------------------------------
GEOACOUSTIC PARAMETERS OF HALFSPACE ABOVE OCEAN:
*(2) cp; cs; rho; kp; ks;
343.  0.  .00121  0.0  0.0
-----------------------------------------------------------------------
SOUND VELOCITY PROFILE IN OCEAN:
*(3) nsvp = # of depth, sound velocity points
20
*(4) depth, sound velocity points
    .00  1492.24 1.050 0. [rho, kp on first line; nsvp lines like this]
  10.00  1492.37
  20.00  1492.26
  30.00  1491.84
  50.00  1491.11
  75.00  1490.90
 100.00  1491.09
 150.00  1491.67
 200.00  1492.26
 300.00  1493.48
 500.00  1496.68
 600.00  1498.16
 800.00  1499.48
1000.00  1498.57
1200.00  1496.68
1400.00  1494.84
1500.00  1494.49
1750.00  1495.17
2000.00  1496.79
2067.00  1497.10
-----------------------------------------------------------------------
GEOACOUSTIC PARAMETERS OF OCEAN BOTTOM LAYERS:
*(5) number of bottom layers
3
LAYER PROFILES: type: 1=linear, 2=blug, 3=blug, 4=blug;   examples:
1  thick  cp1 cp2  cs1 cs2  rho1 rho2  kp1 kp2  ks1 ks2
2  thick  cp1   g  cs1 cs2  rho1 rho2  kp1 kp2  ks1 ks2  beta rblug
3  thick  cp1 cp2  cs1 cs2  rho1 rho2  kp1 kp2  ks1 ks2  beta rblug
4  thick  cp1 cp2  cs1 cs2  rho1 rho2  kp1 kp2  ks1 ks2     g rblug
   [beta=-999 ==> set beta such that g continuous from previous layer,
    rblug=0   ==> use usual plane wave r/t coeff at bottom of layer]
*(6)
3  200.00 1499  1827  87.  484. 1.50 1.77  .100 .144  15  21.6 .4842 0
3  250.00 1827  2194  484  749. 1.77 1.96  .144 .141 21.6 21.0  -999 0
3  450.00 2194  2738  749 1174  1.96 1.98  .141 .088 21.0 13.2  -999 0
-----------------------------------------------------------------------
GEOACOUSTIC PARAMETERS OF SUBSTRATE:
*(7) cp; cs; rho; kp; ks; in substrate
6000  3000  2.6  .03  .07
```

the magnitude of the *strongest eigenray found*, not to the magnitude of the *total* field. It is possible that the strongest eigenray could destructively interfere with another eigenray, and thus make weaker eigenrays relatively more important to the total field. An example is when a source or receiver is placed a fraction of a wavelength from a pressure-release interface. (2) It is possible for two or more weak eigenrays to add together constructively, and thereby affect the total field by a greater amount.

(iii)
- $II_{\text{SEG}} = 1$ for linear segmentation of ocean SVP.
- $II_{\text{SEG}} = 2$ for curved segmentation of ocean SVP.

If refraction in the ocean water column is important, linear segmentation can generate false caustics. Curved segmentation can help alleviate this problem, although irregular SVP's may still generate unrealistic caustics. See Section 5.2 for a more detailed discussion.

(iv) SV_{TOL} is the sound velocity error tolerance allowed when fitting the ocean sound velocity profile. GAMARAY tries to skip points in the SVP, and then checks to see if the fitted profile is within SV_{TOL} m/s of the skipped point. If so, the point is skipped. Eliminating layers in the ocean SVP in this way makes the program run somewhat faster, and in some cases may make the SVP smoother and less prone to the generation of false caustics (see Section 5.2).

(v) R_{TERP} (Range inTERPolation) is the distance in range across which GAMARAY will interpolate the eigenray characteristics instead of computing them exactly. Such interpolation can save computation time when many closely spaced source-receiver ranges are involved. In general, $R_{\text{TERP}} = 200$–400 m is quite safe. If the phase of the reflection/transmission coefficient is interpolated across more than $\pi/4$, then a warning message is written to the **.out** file, and a smaller R_{TERP} should be used.

(2) OPTIONS (1=yes, 0=no)

(i) PIC is the option for generation of a plot of the eigenrays and the geoacoustic profile on the file with the suffix **_pic**. If PIC is negative, the program will plot $|PIC|$ points per ray (the default is 120, the maximum is 200).

(ii) BAR is the option for generation of a bar graph of eigenray arrivals, including their magnitudes, receiver angles, and frequency-dependent attenuations, on the file with suffix **$bar**.

(iii) EIG is the option for generation of a list of eigenrays at each receiver. The list includes eigenray characteristics such as the path taken, travel time, source and receiver angles, reflection/transmission loss, geometric spreading loss, surface scattering loss, and total field. Eigenrays are calculated at the frequency

$FREQ$ given in line (1). The list is placed in a file with the suffix **.eig**. (See Section 4.1 for more details.)

(iv) $CAUS$ is the option to include caustic corrections (see Section 5.1).

(v) BD is the option to include the frequency dependent phenomenon of beam displacement in the ray trajectories. Users are cautioned to select this option only in low-frequency, shallow water cases because it can consume significant computation time, and anomalous results may be encountered, especially when solid-solid interfaces are involved (see Section 5.4).

(vi) DAT is the option for generation of a binary data file (having the suffix **_dat**) containing all the eigenray characteristics necessary to compute the transfer function for each source-receiver configuration.

(vii) $DIAG$ is the option to print out diagnostic messages during the run. This option can create a large text file and should only be chosen if the user is familiar with the computer program.

• (viii) $METRIC$ is the option to enter the values in the **.opt** file in metric units ($METRIC = 1$) or in English units ($METRIC = 0$). For source depths and receiver arrays, distances are in meters for metric and feet for English; for ranges and source tracks, distances are in km for metric and nautical miles (nmi) for English. Source velocities are in m/s for metric and knots for English.

(3) OPTIONS. For the options II_{PL} and II_{TF} below: 0=no plots or listings; 1=magnitude of the coherent propagation loss, PL_C; 2=incoherent propagation loss, PL_I; 3=both PL_C and PL_I; 4=only a list file containing both sets of data. Options 1, 2, and 3 automatically generate a text file with the suffix **.ls** that contains a listing of the propagation loss data. PL_C is the complex number calculated from the complex pressure fields p_j of the N eigenrays found:

$$PL_C = \sum_{j=1}^{N} p_j \tag{2.5}$$

PL_I is a real number calculated from an incoherent (phaseless) sum of the eigenray magnitudes:

$$PL_I = \left(\sum_{j=1}^{N} |p_j|^2 \right)^{1/2} \tag{2.6}$$

Although PL_I does not represent a physical quantity, it tends to be a good estimate of the range-averaged PL_C. Actual field measurements, especially at high frequencies, often resemble PL_I.

(i) II_{PL} is the option for generation of plots and/or listings of PL vs. range (or time for moving sources). The frequencies at which PL is calculated are specified in line (4).

(ii) II_{TF} is the option for plots and/or listings of the transfer function vs. frequency. There will be one transfer function plot for each source-receiver range; the frequencies on the horizontal axis are given in line (4).

(iii) II_{BMP} (0=no, 1=yes) is the option to generate a "beam pattern" (**.bmp**) file from which one can compute the field due to a directional source or receiver. The format of the beam pattern file is given in Section 4.4.

(iv) II_{BLT} (0=no, 1=yes) is the option to generate a bottom loss table (**.blt**) file. Calculations are done for the angles given in the following three items and the frequencies given in line (4).

(v) N_{ANG} is the number of angles at which to calculate the bottom loss table.

(vi) ANG_1 is the first grazing angle at which to calculate the bottom loss table.

(vii) ANG_2 is the final grazing angle at which to calculate the bottom loss table.

When the II_{BLT} option is chosen, GAMARAY chooses N_{ANG} source-receiver ranges such that the specularly reflected rays from source to receivers have the appropriate grazing angles between ANG_1 and ANG_2. Only the single bottom bounce ocean path ('d' 0 1) is included, along with the bottom paths given in line (17). To get the bottom loss, the total field is normalized by the geometric spreading loss and travel time phase of the specularly reflected ray. Calculations are done at the frequencies given in line (4). The bottom loss table output file has the suffix **.blt**; its format is described in Section 4.5.

(4) Frequencies for propagation loss calculations requested in line (3):

- $N_{\mathrm{FR}} > 0$: Give N_{FR} and frequencies $f_1, f_2, \ldots f_{N_{\mathrm{FR}}}$
- $N_{\mathrm{FR}} < 0$: Give N_{FR}, f_1, Δ_f, which generates the frequencies $f_1 + (j-1)\,\Delta_f$, $j = 1, |N_{\mathrm{FR}}|$.

Note: $|N_{\mathrm{FR}}| \leq 105$.

(5) Transfer function and impulse response file options. The transfer function is simply the coherent propagation loss calculated as a function of frequency. The impulse response function is the inverse FFT of the transfer function. In the time domain, the total time window for the impulse response is of length $T_W = N_{\mathrm{FFT}}/f_s$. To prevent wraparound, the time window starts at time $T_S = T_E - .04T_W$, where T_E is the travel time of the earliest arriving eigenray.

(i) II_{FFT} (0=no, 1 or 2=yes) is the option to generate an **_fft** file containing transfer functions of $N_{\mathrm{FFT}}/2{+}1$ complex*8 numbers ($N_{\mathrm{FFT}}{+}2$ real*4 numbers). The transfer functions are obtained by calculating the coherent propagation loss at $N_{\mathrm{FFT}}/2$ equally spaced frequencies between 0 and $f_s/2$. See Section 4.2 for details on the format of the **_fft** file.

- $II_{\text{FFT}} = 1$: Only eigenrays arriving between T_S and $T_S + .5T_W$ are included in the calculations. Such transfer functions are suitable for correlation processing.
- $II_{\text{FFT}} = 2$: No time window editing of eigenrays is performed.

(ii) II_{IR} (0=no, 1 or 2=yes) is the option to generate an **_ir** file containing N_{FFT}-point impulse response functions. See Section 4.3 for details on the format of the **_ir** file.

- $II_{\text{IR}} = 1$: Only eigenrays arriving between T_S and $T_S + .5T_W$ are included in the calculations. Such impulse responses are suitable for correlation processing.
- $II_{\text{IR}} = 2$: Only eigenrays arriving between T_S and $T_S + .96T_W$ are included in the calculations. Such impulse responses should not exhibit wrap-around.

(iii) N_{FFT} is the number of points in the FFT. If N_{FFT} is not a power of 2, it will be set to the nearest one.

(iv) f_s is the sample frequency.

(v) f_{min} is the minimum frequency of interest in the source spectrum. It is used when calculating the eigenrays strengths and testing for weakness using CUT_{DB}.

(vi) f_{max} is the maximum frequency of interest in the source spectrum. It is used to decide how close to the receiver a ray has to come before it is counted as an eigenray. (The range error tolerance is set to $.005\lambda_{\text{min}}$, where $\lambda_{\text{min}} = c_r/f_{\text{max}}$ and c_r is the sound speed at the receiver.)

(6) Source depths (measured positive downward from the ocean surface in m or ft, depending on $METRIC$ option):

- $N_{\text{ZS}} > 0$: Give N_{ZS} and the source depths $ZS_1, ZS_2, \ldots ZS_{N_{\text{ZS}}}$.
- $N_{\text{ZS}} < 0$: Give N_{ZS}, ZS_1, Δ_{ZS}, which generates the depths $ZS_1 + (j-1)\,\Delta_{\text{ZS}}$, $j = 1, |N_{\text{ZS}}|$.

Note: $1 \leq |N_{\text{ZS}}| \leq 60$. Also, when ZS is specified as negative, the source depth is measured from the ocean bottom. For example, if the water depth in the **.svp** file is $H = 2000$ m and $ZS = -10$ m, then the source depth is set to $H + ZS = 1990$ m. The quantity Δ_{ZS} retains its usual definition of positive downward. This facility applies also to receiver depths ZR [see Figure 4] and is useful when modeling sources or receivers that are bottom-mounted.

• (7) Source-receiver ranges in km or nmi, depending on $METRIC$ option.

(i) $N_R > 0$: Give N_R and ranges $R_1, R_2, \ldots R_{N_R}$.

(ii) $N_R < 0$: Give N_R, R_1, Δ_R, which generates the depths $R_1 + (j-1)\,\Delta_R$, $j = 1, |N_R|$.

(iii) $N_R = 0$: Read source track specification in line (9).

Note: For $N_R > 0$, $|N_R| \leq 101$. Otherwise, the number of ranges is only limited by the storage capacity of the computer.

- (8) Source track specification (see Figures 1 and 2). Depending on the $METRIC$ option, distances are in km or nmi, and velocities are in m/s or knots.

  (i) N_{LEG} is the number of straight-line legs of the source track.

  (ii) II_{PIC} is an option to produce a text file containing the coordinates of the source track and the receiving array. Plotting the source track and array geometry is a useful way to verify that the variables in line (9) were entered correctly.

- (9) Leg specification (see Figures 1 and 2).

  (i) TYPE (1=polar form; 2=x-y form).

  (ii)
  - $II_{\text{CONT}} = 0$ makes the current leg independent (not continuous with) the previous leg. For $TYPE = 1$, $T1$ and $T2$ are measured with respect to the CPA point. For the first leg II_{CONT} is automatically set to zero.
  - $II_{\text{CONT}} = 1$ makes the current leg continuous with the previous leg. For $TYPE = 1$, CPA is ignored; for $TYPE = 2$, $X1$ and $Y1$ are ignored. The total time along the current leg is taken to be $T2 - T1$.

  (iii) V_S is the source velocity.
  - For $TYPE = 1$, $V_S = 0$ implies that $T1$ and $T2$ are distances.
  - For $TYPE = 2$, $V_S \neq 0$ implies that $T2$ is computed from the length of the leg and the velocity (the entered value of $T2$ is ignored).

  (iv) Track sampling parameter, $+DT/-NPT$.
  - Positive values are interpreted as DT, the time in min between which samples of the source track are taken.
  - Negative values are interpreted as NPT, the total number of points at which to sample the leg ($NPT \geq 2$).

  (v) $T1$ is the start time of the leg (in min). For $TYPE = 1$, setting $V_S = 0$ implies that $T1$ and $T2$ are given as distances (in km or nmi).

  (vi) $T2$ is the end time of the leg (in min). For $TYPE = 2$, non-zero V_S means that this value is computed automatically.

  (vii)
  - TYPE=1: CPA is the closest point of approach of the source to the reference point of the receiving array. The CPA distance is defined to be

positive for CPAs above the x-axis ($y > 0$) and negative for CPAs below the x-axis ($y < 0$).

- TYPE=2: $X1$ is the starting x-coordinate.

(viii)
- TYPE=1: PHI is the direction of the source track, given by the angle (in deg) from the positive x-axis to the positive y-axis.
- TYPE=2: $Y1$ is the starting y-coordinate.

(ix)
- TYPE=1: Dummy argument, set to zero.
- TYPE=2: $X2$ is the final x-coordinate.

(x)
- TYPE=1: Dummy argument, set to zero.
- TYPE=2: $Y2$ is the final y-coordinate.

(10) Type of receiver array (see Figures 3 and 4).

- $II_{\mathrm{ARR}} = 1$ for a uniformly spaced receiving array.
- $II_{\mathrm{ARR}} = 2$ for a nonuniformly spaced receiving array.

Note: The center of the array for the purposes of defining the CPA in line (9) is defined to be at $(x, y, z) = (0, 0, ZR_1)$. Array locations are specified in m or ft, depending on the $METRIC$ option.

(11) Receiver array specification for $II_{\mathrm{ARR}} = 1$ (see Figure 3).

(i) ZR_1 is the depth of the *top* of the array. Negative ZR_1 is measured from the ocean bottom as explained in item (6) for ZS.

(ii) N_x (≥ 1) is the number of elements in the x-direction.

(iii) N_y (≥ 1) is the number of elements in the y-direction. **Note**: $N_x * N_y \leq 32$.

(iv) N_z is the number of elements in the z-direction. **Note:** $1 \leq N_z \leq 60$.

(v) Δ_x is the element spacing in the x-direction.

(vi) Δ_y is the element spacing in the y-direction.

(vii) Δ_z is the element spacing in the z-direction. Note: a negative Δ_z would make ZR_1 the depth of the *bottom* of the array.

(12) Receiver array specification for $II_{\mathrm{ARR}} = 2$: Specification of receiver depths ZR (see Figure 4).

- $N_{\mathrm{ZR}} > 0$: Give N_{ZR} and the receiver depths $ZR_1, ZR_2, \ldots ZR_{N_{\mathrm{ZR}}}$.
- $N_{\mathrm{ZR}} < 0$: Give N_{ZR}, ZR_1, $\Delta Z_2, \Delta Z_3$, $\ldots$ $\Delta Z_{N_{\mathrm{ZR}}}$, where ZR_1 is the depth of the first element and ΔZ_i is the spacing (positive downward) between elements $i - 1$ and i.

Note: $1 \leq |N_{\text{ZR}}| \leq 60$. Negative ZR_is are measured from the ocean bottom as explained in item (6) for ZS and illustrated in Figure 4.

(13) Receiver array specification for $II_{\text{ARR}} = 2$: Specification of the (x, y) locations of the receivers. One line for each of the N_{ZR} depths given in line (12) must be given (see Figure 4).

(i) N_{xy} is the number of receivers in the (x, y) plane at the current depth, $1 \leq N_{xy} \leq 32$.

(ii) $x_1, y_1\ x_2, y_2 \ldots x_{Nxy}, y_{Nxy}$ are the (x, y) locations of the N_{xy} receivers at the current depth.

(14) Ocean path specification (see Figure 5).

(i) N_{STAND} (≤ 10) is the number of standard sets of ocean paths to be specified in line (15).

(ii) N_{INDIV} (≤ 20) is the number of individual ocean paths to be specified in line (16).

(15) Standard ocean paths (see Figure 5). A "bottom interaction" of a ray path is defined to occur when the ray enters the bottom section of the ocean (the ray need not touch the ocean bottom). The term "top interaction" is defined analogously. The ray types are defined as follows:

'a' All rays.

'r' Refracting rays that do not touch the ocean bottom.

'w' Waterborne rays that either refract or totally reflect off the ocean bottom.

'p' Rays that penetrate into the ocean bottom.

Unless the effects of some specific types of rays are being investigated, one should specify **'a'** to allow GAMARAY to search for all rays. N_{STAND} lines must be given of the form:

(i) starting number of bottom interactions ($0 \leq KB_1$).

(ii) ending number of bottom interactions ($KB_1 \leq KB_2 \leq 48$).

(iii) ray type **'a'**, **'r'**, **'w'**, or **'p'**.

(16) Individual ocean paths (see Figure 5). N_{INDIV} lines must be given of the form:

(i) 'ini dir' is the initial direction of the ray from source ('u' for up, 'd' for down).

(ii) $\#T$ is the number of top interactions of the ray.

(iii) $\#B$ is the number of bottom interactions of the ray.

(17) Bottom paths allowed (see Figure 6). Give the maximum number of down-up traversals T for each of the N_{LAY} bottom layers in the **.svp** file. T_j is the number of traversals of the j'th bottom layer allowed per bottom reflection. For only the simplest, single-reflection paths, specify $T_j = 1$. To allow double reflections in the j'th layer on each bottom interaction, specify $T_j = 2$.

(i) $T_1, T_2 \ldots T_{N_{\text{LAY}}}$

(18) Surface scattering (loss) model (see Section 5.3):

(i) II_{SL} = scattering model.

(ii) SL_{PARM} = scattering parameter.

- $II_{\text{SL}} = 0$ for no surface scattering; SL_{PARM} = dummy.
- $II_{\text{SL}} = 1$ for Eckart scattering; SL_{PARM} = wind speed in knots.
- $II_{\text{SL}} = 2$ for the Modified Eckart scattering model; SL_{PARM} = wind speed in knots.
- $II_{\text{SL}} = 3$ for the Beckmann-Spizzichino scattering model; SL_{PARM} = wind speed in knots.
- $II_{\text{SL}} = 4$ for uniform scattering (angle- and frequency-independent); SL_{PARM} = loss in dB per surface reflection.
- $II_{\text{SL}} = 5$ to read in a bottom loss table; SL_{PARM} = file name in single quotes. The required format for the bottom loss table file is given in Section 4.5.

(19) Substrate scattering (loss) model. This line uses the same parameters as for surface scattering in line (18).

(20) R–θ plot option (see Section 5.6 for details on the use of these plots):

(i) II_{RTH} (0=no plot, 1=plot source angle, 2=plot receiver angle, 4=output MATLAB file) is the plot option.

(ii) TH_1 is the lower grazing angle θ on horizontal axis.

(iii) TH_2 is the upper grazing angle θ on horizontal axis.

(iv) R_1 is the lower range on vertical axis.

(v) R_2 is the upper range on vertical axis.

- Setting $II_{\text{RTH}} = 4$ produces a MATLAB file with the suffix **_art** that contains the travel time t and the horizontal range R as a function of the Snell invariant $a = \cos\theta/c$ for each ray path specified. The file should be FTPed in binary mode and loaded into the MATLAB application. Positive and negative values of a indicate rays that arrive at the receiver from above and below, respectively. Rays are traces at all possible angles (the last four items in line (20) are ignored).

2.3 Example Source Tracks

In this section several example source track specifications and the resulting source tracks are illustrated. The first example uses the *x*-*y* form, which is most useful when the user has the *x*-*y* coordinates of the source. The coordinates may be known as a function of time, or the source velocity may be known over each leg of the track. In the first example, the source follows a triangular pattern at a constant speed of 5 m/s. The track is sampled at evenly spaced time intervals of 15 min, during which time the source travels 4500 m. The specification in lines (7)–(9) of the **.opt** file are as follows:

```
SRC-REC RANGES: nr>0 ==> give r1,r2,r3... in m; nr<0 ==> give r1,delr;
                nr=0 ==> read source track info on next 2 lines
*(7) nr  r1  r2(dr) ...        (km or nmi)
0  1000
*(8) SOURCE TRACK: NLEG(# linear legs); IIPIC (option to plot track)
3  1
LEG SPECIFICATION: V in m/s or kt; NPT>1; T,DT in min;
   CPA,X,Y in km or nmi; PHI in deg
  (for TYPE=1, V=0 ==> T is dist in km; for TYPE=2, V=0 ==> T2 ignored)
     1 IICONT   V   T1   T2  +DT/-NPT    CPA,PHI    DUM,DUM (POLAR FORM)
*(9) 2 IICONT   V   T1  (T2) +DT/-NPT     X1,Y1      X2,Y2    (X-Y FORM)
     2      0  5.0   0    0      15      -40,0       40,0
     2      1  5.0   0    0      15        0,0        0,-40
     2      1  5.0   0    0      15        0,0        0,40
-----------------------------------------------------------------------
```

The source track that results from this specification is illustrated in Figure 1. Note that the $T1$, $X1$, and $Y1$ variables are only used on the first leg. Since the last two legs are specified to be continuous, they are automatically set to start where the last leg finished. Also, since V_S is non-zero in each line, the variable $T2$ is computed automatically as well.

The second example uses the polar form to specify several separate tracks. The polar form is most useful when the user knows the CPA and direction of the source, rather than the *x*-*y* coordinates. The specification is as follows:

```
SRC-REC RANGES: nr>0 ==> give r1,r2,r3... in m; nr<0 ==> give r1,delr;
                nr=0 ==> read source track info on next 2 lines
*(7) nr  r1  r2(dr) ...        (km or nmi)
0  1000
*(8) SOURCE TRACK: NLEG(# linear legs); IIPIC (option to plot track)
3  1
LEG SPECIFICATION: V in m/s or kt; NPT>1; T,DT in min;
```

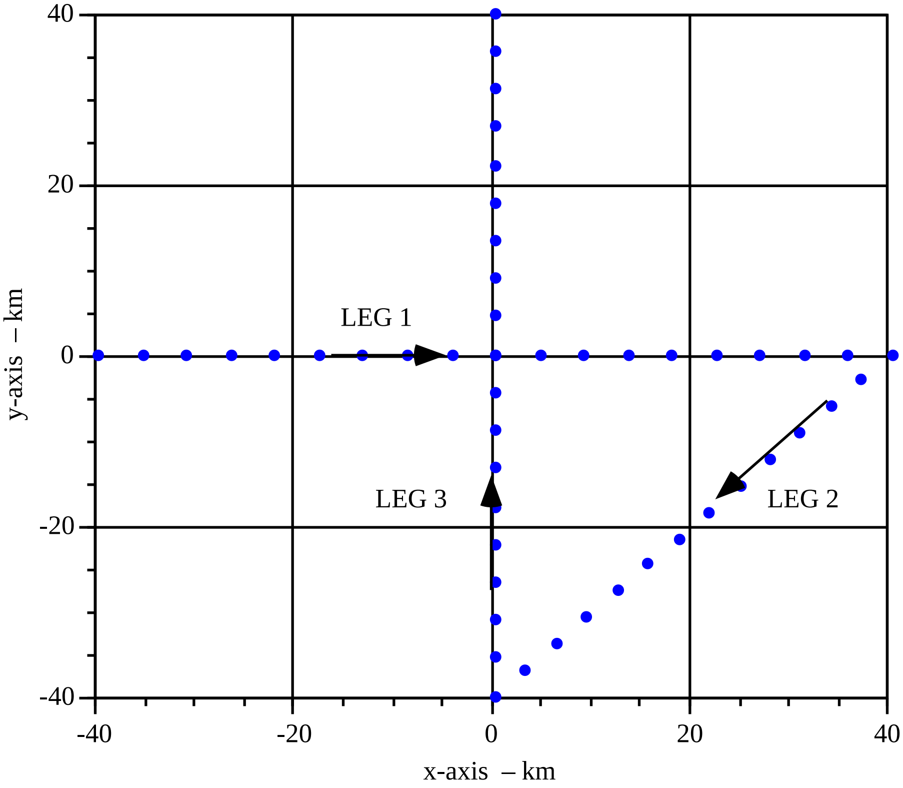

Figure 1: Source track resulting from x-y specification.

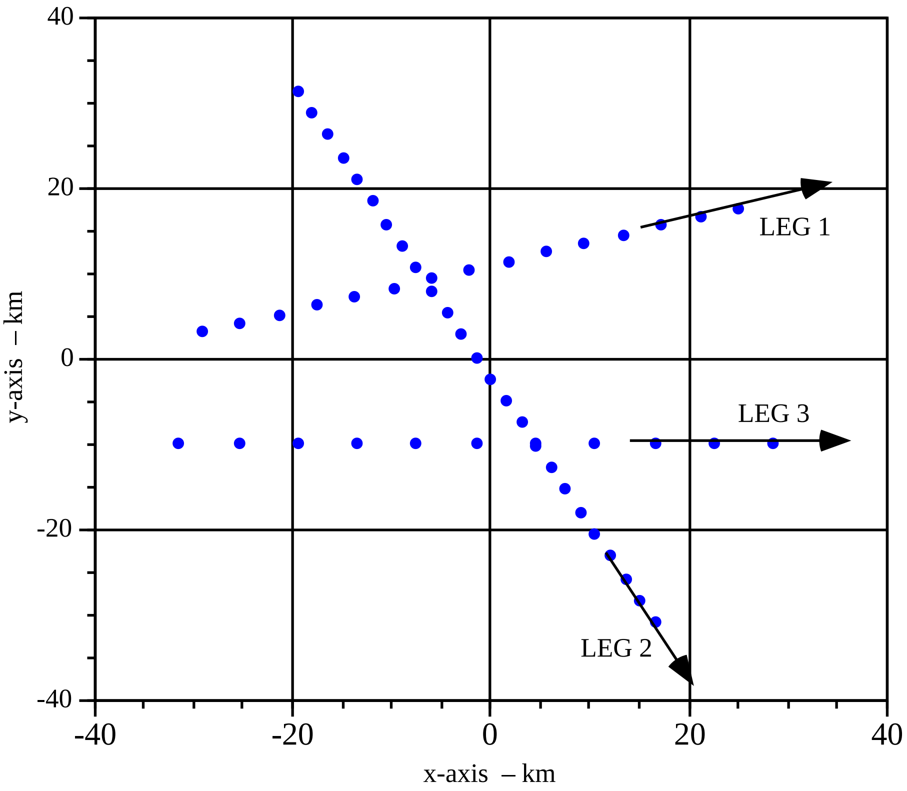

Figure 2: Source track resulting from polar specification.

```
   CPA,X,Y in km or nmi; PHI in deg
  (for TYPE=1, V=0 ==> T is dist in km; for TYPE=2, V=0 ==> T2 ignored)
     1 IICONT   V   T1   T2  +DT/-NPT    CPA,PHI    DUM,DUM (POLAR FORM)
*(9) 2 IICONT   V   T1  (T2) +DT/-NPT     X1,Y1      X2,Y2    (X-Y FORM)
     1      0   0  -30   30     -16      10,15       0,0
     1      0  5.0 -120 120      10       0,-60      0,0
     1      0   0  -30   30       6     -10,0        0,0
-----------------------------------------------------------------------
```

The source track produced is illustrated in Figure 2. For the first leg V_S is set to zero, which means that $T1$ and $T2$ are interpreted as distances (±30 km) relative to CPA. The entry of -16 for DT/NPT causes 16 points to be taken along the 60-km track, resulting in a spacing between points of $60/(16-1) = 4$ km. The CPA is 10 km, and the angle is 15° with respect to the x-axis. The second leg has a velocity of 5 m/s, the start time is 120 min before CPA, the stop time is 120 min after CPA, the track is sampled every 10 min, the CPA is zero, and the angle is 60° below the x-axis. The final leg is similar in structure to the first, except that the CPA is negative, indicating that it occurs below

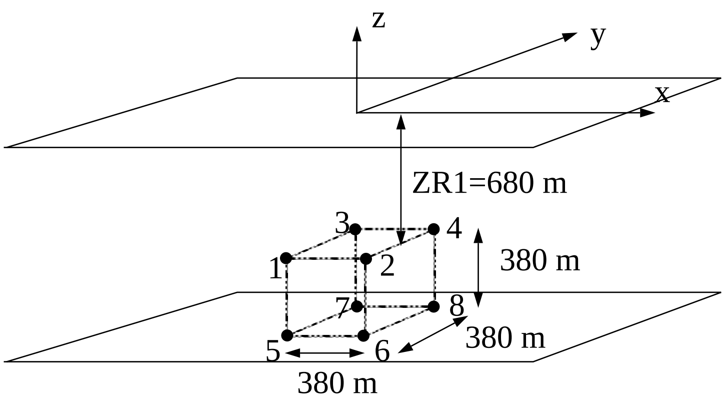

Figure 3: Example of a uniform volumetric array. The receiver numbers are those that are used in the **.eig** and **_fft** files.

the x-axis instead of above it. Note that since $V_S = 0$, DT is interpreted as the distance traveled between sample points, in this case 6 km.

2.4 Example Receiver Arrays

This section gives several examples of receiver array specifications. The first example (see Figure 3) is a uniform array whose elements form a cube that is 380 m on each side. Its top is 680 m below the ocean surface. The specification for such an array is as follows:

```
*(10) REC ARRAY SPEC: IIARR(1=uniform,2=nonuniform) (dist in m or ft)
1
IIARR=1: uniformly spaced receiver array
*(11) zr1  nx  ny  nz   dx    dy    dz  [dx,dy,dz are element spacings]
    680.  2   2   2   380.   380.   380.
```

To obtain a planar, 4-element, horizontal array, one would set $N_z = 1$; for a planar, 4-element vertical array, one would set $N_y = 1$; for a linear 2-element horizontal array, one would set $N_y = N_z = 1$; and for a 2-element vertical array, one would set $N_x = N_y = 1$.

The next example illustrates how to specify an L-array using the nonuniform array option $II_{\mathrm{ARR}} = 2$. The 4-element vertical portion of the L-array begins 10 m off the bottom and extends upward at the origin $x = y = 0$, and the 6-element horizontal portion lies on the bottom of the ocean along the x-axis. The specification is

```
*(10) REC ARRAY SPEC: IIARR(1=uniform,2=nonuniform) (dist in m or ft)
2
IIARR=1: uniformly spaced receiver array
```

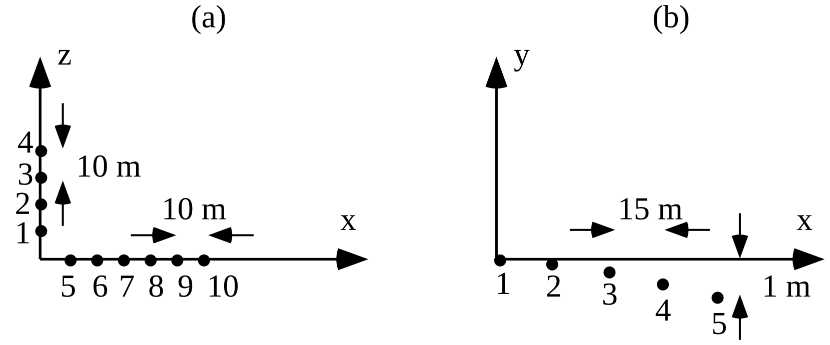

Figure 4: (a) L-array in the x-z plane. (b) Horizontal array on the bottom with curvature. The receiver numbers are those that are used in the **.eig** and **_fft** files.

```
*(11) zr1  nx  ny  nz   dx    dy    dz  [dx,dy,dz are element spacings]
IIARR=2: receiver array depths. nzr>0 ==> give zr1,zr2,... ;
   nzr<0 ==> give zr1,dz2,dz3... (dz are element spacings, pos down)
*(12) nzr   zr1  zr2(dz2) zr3(dz3) ...
        5  -10  -20  -30  -40  -1.e-10
IIARR=2: specify (x,y) receiver positions for each of the nzr depths:
*(13) nxy  x(1),y(1)  x(2),y(2) ... x(nxy),y(nxy) [nzr lines like this]
        1    0,0
        1    0,0
        1    0,0
        1    0,0
        6    10,0  20,0  30,0  40,0  50,0  60,0
------------------------------------------------------------------------
```

The resulting array is illustrated in Figure 4(a). Note that the array has been specified such that the receiver numbers begin at the bottom of the vertical array, move up the vertical array, and end by moving out on the horizontal array. The depths of the arrays are negative, indicating that they are to be measured from the bottom of the ocean upward. The final depth is given as $-1.\mathrm{e}-10$, which effectively places the horizontal array on the bottom.

The final example is a horizontal array on the ocean bottom that is slightly curved. It is illustrated in Figure 4(b) and is specified by the following:

```
*(10) REC ARRAY SPEC: IIARR(1=uniform,2=nonuniform) (dist in m or ft)
2
IIARR=1: uniformly spaced receiver array
*(11) zr1  nx  ny  nz   dx    dy    dz  [dx,dy,dz are element spacings]
```

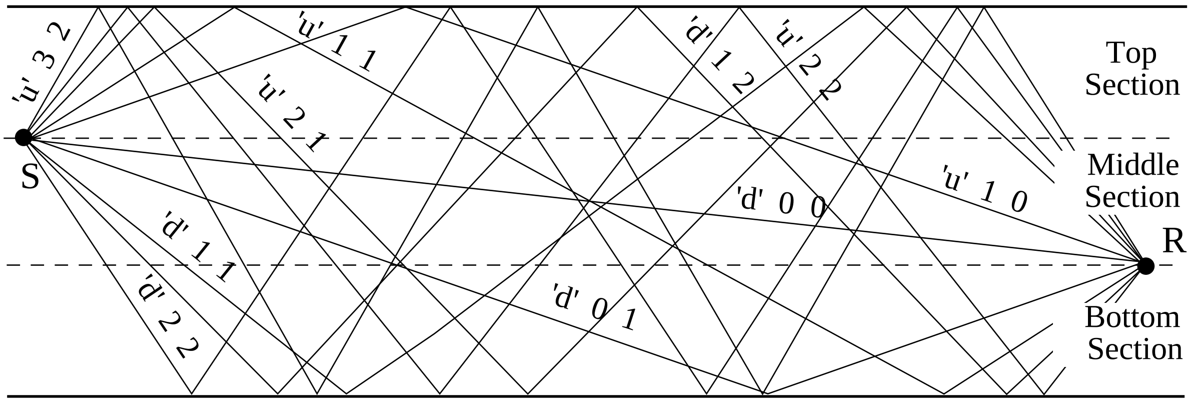

Figure 5: Example of ocean paths and their specification.

```
IIARR=2: receiver array depths. nzr>0 ==> give zr1,zr2,... ;
   nzr<0 ==> give zr1,dz2,dz3... (dz are element spacings, pos down)
*(12) nzr   zr1  zr2(dz2) zr3(dz3) ...
        1  -1.e-10
IIARR=2: specify (x,y) receiver positions for each of the nzr depths:
*(13) nxy  x(1),y(1)  x(2),y(2) ... x(nxy),y(nxy) [nzr lines like this]
        6    0,0  15,-.1  30,-.2  45,-.35  60,-55  75,-.75  90,-1.0
------------------------------------------------------------------------
```

All the receivers lie at the same depth (on the bottom), so only one depth is given. The six receivers are then given by their x-y locations on line 13.

2.5 Eigenray Path Specification

This section illustrates the eigenray path specifications required in lines (14)–(17) of the **.opt** file. Ocean paths and their specification are illustrated in Figure 5. All paths with 0–2 bottom interactions ($KB_1 = 0, KB_2 = 2$ in line (15)) are shown. Each eigenray is labeled with the individual specification that would be used in line (16).

Figure 6 illustrates the paths eigenrays may take in the ocean bottom. When the number of bounces in each layer is limited to one in line (17) of the **.opt** file, the bottom paths shown in Figure 6(a) are considered. When the number of bounces is increased to two, the bottom paths in Figure 6(b) are considered. Multiple eigenrays with the same angle and travel time, such as the three with $T(1) = 2, T(2) = 2$, are treated by GAMARAY as a single eigenray with a reflection/transmission coefficient that includes all the multipaths. From the figure, it is clear that for a geoacoustic environment with multiple layers and multiple bounces allowed within those layers, the number of possible eigenrays can be extremely large.

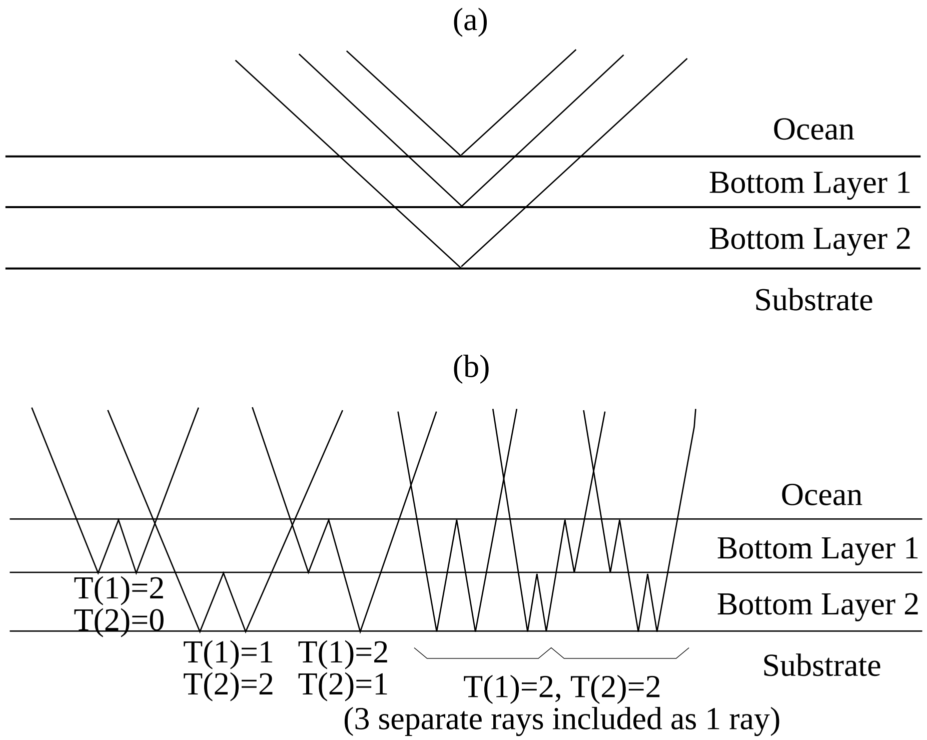

Figure 6: Example bottom paths and their specification. (a) Single bottom-bounce paths with $T(j) = 1,\ j = 1, 2$. (b) Additional paths produced by setting $T(j) = 2$, $j = 1, 2$.

Table 2: Example **.opt** file

```
$$$ OPTION FILE FOR GAMARAY PROGRAM ON ALLIANT $$$
$$$ NOTE: program reads lines following starred lines.  do not delete
          any starred lines. do not omit data, even if not needed. $$$
-----------------------------------------------------------------------
USER OPTIONS: frequency (for eig list, bd); ray cutoff margin in db;
   ocean svp segments (1=linear, 2=curved); ocean sv tolerance;
   range interpolation allowed
*(1) freq  cutdb  iiseg  svtol  rterp
      100     30      1      0    200
-----------------------------------------------------------------------
OPTIONS (1=yes, 0=no):  ray picture; bar graph; eigenray list;
   include beam displacement; include caustic corrections; output
   data file; output diagnostic messages; metric(1) or English(0)
*(2) pic; bar; eig; caus; bd; dat; diag; metric
       0    0    0     1   0    0     0    1
-----------------------------------------------------------------------
OPTIONS: iipl(pl vs r); iitf(trans fun vs f); iibmp(beam pat file;
iiblt(bot loss table); iipl,iitf: 0=none; 1=coh; 2=incoh; 3=both; 4=list
*(3) iipl; iitf; iibmp;  iiblt  nang  ang1  ang2
        4     0      0       0     0     0    90
FREQUENCIES: nfr>0 ==> give f1,f2,f3...; nfr<0 ==> give f1,delf
*(4) nfr  f1  f2(delf)  ...
1  60
-----------------------------------------------------------------------
FFT FILE (iifft), IMPULSE RESPONSE FILE (iiir) options
   [fs=sample f; fmin,fmax=min,max freq of interest in source spec]
*(5) iifft  iiir   nfft      fs    fmin   fmax
         0     0   4196    1000      10    500
-----------------------------------------------------------------------
SOURCE DEPTHS: nzs>0 ==> give zs1,zs2,zs3...; nzs<0 ==> give zs1,delzs
*(6) nzs  zs1  zs2(delzs) ...  (m or ft)
1  100
-----------------------------------------------------------------------
SRC-REC RANGES: nr>0 ==> give r1,r2,r3... in m; nr<0 ==> give r1,delr;
                nr=0 ==> read source track info on next 2 lines
*(7) nr  r1  r2(dr) ...        (km or nmi)
2  1000  2000
*(8) SOURCE TRACK: NLEG(# linear legs); IIPIC (option to plot track)
1  1
LEG SPECIFICATION: V in m/s or kt; NPT>1; T,DT in min;
   CPA,X,Y in km or nmi; PHI in deg
  (for TYPE=1, V=0 ==> T is dist in km; for TYPE=2, V=0 ==> T2 ignored)
     1 IICONT   V   T1   T2  +DT/-NPT    CPA,PHI    DUM,DUM (POLAR FORM)
*(9) 2 IICONT   V   T1  (T2) +DT/-NPT     X1,Y1      X2,Y2    (X-Y FORM)
     2      0  5.0   0    0      15      -40,0       40,0
     2      1  5.0   0    0      15       40,0        0,-40
     2      1  5.0   0    0      15       0,-40       0,40
-----------------------------------------------------------------------
*(10) REC ARRAY SPEC: IIARR(1=uniform,2=nonuniform) (dist in m or ft)
1
IIARR=1: uniformly spaced receiver array
*(11) zr1  nx  ny  nz   dx    dy    dz  [dx,dy,dz are element spacings]
    100.  2   2   2   50.   50.   50.
IIARR=2: receiver array depths. nzr>0 ==> give zr1,zr2,... ;
   nzr<0 ==> give zr1,dz2,dz3... (dz are element spacings, pos down)
*(12) nzr   zr1  zr2(dz2) zr3(dz3) ...
        2  2067    2007
IIARR=2: specify (x,y) receiver positions for each of the nzr depths:
*(13) nxy  x(1),y(1)  x(2),y(2) ... x(nxy),y(nxy) [nzr lines like this]
        2   -2,0        2,0
        2   -2,0        2,0
-----------------------------------------------------------------------
```

Table 2: (continued) Example **.opt** file

```
OCEAN PATH SPECIFICATION: # standard sets; # individual paths
*(14) nstand   nindiv
           1        0
standard sets: range of bottom interactions; ray type: 'r'=refracting,
   'w'=waterborne, 'p'=penetrating, 'a'=all [ex: 0  2  'a']
*(15) kb1  kb2  'type'             [nstand lines like this]
       00   01   'a'
individual paths: initial dir ('u' or 'd'), #t, #b, ray type
*(16) 'ini dir'  #t  #b  'type'    [nindiv lines like this]
         'd'     01  01   'a'
-----------------------------------------------------------------------
BOTTOM PATHS ALLOWED: give maximum # of down-up traversals T in
the NLAY bottom layers on EACH bottom interaction (T usually <= 2)
*(17) T(1)  T(2)  T(3) ... T(NLAY) [NLAY integer numbers]
1  1  1
-----------------------------------------------------------------------
SCATTERING MODEL FOR OCEAN SURFACE, BOTTOM SUBSTRATE: ii, parm
none:              ii=0   parm=0
eckart:            ii=1   parm=wind speed in knots
modified eckart:   ii=2   parm=wind speed in knots
beckmann-spizz:    ii=3   parm=wind speed in knots
uniform:           ii=4   parm=loss in dB per bounce
loss table:        ii=5   parm='file name' in /inp/gama directory
*(18) iisl  slparm    FOR OCEAN SURFACE SCATTERING LOSS
        0       0
*(19) iibl  blparm    FOR BOTTOM SUBSTRATE SCATTERING LOSS
        0       0
-----------------------------------------------------------------------
R-THETA PLOT: iirth (0=no, 1=src angle, 2=rec angle,
   4=MATLAB file of a, R, t [no parameters needed]);
   th1,th2=start, end grazing angles (deg); r1,r2=range interval
*(20) iirth    th1  th2      r1       r2
         0      0   90       0    35000
```

3 Running GAMARAY

The script to run GAMARAY may be found in the file **/usr/fnevan/script/gamarun**. The script is invoked by typing

> **gamarun** <**.svp** file prefix> <**.opt** file prefix> <output file prefix> <output file directory (optional)>.

The output files are placed in the current working directory unless the user supplies the optional fourth parameter.

The easiest way to illustrate the use of the **gamarun** script is by way of example. If one types

> **gamarun abc jjj nnn** ,

then the script will search the user's **/inp/gama** directory for all files of the form **abc*.svp** and of the form **jjj*.opt**, where the * symbol is the UNIX wildcard operator. If the files **abc1.svp**, **abc2.svp**, **jjja.opt**, and **jjjb.opt** exist in the user's **/inp/gama** directory, then GAMARAY will be run for all four combinations of the two **.svp** files with the two **.opt** files.

The output files produced by the GAMARAY run will all have the prefix **nnn**, as given by the third parameter in the **gamarun** call. All GAMARAY runs generate a **.out** file (**nnn.out** in this case) that summarizes the amount of memory allocated, the input parameters used for the run, the CPU time taken for the run, and any warning messages that are given during the run. The names of the other output files have the following suffixes: **.eig** for the text file containing the eigenray list; **.ls** for the text file containing the list of propagation loss data; **_fft** for the binary file containing the receiver transfer functions; **_ir** for the binary file containing the receiver impulse responses; **_pic** for the eigenray picture file; **$bar** for the eigenray bar graph file; and **_rth** for the R–θ plot file.

Since a single **gamarun** invocation can be used to run GAMARAY on multiple input files, the characters replaced by the * wildcard operator in finding the **.svp** and **.opt** files are also included in the output files' names. For example, the **.eig** files for the previous invocation of **gamarun** would be named **nnn1a.eig** (resulting from the input files **abc1.svp** and **jjja.opt**), **nnn2a.eig** (from inputs **abc2.svp** and **jjja.opt**), **nnn1b.eig** (from inputs **abc1.svp** and **jjjb.opt**), and **nnn2b.eig** (from inputs **abc2.svp** and **jjjb.opt**).

In addition, when multiple source depths are specified in the **.opt** file, the two-digit suffix (*i.e.*, 01, 02, and so forth) corresponding to the source depth number is also appended to the output file name roots. For example, if the **.opt** file specified $N_{\mathrm{ZS}} = 2$ and II_{FFT}=1, and the output file prefix is **nnn**, then the two **_fft** output files would be named **nnn01_fft** and **nnn02_fft**.

4 OUTPUT FILES

4.1 The .eig Eigenray List File

The **.eig** file is generated when the *EIG* option is chosen in line (2) of the **.opt** file. An example is shown in Table 3. The first portion of the file is a header that summarizes the input and output file names, some of the options specified in the **.opt** file, and the important features of the environment specified in the **.svp** file. Following the header are lists of eigenrays for each of the source-receiver ranges. For each range, a two-line header giving the source and receiver depths and sound speeds, range, and frequency is given. JZS is the counter for the source depth; JREC is the counter for the receiver number in the array (see Figures 3 and 4). SRC SEQ # is the sequence number of the source along the given track.

The following information is given for each of the eigenrays in the list:

(1) RAY # is the eigenray number.

(2) OCEAN PATH is the path in the ocean taken by the eigenray, given by ID (the initial direction from the source, ‘u’ for up, ‘d’ for down), #T (the number of top or surface interactions), and #B (the number of bottom interactions. (Recall that top and bottom “interactions” include turnings in the top and bottom sections of the ocean also.) The next letter to the right is the ray type [see line (15) of the **.opt** file in Section 2.2].

(3) BOTLAYTRAV gives the path through the bottom taken by the eigenray, characterized by the total number of complete down-up traversals in each of the bottom layers. Space for five bottom layers is given on the first line; when necessary, information for additional layers appears on a second line.

(4) SRC ANGLE and REC ANGLE are the source and receiver grazing angles in degrees. SRC ANGLE is positive for rays that *leave toward* the surface and negative for rays that leave toward the bottom. REC ANGLE is positive for rays that *arrive from* the surface and negative for rays that arrive from the bottom.

(5) TURN SV is the sound velocity at which the eigenray would (or did) turn (become horizontal). The depth at which the ray turned can then be obtained by examining the sound velocity profile. The maximum printed value for TURN SV is 9999.99.

(6) TRAVEL TIME is the eigenray travel time in seconds.

(7) FDEP ATTN is the frequency-dependent attenuation in dB accumulated by the eigenray at the frequency listed in the header. The attenuation at a new frequency

may be calculated by multiplying FDEP ATTN by the ratio of the new frequency to the listed frequency.

(8) REFL/TRAN is the combined reflection and transmission coefficient accumulated by the eigenray. The magnitude MAG in dB and the phase PH in degrees are given.

(9) GEOM LOSS is the geometric spreading loss of the ray in dB.

(10) SURF SCAT is the surface scattering loss in dB at the listed frequency.

(11) SUBS SCAT is the substrate scattering loss in dB at the listed frequency.

(12) TOTAL ATTEN is the magnitude of the total field due to the eigenray in dB.

(13) TOTAL PH is the phase of the total field due to the eigenray in degrees.

(14) # PS is the number of 90° phase shifts applied to the ray field due to grazings of caustic surfaces.

Eigenrays treated with caustic corrections are indicated by a symbol in the first column to the right of the RAY #. An ampersand (&) means that the ray is one of a pair of eigenrays in the insonified zone of the caustic. The total field due to both rays is listed under TOTAL ATTEN and TOT PH for each ray. A dollar sign ($) in the column to the right of the RAY # indicates a shadow zone ray. The ray characteristics given on the first line are those of the caustic ray. The range at which the caustic surface crosses the receiver depth is given on the next line. Also given are the 50-dB extent of the shadow zone and the second range derivative R'' (see Reference 2). Eigenrays having beam displacement are indicated by an underscore character (_) in the second column to the right of RAY #.

At the end of each eigenray list, the following quantities are given: (1) the magnitude and phase of the total coherent propagation loss at the receiver [see Eq. (2.5)], (2) the incoherent propagation loss [see Eq. (2.6)], (3) the difference in travel time between the first and last eigenray listed, and (4) the number of minutes computation time taken.

4.2 The _fft Output File

The **_fft** file is generated when the II_{FFT} option is chosen in line (5) of the **.opt** file and contains the transfer functions for each source-receiver range. The program opens the file with the Fortran statement:

```
open(16,file=test_fft,form='unformatted',organization='stream')
```

The file is generated by the following Fortran unformatted write commands:

Table 3: Example **.eig** eigenray list file.

```
SVP FILE = /usr/fnevan/inp/gama/nosc.svp
OPT FILE = /usr/fnevan/inp/gama/nose.opt
OUTPUT FILE ROOT = /disk5/fnevan/modruns/nosc/nosce
SVP TITLE = nosc#1;  DATE = 08/23/1990;  TIME = 16:32:45
EIG CUTOFF MARGIN =   10.00 dB; IISEG = 1; SVTOL =  0.000; RTERP =   200; IICAUS = 1; IIBD = 0
SURFACE SCAT = NO ; WIND =   0.0 KT
SUBSTRATE SCAT = NO ; WIND =   0.0 KT
# OCEAN PATHS =  14; MAX # TRAV IN BOTTOM LAYERS =  2  2  2
NZS =  1; NZR =  1; # REC =  1; # SRC SEQ =    4; WATER DEPTH =   2067.00 M; SED THICK =   900.00 M
SOUND SPEEDS (M/S): OCEAN SURF = 1492.24; OCEAN BOT = 1497.10
                    SED SURF = 1499.00; SED BOT = 2738.00; SUBSTRATE = 6000.00
IIFFT = 0; IIIR = 0; NFFT =      4; FS =    153.60 HZ
DF =   38.400 HZ; FMIN,FMAX =     10.     50. HZ; TIME WINDOW DURATION = 999.9900 SEC


        ZS =  100.00 M; ZR = 2067.00 M; RANGE =   1000.00 M; T/S =   1000.00; JZS =  1; JREC =  1
           SRC SEQ # =    1; CS = 1491.090 M/S; CR = 1497.100 M/S; FREQUENCY =     8.75 Hz

 RAY OCEAN PATH  BOTLAYTRAV   SRC    REC   TURN    TRAVEL  FDEP REFL/TRAN GEOM SURF SUBS   TOTAL  TOT  #
  #   ID #T #B    1 2 3 4 5  ANGLE  ANGLE   SV    TIME (S) ATTN  MAG   PH LOSS SCAT SCAT   ATTEN   PH PS
---- ----------- ---------- ------ ------ ------ --------- ---- ---- ---- ---- ---- ---- ------- ---- --
   1   d  0  0 w  0         -63.15  63.03 3301.2   1.47487    0    0    0  -67    0    0  -66.89  -34  0
   2   u  1  0 w  0          65.31  65.20 3569.8   1.59561    0    0  180  -68    0    0  -67.57  166  0
COHERENT PL =  -76.07 dB; PHASE = -102 deg; INCOH PL =  -64.21 dB; DTAU =   0.121 s; CP MIN =    0:08
EOD -----------------------------------------------------------------------------------------------------

                                                 ⋮

        ZS =  100.00 M; ZR = 2067.00 M; RANGE =  12000.00 M; T/S =  12000.00; JZS =  1; JREC =  1
           SRC SEQ # =    4; CS = 1491.090 M/S; CR = 1497.100 M/S; FREQUENCY =     8.75 Hz

 RAY OCEAN PATH  BOTLAYTRAV   SRC    REC   TURN    TRAVEL  FDEP REFL/TRAN GEOM SURF SUBS   TOTAL  TOT  #
  #   ID #T #B    1 2 3 4 5  ANGLE  ANGLE   SV    TIME (S) ATTN  MAG   PH LOSS SCAT SCAT   ATTEN   PH PS
---- ----------- ---------- ------ ------ ------ --------- ---- ---- ---- ---- ---- ---- ------- ---- --
   1   d  0  0 w  0         -10.46   9.13 1516.3   8.12725    0    0    0  -82    0    0  -82.16   41  0
   2   u  1  0 w  0          11.21   9.98 1520.1   8.15233    0    0  180  -82    0    0  -82.05  -60  0
   3   d  0  1 a  1         -10.65  -9.34 1517.2   8.12861    0    0  -90  -82    0    0  -82.51  -45  1
   4   u  1  1 a  1          11.44 -10.24 1521.3   8.15408    0    0   90  -82    0    0  -82.39 -145  1
   5   d  1  1 a  1         -29.35  28.93 1710.6   9.04392   -1    0   90  -82    0    0  -83.30  138  1
   6   u  2  1 a  1          30.21  29.81 1725.4   9.11057   -1    0  -90  -82    0    0  -83.39  168  1
   7   d  1  2 a  2         -32.47 -32.10 1767.4   9.10403   -3   -1    0  -81    0    0  -84.48 -122  2
   8   u  2  2 a  2          33.62 -33.27 1790.6   9.17703   -3   -1 -180  -81    0    0  -84.61  -72  2
COHERENT PL =  -75.53 dB; PHASE = -104 deg; INCOH PL =  -73.98 dB; DTAU =   1.050 s; CP MIN =    0:09
EOD -----------------------------------------------------------------------------------------------------
```

```
write(16) XH(1:20)
write(16) TF(1:NFFT/2 + 1)
```

where XH is a real∗4 header and TF is the complex∗8 array containing the transfer function. Thus, for each source-receiver range, the **_fft** file contains a 20-word header and an ($N_{\mathrm{FFT}} + 2$)-word transfer function (containing $N_{\mathrm{FFT}}/2 + 1$ complex numbers). The header contains the following information:

XH(1)	=	T_{MIN}
XH(2)	=	A/D run number (unused, set to 0)
XH(3)	=	source sequence number
XH(4)	=	receiver number
XH(5)	=	N_{FFT}
XH(6)	=	f_s
XH(7)	=	f_0 (set to 0)
XH(8)	=	quality mask (unused, set to 0)
XH(9)	=	percentage overlap (unused, set to 0)
XH(10)	=	number of frequency bins in FFT (set to $N_{\mathrm{FFT}}/2 + 1$)
XH(11)	=	number of receivers, N_{REC}
XH(12)	=	number of FFTs/channel/PRI (unused, set to 1)
XH(13)	=	Δ_x
XH(14)	=	Δ_y
XH(15)	=	ZR
XH(16)	=	c_s
XH(17)	=	c_r
XH(18)	=	range
XH(19)	=	ZS
XH(20)	=	total number of source sequences in file

Here, T_{MIN} is the reference time for the transfer function: if one inverse FFT'ed the transfer function, one would get an impulse response that began at time T_{MIN} relative to an impulse at time zero at the source. T_{MIN} is chosen so that the first eigenray arrives 4% of the way into the time window. The variable f_0 is the starting frequency of the FFT and is always set to 0. The variables Δ_x and Δ_y are the receiver positions relative to the origin shown in Figures 3 and 4. The variables c_s and c_r are the sound speeds at the source and receiver, respectively. The variable in XH(10) is the number of frequencies included in the FFT.

An FFT with its header is written for each source-receiver range. The order in which the FFTs appear is as follows: for each source position along the source track, an FFT is output for each of the N_{REC} receivers in the array. The receivers in the array are numbered as indicated in Figures 3 and 4. For each source position, the reference time T_{MIN} is the same for all receivers. As a consequence, one can perform crosscorrelations between receivers without the need to time delay the FFT files.

4.3 The _ir Impulse Response File

The **_ir** file is generated when the II_{IR} option is chosen in line (5) of the **.opt** file and contains the impulse responses for each source-receiver range. It is opened with the command:

```
open(15,file=test_ir,form=‘unformatted’,organization=‘stream’)
```

A single header is written at the beginning of the file:

```
write(15) TITLE(1:8)
write(15) KHEAD(2:50)
```

followed by one real∗4 N_{FFT}-word impulse response for each source-receiver range:

```
write(15) IR(1:NFFT)   .
```

The order in which the impulse responses appear is the same as that for the FFTs in the **_fft** file described above.

The character array TITLE contains the first 8 characters of the **.svp** file title, and the integer array KHEAD contains the following values:

KHEAD(2)	=	source sequence number
KHEAD(3)	=	N_{FFT}
KHEAD(4)	=	f_s
KHEAD(5)	=	T_{MIN}
KHEAD(6)	=	number of receivers, N_{REC}
KHEAD(6+j)	=	ZR_j, $j = 1, 2, \ldots N_{\mathrm{REC}}$
KHEAD(29)	=	ZS
KHEAD(29+j)	=	$\Delta_x(j)$, $j = 1, 2, \ldots N_{\mathrm{REC}}$

The real variables are placed in KHEAD by equivalencing a real variable XHEAD to KHEAD and setting the corresponding entry of XHEAD to the variable.

Note: The header for the **_ir** file was inherited from previous versions of the GAMARAY program. Due to the fact that the header cannot describe three-dimensional arrays (Δ_y is not included in the header), it is advised to take the inverse FFT of the transfer function in the **_fft** file to get the impulse response instead of using the II_{IR} option to generate the **_ir** file.

4.4 The .bmp Beam Pattern File

The **.bmp** file is generated when the II_{BMP} option is chosen in line (3) of the **.opt** file. The **.bmp** file was designed to allow the user to take the eigenrays that make up the total field at a receiver and apply a beam pattern to the source and/or receiver. It is a binary file created by the following open statement and unformatted write statements:

```
    open(18,file=‘test.bmp’,form=‘unformatted’,organization=‘stream’)
    write(18) NZS,NZR,NR,NFR,FREQ(1:NFR)
    do 10 JZS=1,NZS
       write(18) ZS
       do 20 JR=1,NR
          do 25 JZR=1,NZR
             write(18) ZR,R,PHI,NRAY
             do 30 JRAY=1,NRAY
                write(18) THS,THR,CPL(1:NFR)
30           continue
25        continue
20     continue
10  continue
```

Here, PHI is the angle in degrees in the x-y plane between the positive x-axis and the line from the receiver to the source [only applicable when a source track is specified (see Figures 1 and 2), THS is the eigenray grazing angle in degrees at the source (positive for rays *leaving toward* the surface), THR is the eigenray grazing angle in degrees at the receiver (positive for rays *arriving from* the surface), and CPL is the complex field due to the eigenray. All quantities are real∗4 or complex∗8 variables.

4.5 Bottom Loss Output (.blt) and Input files

A **.blt** output file is generated when the II_{BLT} option is chosen in line (3) of the **.opt** file. Bottom loss is calculated for the angles given in line (3) and the frequencies given in line (4). A bottom loss input file is read when II_{SL} or II_{BL} is set to 5 in line (18) or (19) of the **.opt** file. The format for both types of files is as follows:

```
*(1) # angles, na; # frequencies, nf
2  2
*(2) list of frequencies 10  50
*(3) angle  loss,phase  loss,phase...[nf pairs] [na lines like this]
10  2,0  4,0 20  3,0  5,0
```

The asterisks in the file have the same significance as in the **.svp** and **.opt** files described in the beginning of Section 2. The bottom loss magnitude is given in positive dB, the phase in degrees. When using a table, GAMARAY does a bilinear interpolation to calculate the bottom loss at a given frequency and grazing angle. Care must be taken that the bottom loss phases do not change by more than 180° between sample points (the program *does* track the phase if it changes from +180° to −180° or from 0° to 360°).

5 THEORETICAL CONSIDERATIONS

The theory upon which the GAMARAY model is based is described in References 1 and 2. Some of the considerations the user should take into account are described here.

5.1 Caustic Corrections

The user has the option in line (2) of the **.opt** file to include caustic corrections in the eigenray calculations. A caustic is defined as a point in the ocean environment where a ray bundle emitted by the source is focused such that its cross sectional area vanishes. Caustics most often arise when rays refract (turn) in a medium whose sound velocity is increasing in the direction the ray is travelling. Caustics also arise near critical angles when beam displacement is included in the ray trajectories. On one side of the caustic surface (the "insonified zone"), two rays with similar angles and paths arrive at a receiver; on the other side (the "shadow zone") no rays arrive with the given path arrive at a receiver. The ray amplitude according to classical ray theory is infinite at the caustic, anomalously large in the insonified zone, and zero in the shadow zone. The caustic corrections used in GAMARAY are summarized in Reference 2 and are based on those given in Reference 3. They correct classical ray theory for receivers in caustic regions, yielding finite ray amplitudes in the insonified zone and exponentially decaying ray amplitudes in the shadow zone. Caustic corrections are frequency dependent and are more important at lower frequencies.

When a caustic is found by GAMARAY, the 40-dB shadow zone extent in wavelengths λ is calculated for the lowest frequency of interest in the problem. If the shadow zone extent is larger than 500λ, the caustic is counted as a false caustic and the shadow zone field is not computed. A message is written to the **.out** file that includes the source grazing angle, the sound velocity at which the caustic ray turns, the caustic range, and the shadow zone extent. Usually, the false caustic is due to an irregularity in the SVP at the depth where the caustic ray turns. The user may want to try to smooth out the SVP in the area in question if it appears that the false caustic is giving anomalous results.

It is advised to select the caustic correction option when running GAMARAY. The computation time required to incorporate the caustic corrections is usually not large, and the potential exists for significant anomalies when a receiver lies near a caustic.

5.2 Sound Velocity Profile Segmentation

The GAMARAY model allows the user several options for constructing the sound velocity profile in the ocean and in the bottom. For the ocean profile, a list of (depth, sound velocity) pairs are given in line (4) of the **.svp** file. The user may then specify in line (1) of the **.opt** file that these points be connected by simple straight-line segments or by more complicated curved segments. Linear segmentation of the SVP can cause problems in cases where eigenrays refract near the depth where the sound velocity gradient is discontinuous. False caustics may be formed, and the field calculated may be erroneous, even with caustic corrections. Thus, linear segmentation should not be used when source and receiver are both in a sound velocity duct.

The curved segments are designed to preserve continuity of the sound speed gradient at the depths where the sound velocity is given.[4] The beneficial result is that the R–θ function (see Reference 1) is continuous for refracting rays. However, even with curved segmentation, an irregular sound velocity profile can still generate false caustics. When refracted eigenrays are important, successful modeling using GAMARAY requires the user to input a sufficiently smooth sound velocity profile. On the other hand, curved segmentation should only be used when necessary because the ray tracing requires more computation time.

In the layers of the ocean bottom, the user may choose either a linear or a “BLUG” profile of the compressional sound velocity. The BLUG profile (named for the Bottom Loss UpGrade program) is a three-parameter, curved profile that was designed to be typical of ocean bottom sediment layers (see Reference 5). The various options for specifying a BLUG layer are given in line (6) of the **.svp** file.

The provision has been made to construct multiple BLUG profile layers that preserve continuity of the sound velocity gradient [see item(6)(xiii) in the **.svp** file]. The usefulness of this capability is that more complicated sound velocity profiles may be constructed without introducing false caustic-generating discontinuities in the SVP. The user may also use this capability to specify a piecewise linear attenuation profile in the ocean bottom (attenuation often initially increases and then decreases as a function of depth).

5.3 Interface Scattering Models

The scattering models allowed by GAMARAY in lines (18) and (19) of the **.opt** file are discussed in Reference 6. The Eckart model is based on the following equation for the linear loss factor L (loss in dB $= 20 \log L$) in terms of the frequency f and the ray grazing angle θ:

$$L_E = e^{-2g} \quad , \tag{5.1}$$

where

$$g = (2\pi f \sigma \sin\theta / c)^2 \quad , \tag{5.2}$$

σ is the rms displacement of the rough surface and c is the sound speed. The roughness parameter σ in meters is obtained from the wind speed w in knots via the Pierson-Moskowitz relation

$$\sigma = 1.41 \times 10^{-3}\, w^2 \quad . \tag{5.3}$$

The modified Eckart model is governed by the following equation:

$$L_{\mathrm{ME}} = \max[I_0(2g)\, e^{-2g}, 0.282] \quad , \tag{5.4}$$

where I_0 is the zero-order modified Bessel function and the maximum loss per reflection is $20\log(0.282) = 11$dB. The Beckmann-Spizzichino model is based on a composite model adopted in the Generic Sonar Model (see Reference 7):

$$L_{\mathrm{BS}} = \left(0.3 + \frac{0.7}{1 + 0.01A^2}\right)\sqrt{1 - \min[0.99, \max(V, 0.5\sin\theta)]} \quad , \tag{5.5}$$

where

$$A = 1.16 \times 10^{-5}\, w^2 f \tag{5.6}$$

$$V = \sin\theta - \frac{\sin\theta}{\theta\sqrt{T\pi}}\, \exp(-0.25T\theta^2) \tag{5.7}$$

$$T = \frac{500}{3 + 2.6w} \quad . \tag{5.8}$$

5.4 Beam Displacement

The user has the option in line (2) of the **.opt** file to include the phenomenon of beam displacement in the ray trajectories. Beam displacement, or, more accurately, ray displacement occurs when a ray is totally reflected at an interface. Rays that experience beam displacement differ from the specularly reflected rays of classical ray theory in that they travel along the interface before being reflected. There are two types of beam displaced rays: reflected rays and lateral wave rays (see Figure 7).

Reflected rays have a relatively small displacement and differ from the classical rays by a small phase shift. It is only in shallow water, where many rays with many bottom bounces make up the field, that beam displacement is important to include. The lateral wave ray, which is not predicted by classical ray theory, represents the wave that strikes the interface at the critical angle, travels along the interface, and is continuously reradiated into the upper medium. Lateral waves are more important when source and receiver are close (in terms of wavelengths) to the interface.

The phenomenon of beam displacement is frequency dependent, *i.e.*, the angles, travel times, and magnitudes of eigenrays with beam displacement depend on frequency. When the BD option is chosen, GAMARAY chooses a set of frequencies at which to compute the eigenray characteristics. When an eigenray with displacement is found, GAMARAY goes back and finds the same ray at all the frequencies in the set. In computing the frequency-dependent field due to the eigenray, the ray characteristics are then linearly interpolated as a function of frequency. Due to the fact that rays with displacement must be found at a number of frequencies (typically three to five), computation times for broadband,

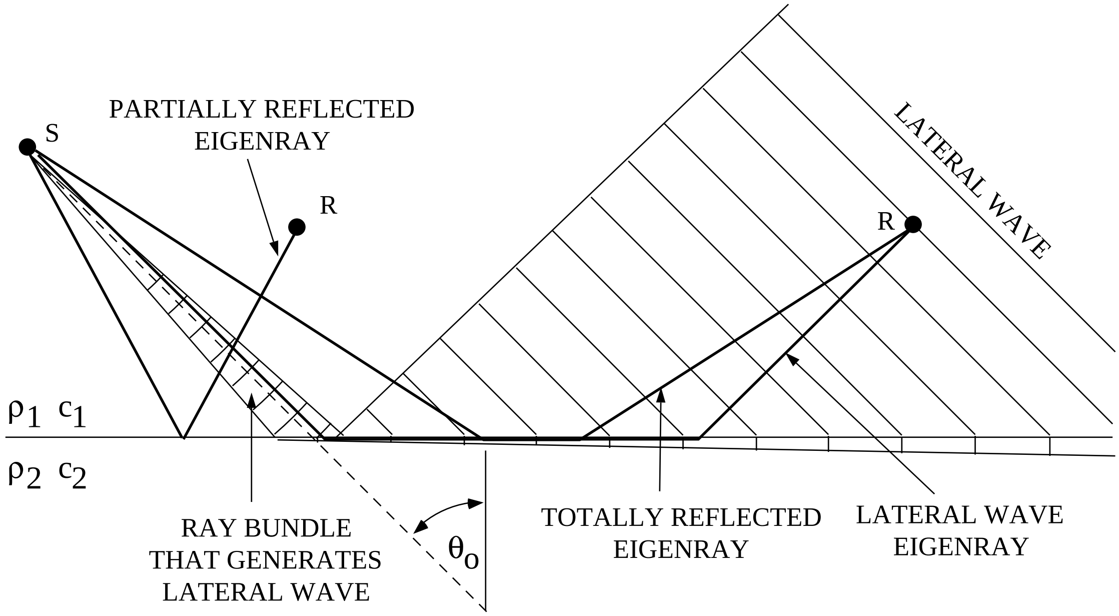
PARTIALLY REFLECTED EIGENRAY
S
R
R
LATERAL WAVE
ρ_1 c_1
ρ_2 c_2
RAY BUNDLE THAT GENERATES LATERAL WAVE
θ_0
TOTALLY REFLECTED EIGENRAY
LATERAL WAVE EIGENRAY

Figure 7: Eigenray picture for the reflected field, showing the generation of the lateral wave by the ray bundle incident on the interface at the critical angle $\theta_{\rm cr} = \arcsin(c_1/c_2)$

shallow-water problems can be relatively large. In summary, it is recommended that beam displacement only be used by the "experienced" user and then, only in low-frequency, shallow water environments where many totally reflected rays make up the field.

5.5 Eigenrays With Complex Bottom Paths

When eigenrays interact with a layered bottom, it is most convenient to characterize their paths by the total number of traversals of each layer. The issue of complex bottom paths is described in detail in Sec. IV of Reference 1. GAMARAY characterizes eigenrays by their "ray family," defined by the ray's initial direction, the number of top interactions N_T, the number of bottom interactions $T(0)$, and the number of down-up traversals in each of the NLAY bottom layers $T(j), j = 1, 2, ...NLAY$. These parameters appear in the **.eig** eigenray listing output file in Table 3. There may be many different eigenrays belonging to the same family but taking different physical paths (for example, see the ray family $T(1) = 2$, $T(2) = 2$ in Figure 6). All eigenrays in the same family have the same travel times, spreading losses, and angles, but they may have different reflection/transmission coefficients. GAMARAY counts all eigenrays in the same family as a single eigenray by computing a single reflection/transmission coefficient that is the sum of the coefficients of all the eigenrays in the family. The expression [equivalent to Eq. (18) of Reference 1] for the ray family reflection/transmission coefficient C_F is:

$$C_F = (-1)^{N_T} R_{N,N+1}^{b_N} \prod_{n=0}^{N-1} S_{n,n+1} \tag{5.9}$$

$$\times \sum_{j=1}^{\bar{b}_n} C_j^{b_n} \, C_{j-1}^{b_{n+1}-1} \, (R_{n,n+1} R_{n+1,n})^{\bar{b}_n - 1} \, (T_{n,n+1} T_{n+1,n})^j \quad , \tag{5.10}$$

where

$$S_{n,n+1} = \begin{cases} R_{n,n+1}^{b_n - b_{n+1}} & b_n \geq b_{n+1} \\ R_{n+1,n}^{b_{n+1} - b_n} & b_n < b_{n+1} \end{cases} \quad , \tag{5.11}$$

$$C_k^n = \frac{n!}{k!(n-k)!} \quad , \tag{5.12}$$

$$\bar{b} = \min(b_n, b_{n+1}) \quad . \tag{5.13}$$

In addition, N is the deepest layer of penetration of the ray, (-1) is the reflection coefficient at the surface, $R_{n,n+1}$ and $T_{n,n+1}$ are the reflection and transmission coefficients from the n'th to the $(n+1)$'st layers, respectively, and $b_n = T(n)$ is the number of down-up traversals of the j'th bottom layer.

5.6 Use of R–θ Plots

The user has the option in line (20) of the **.opt** file to generate R–θ plots, which are equivalent to the R–a diagrams (where $a = \cos\theta/c$ is the Snell invariant) discussed in References 1 and 2. R–θ functions are constructed by launching rays from the source at grazing angles θ and plotting the horizontal ranges R at which they cross the receiver depth z_r. When the II_{RTH} option is chosen, GAMARAY plots a single R–θ function for each complete ray path specified by the user. (As discussed in Section 5.5 and Reference 1, complete ray paths are characterized by the number of traversals of the ocean layers and the bottom layers.) Eigenrays at a particular source-receiver range R_0 are identified by finding the intersections of the R–θ functions with the horizontal line at $R = R_0$. Local minima or maxima in the R–θ functions indicate caustics. For example, if a minimum occurs at the point (R_c, θ_c), then a ray bundle leaving the source at angle θ_c and taking the given ray path is focused at range R_c and receiver depth z_r. Receivers at ranges $R > R_c$ lie in the insonified zone, while those at ranges $R < R_c$ lie in the shadow zone of the caustic.

R–θ plots are useful for understanding the eigenray structure at a particular receiver depth. One can immediately see which eigenrays arrive at receivers as a function of range and at which angles they leave the source. R–θ plots are also useful in identifying the cause of anomalous features in propagation loss curves at particular ranges. One important characteristic that the R–θ plots do not indicate is the relative strengths of the eigenrays. Since GAMARAY places all the R–θ functions corresponding to the different ray paths on the same plot, it is wise to limit the number of ocean paths and bottom paths specified in the **.opt** file.

6 HINTS FOR RUNNING GAMARAY

6.1 Preparing the .svp File

When applying the GAMARAY model to a particular acoustic propagation problem, the first step is to create the **.svp** file appropriate for the area being modeled. The sound velocity profile in the water column may be obtained from a database or from measurements. The geoacoustic profile of the ocean bottom may be obtained from the BLUG database, from seismic studies of the area, or from the literature. A classic reference for geoacoustic modeling appears as Reference 8. Alternatively, the ocean bottom may be best modeled by a bottom loss table obtained from measurements. In such cases one sets the number of bottom layers to zero in line (5) of the **.svp** file and makes the substrate a perfectly reflecting halfspace [by setting $CP = RHO = 10000$, for example, in line (7)]. The bottom loss file is specified in line (19) of the **.opt** file by setting $II_{\mathrm{BL}} = 5$.

Before finalizing the **.svp** file, it is worthwhile to study a plot of the SVP and eliminate points that are not needed. For example, measured SVPs can have closely spaced points that are nearly linear. Eliminating some of the points will save computation time but will have no effect on the calculations.

The BLUG database[5] for ocean bottom geoacoustic parameters was compiled by fitting ray model calculations to experimental measurements of bottom loss. Bottom loss was obtained by detonating shots and frequency-averaging the received spectra. The database provides the following geoacoustic parameters for the sediment layer: (1) the ratio of the sound speeds at the sediment surface and the water above it, $c_{\mathrm{sed}}/c_{\mathrm{water}}$, (2) the surface density RHO_1 (assume $RHO_2 = RHO_1$ if no other information is available), (3) the initial sound velocity gradient g, (4) the curvature parameter β, (5) the surface compressional wave attenuation KP_1, (6) the attenuation gradient, from which one may obtain KP_2, and (7) the basement reflection coefficient R_{BLUG}. The sediment layer thickness H is not given and must be obtained from the two-way travel time τ_2. An approximate formula to convert τ_2 to thickness H is (see Sec. IE of Reference 8):

$$H = c_{\mathrm{sed}} \left(e^{\bar{g}\tau_2/2} - 1 \right) / \bar{g} \quad , \tag{6.1}$$

where $\bar{g}$ is the average gradient in the layer. Shear speeds and attenuations are not available from the database.

Also given in the BLUG description of the ocean bottom are the thickness and density of a thin layer above the sediment. Except for the density, the thin layer is assumed to have the same parameters as the surface of the sediment layer. The thin layer was included in the database to account for anomalously low bottom losses observed at certain sites and at certain (steep) angles and (high) frequencies. The density of the thin layer can be unrealistically high. The thin layer should be ignored when its thickness

H_{TL} is less than a fraction of the smallest wavelength of interest λ_{min}: $H_{\mathrm{TL}} < \lambda_{\mathrm{min}}/6$, or, equivalently when the maximum frequency of interest $f_{\mathrm{max}} < c_{\mathrm{sed}}/(6H_{\mathrm{TL}})$. When the thin layer *is* included in the geoacoustic profile, care should be taken that the eigenrays it generates have realistic characteristics. (An example of an unrealistic eigenray would be one that travels large distances horizontally in the isospeed thin layer.)

Reflection at the substrate interface may be treated in one of three ways. The first way is to use the angle-independent R_{BLUG} from the BLUG database, in which case the geoacoustic parameters given for the substrate are ignored. The second way is to set $R_{\mathrm{BLUG}} = 0$ and estimate the geoacoustic parameters of the substrate, in which case the plane-wave reflection coefficient is used. The third way is the same as the second, except in addition to the reflection loss, an angle- and frequency-dependent substrate scattering loss is included by way of a scattering loss table (see Section 4.5) or an analytic model (see Section 5.3).

GAMARAY allows multiple BLUG layers to be specified in the ocean bottom. In order to create a bottom layer with a more complicated curved profile, one may divide it up into several separate layers and use the $\beta = -999$ option described in item (6)(xiii) of the **.svp** file. The top BLUG layer is specified by c_1, c_2, and either β or g in the usual way. For subsequent layers, c_1 must be identical to the c_2 of the previous layer, and β must be set to -999. GAMARAY then computes the β necessary for the sound speed gradient to be continuous. The density must be continuous across the layer interfaces, but piecewise linear profiles of density and attenuation may be specified.

6.2 Preparing the .opt File

Before running GAMARAY on a large problem, the user should first become familiar with the eigenray structure of the environment. For example, if the problem consists of a 300-point source track and a 16-element vertical array, one should first run a test case for a 10-point source track and a single receiver. Eigenray lists and pictures should be examined to get an idea of which eigenrays are important at the ranges of interest. (In general, more eigenrays are important at longer ranges.) If it is not clear why certain rays do not appear in the eigenray list, the CUT_{DB} parameter may be increased to include more rays. In general, the user should decrease CUT_{DB} to 25–40 dB for large runs in order to avoid the unnecessary calculation of weak eigenrays.

For the test run, the maximum number of ocean path bottom bounces [line (15) in the **.opt** file] should be set to a high number so that all significant eigenrays are sure to be found. A message in the **.out** file informs the user when the program has ignored rays with more than a certain number of bottom bounces because they are too weak.

Another point regarding the ocean path specification concerns the possibility of inadvertently ignoring significant ray paths. When source and receiver are both near the surface, the classification of ray paths by number of bottom bounces [as in line (15) of the **.opt** file] is convenient and natural because the four eigenray multipaths with a given number of bottom bounces have very similar paths [see Figure 8(a)]. But when source and receiver are both near the bottom [see Figure 8(b)], the natural classification scheme is by number of surface reflections. Finally, when the source is near the surface and the receiver is near the bottom (or *vice versa*), the four similar multipaths are divided still differently [see Figure 8(c)]. A common mistake in the latter case is to specify ocean paths with 0 and 1 bottom bounce, which would include only two of the four multipaths that traverse the ocean three times [see Figure 8(d)]. If the **'d' 1 1** ray is significant, then the **'d' 1 2** may also be significant. The correct way to specify the ocean paths would be to add the two individual paths **'d' 1 2** and **'u' 2 2** in line (16) of the **.opt** file.

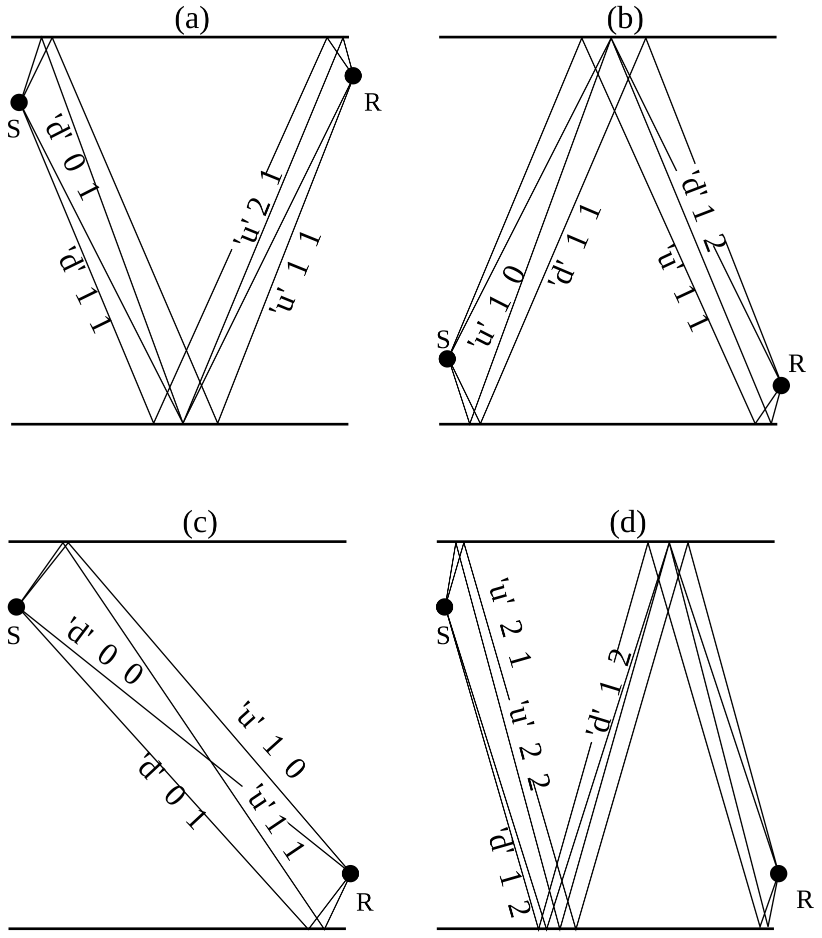

Figure 8: Eigenray multipaths for various source-receiver geometries: (a) both near the surface, (b) both near the bottom, and (c) source near surface, receiver near bottom. Part (d) shows the four multipaths with three traversals of the ocean for the geometry in part (c).

7 APPLICABILITY AND LIMITATIONS

When using the GAMARAY propagation model, it is important to keep in mind its applicability and limitations. Like any model, GAMARAY is accurate and efficient for some environments and problems, but inaccurate and/or inefficient for others. The following sections address the issue of determining when GAMARAY may be used with confidence, when it may be used with some caution, and when it should not be used at all.

7.1 Short vs. Long Range Propagation Problems

In the context of ray theory modeling, the most convenient way to define short and long range and deep and shallow water is in terms of the ratio R/h, where R is the horizontal source-receiver range, and h is the water depth. We define short range (deep water) to be cases where $R/h \leq 5$ and long range (shallow water) to be cases where $R/h \geq 10$. The region $5 < R/h < 10$ is a transition region between the two regimes.

At short ranges the significant eigenrays are usually those with two or fewer bottom interactions. Eigenrays strike the bottom at relatively steep angles and experience relatively high loss upon each interaction. The reflected rays are only partially reflected, and the transmitted rays have long paths and high frequency-dependent attenuation in the bottom. In addition to the bottom loss, eigenrays with more bottom interactions also have considerably higher geometric spreading losses. Since most of the distance traveled is vertical, additional traversals of the ocean add significantly to the ray's total path length.

In contrast, at long ranges the eigenray field is composed of many rays with similar magnitudes. When both source and receiver are located in a duct, rays at long ranges may not interact with the bottom at all. When rays do interact with the bottom, the interactions occur at shallow angles and result in little loss. They may undergo either total reflection (off a sand bottom, for example) or near-total penetration and shallow turning (in a sediment bottom, for example). Because the rays are traveling relatively great distances horizontally and short distances vertically, additional bottom interactions do not add appreciably to the eigenrays' geometric spreading losses.

In general, GAMARAY is better suited to short range problems than to long range problems. In terms of efficiency, short range cases are almost always more efficient than long range cases because the number of eigenrays increases as the source-receiver range increases. In terms of accuracy, GAMARAY is almost always accurate for short range cases, but may or may not be accurate for long range cases. Several specific examples of long range propagation are given in the following sections.

7.2 Shallow Water

We define shallow water to have depths $h < 200$ m. In shallow water, the "long range" regime, as defined in the previous section, begins at relatively short ranges (ten water depths for $h = 100$ m is just 1 km). The sound speed gradient in the water column is usually small, the bottom usually has a higher sound speed than the water, and bottom interaction is extremely important to the propagation characteristics. In terms of ray modeling, shallow water problems are difficult because there are often an exorbitantly high number of eigenrays to calculate. When the ocean bottom has a layered structure the number of possible eigenrays can rise explosively.

In the simplest of shallow water environments, where the bottom is a fluid halfspace with sound speed c_b and the water sound speed c_w is isovelocity (the "Pekeris model"), GAMARAY gives reasonably accurate results. The beam displacement and caustic correction options must be chosen. The number of significant eigenrays at a particular range may be approximated as follows. All rays that reflect from the bottom at angles steeper than the critical angle $\theta_{\rm cr} = \arccos(c_w/c_b)$ are not significant due to their large number of partial reflections. Using simple geometry and assuming straight-line ray paths, one can estimate the number of rays as

$$N_{\rm RAY} = 2(R/h)\tan\theta_{\rm cr} \quad . \tag{7.1}$$

Table 4 gives the number of rays in terms of the ratio R/h for various bottom sound speeds and critical angles. For a given water depth h, $N_{\rm RAY}$ increases linearly with range R. One can see that for hard bottoms and long ranges, the number of rays can get very large.

Table 4: APPROXIMATE NUMBER OF EIGENRAYS $N_{\rm RAY}$ IN ISOSPEED SHALLOW WATER AS A FUNCTION OF BOTTOM SOUND SPEED c_b. $N_{\rm RAY}$ IS EXPRESSED IN TERMS OF THE RATIO OF THE HORIZONTAL RANGE TO THE WATER DEPTH R/h.

c_b	$\theta_{\rm cr}$	$N_{\rm RAY}$
1875 m/s	36.9°	$1.5R/h$
1678 m/s	26.6°	$1.0R/h$
1546 m/s	14.0°	$0.5R/h$
1510 m/s	7.1°	$0.25R/h$

When a shallow water environment also contains layers, the number of possible eigenrays grows enormously. GAMARAY attempts to avoid unnecessary calculations by keeping track of weak eigenrays, but usually there are still a large number of eigenrays with approximately the same strength that interfere with one another. The user should, when tackling such an environment, run a series of test cases that gradually increase in complexity . For example, instead of starting with an 8-receiver, 50-source-point run, one should start with a single receiver and the shortest source-receiver range. If the computations require a reasonable amount of time for that case, one may then attempt longer and longer ranges, still one source-receiver range at a time. Only when the user has a good estimate of the time required should the whole run be made.

7.3 Deep Water

We define deep water to have depths $h > 1000$ m. In deep water, the sound velocity profile plays an important role in the propagation, especially at long ranges. The sound velocity often has a local minimum near the surface, occasionally has a second local minimum in the mid-water region, and almost always increases monotonically toward the bottom. In contrast to shallow water environments, the depths of the source and receiver in deep water play an important role in determining the propagation characteristics. When both source and receiver lie close to the same relative minimum in the sound speed, a "duct" is formed, and there may be many eigenrays with different numbers of turnings above and below the source and receiver. When one sensor is near the surface and the other is near the bottom, a different eigenray structure is typically observed. The direct path eigenray exists from zero range out to a range called "RAP" (Reliable Acoustic Path) range. At RAP range the the direct eigenray turns at the near-bottom sensor depth; beyond RAP range the first eigenray arrival is either a bottom-penetrating ray or a ray that traverses the ocean three times, rather than just once.

In deep water, as in shallow water, GAMARAY is more efficient at short ranges than at long ranges because fewer significant eigenrays exist. The accuracy of GAMARAY in predicting ducted propagation will be discussed in the following section. It should be accurate at all ranges in cases where one sensor is near the bottom.

7.4 Ducted Propagation

Ducted propagation occurs when both source and receiver are located at or near the same relative minimum in the sound velocity profile. Ray theory calculations can be inaccurate in such cases because there exist large numbers of eigenrays that travel nearly horizontally and form complicated caustic structures. (If source and receiver are exactly at the sound velocity minimum, there are, theoretically, an infinite number of eigenrays.) Also, ray theory does not account for energy that "leaks" from one duct to another duct.

At short ranges GAMARAY should provide fairly accurate results for ducted propagation. In terms of modeled impulse responses, the eigenray arrivals that have traveled outside the duct should be accurate, while those that have traveled within the duct are somewhat suspect. Since the non-ducted eigenray arrivals are usually separated in the time domain from the ducted arrivals, GAMARAY can still be useful when one is interested in the effect of the bottom on a transient, broadband signal.

At long ranges GAMARAY should predict a reasonably accurate level for the propagation loss curve, but probably will not give perfect agreement for the interference pattern in the curve. If the sound velocity profile has *two* ducts, GAMARAY will not predict the leakage of low frequency energy out of duct in which the source is located. The option for caustic corrections should always be chosen for ducted propagation. False caustic warning messages are common when GAMARAY is run for ducted propagation. In some cases, smoothing the SVP and using curved segmentation can reduce the number of false caustics.

7.5 Range-Varying Environments

Since GAMARAY assumes a range-invariant environment, one cannot apply the model to problems with significant range variation. For cases where the ocean sound velocity profile changes, propagation over short ranges and at steep angles may be unaffected, but propagation over long ranges by way of rays that refract in the water column will be affected greatly. For cases where the water depth varies, a total change of 10 or 20 percent will affect the propagation significantly. The eigenray arrival structure that appears in a modeled impulse response is usually less sensitive to range variation than modeled propagation loss curves.

7.6 Problematic Eigenrays

The user of GAMARAY should be aware that ray theory calculations are not correct in certain situations. One such situation is when a ray turns near the ocean surface (or any other interface). For a given source and receiver depth, there is a particular range at which the eigenray turns at the surface. At shorter ranges there are two eigenrays: one that turns in the water column and one that reflects off the ocean surface. At longer ranges neither eigenray exists. The result is that there is a discontinuity in the field computation at the transition range. The same phenomenon occurs when a ray turns at a local maximum in the sound velocity profile (SVP) and crosses the receiver depth at a range R_0. A slightly steeper ray penetrates across the SVP maximum and crosses the receiver depth at a longer range discontinuous from R_0.

The previous paragraph deals with a problem that is a subset of a more general problem with ray theory calculations: Ray calculations are made along the ray path itself and do not account for media variations nearby. Thus, a computation in which a ray is totally reflected from a thin, hard layer in the ocean bottom is not correct. In reality, if the layer is thin relative to the wavelengths in question, the ray will pass through the layer with little change. Another example of this phenomenon is reflection from an ice layer above the ocean that is thin in relation to the wavelength of interest. In reality, the wave “senses” the pressure release interface above the ice interface.

Another problematic area for ray theory calculations is in the neighborhood of the critical angle θ_{cr}. A critical angle exists when a ray is incident on an interface across which the sound speed increases from c_1 to c_2 and is computed as $\theta_{\mathrm{cr}} = \arccos(c_1/c_2)$. Rays steeper than θ_{cr} are partially transmitted into medium 2 and partially reflected back into medium 1. Rays shallower than θ_{cr} are totally reflected back into medium 1. Eigenrays that reflect from an interface at angles very close to θ_{cr} can be inaccurate. As the number of such reflections increases, the inaccuracy becomes worse. The inclusion of the beam displacement in ray path trajectories generally improves the accuracy.

References

1. E. K. Westwood and P. J. Vidmar, “Eigenray finding and time series simulation in a layered-bottom ocean,” *J. Acoust. Soc. Am.* **81**, 912–924 (1987).

2. E. K. Westwood and C. T. Tindle, “Shallow water time series simulation using ray theory,” *J. Acoust. Soc. Am.* **81**, 1752–1761 (1987).

3. D. W. White and M. A. Pederson, “Evaluation of shadow-zone fields by uniform asymptotics and complex rays,” *J. Acoust. Soc. Am.* **69**, 1029–1059 (1980).

4. H. Weinberg, “A continuous-gradient curve-fitting technique for acoustic-ray analysis,” *J. Acoust. Soc. Am.* **50**, 975–984 (1971).

5. C. W. Spofford, R. R. Greene, and J. B. Hersey, “The estimation of geoacoustic ocean sediment parameters from measured bottom-loss data,” SAI Report 83-879-WA, Science Applications, Inc., McLean, VA, March 1983.

6. A. I. Eller, “Findings and recommendations of the surface loss model working group: final report,” Naval Ocean Research and Development Activity Technical Note 279, NSTL, MS (1984).

7. H. Weinberg, “Generic Sonar Model,” TD 5971C, Naval Underwater Systems Center, New London, CT (December 1981).

8. E. L. Hamilton, “Geoacoustic modeling of the seafloor,” *J. Acoust. Soc. Am.* **68**, 1313–1340 (1980).