N-96-01
II-A-140

A MODEL FOR THE PREDICTION OF HIGHWAY CONSTRUCTION NOISE

October 1981

Office of Noise Abatement & Control
U. S. Environmental Protection Agency
Washington, D. C. 20460

N-96-01
II-A-140

TECHNICAL REPORT DATA
(Please read Instructions on the reverse before completing)

1. REPORT NO. EPA 550/9-81-314-B	2.	3. RECIPIENT'S ACCESSION NO.
4. TITLE AND SUBTITLE A Model for the Prediction of Highway Construction Noise		5. REPORT DATE December 1980
		6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) Kenneth J. Plotkin		8. PERFORMING ORGANIZATION REPORT NO. WR 80-58
9. PERFORMING ORGANIZATION NAME AND ADDRESS Wyle Laboratories/Wyle Research 2361 Jefferson Davis Highway, #404 Arlington, Virginia 22202		10. PROGRAM ELEMENT NO.
		11. CONTRACT/GRANT NO. DOT-FH-11-9455
12. SPONSORING AGENCY NAME AND ADDRESS U.S. Environmental Protection Agency, Office of Noise Abatement & Control (ANR-471) Washington, D.C. 20460, and U.S. Department of Transportation, Federal Highway Administration, Washington, D.C. 20590		13. TYPE OF REPORT AND PERIOD COVERED Final
		14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES
Completed under an Interagency Agreement jointly sponsored by both EPA (Office of Noise Abatement and Control) and FHWA.

16. ABSTRACT

This study investigated the noise associated with highway construction activities. It involved the identification and examination of: highway construction activities, noise characteristics associated with highway construction activities, availability of highway construction noise abatement measures, demonstration of construction site noise abatement measures, and development of a computer-based model for use as a tool to predict the noise impact of construction activities and to plan mitigation measures. The model was developed for use on the FHWA computer (IBM 360).

A total of seven reports were prepared in this study and have been released for public distribution.

This report (Part B) presents a complete description of the highway noise prediction model. The report contains a description of the model's formulation and construction, a description of the program, and a user's manual.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group
Highway Noise Model Construction Noise Model Construction Noise Levels Noise Abatement		

18. DISTRIBUTION STATEMENT Unlimited	19. SECURITY CLASS *(This Report)* Unclassified	21. NO. OF PAGES
	20. SECURITY CLASS *(This page)* Unclassified	22. PRICE

EPA Form 2220-1 (9-73)

A MODEL FOR THE PREDICTION OF HIGHWAY CONSTRUCTION NOISE

October 1981

Office of Noise Abatement & Control
U. S. Environmental Protection Agency
Washington, D. C. 20460

FORWARD

This study was jointly sponsored, through an Interagency Agreement (IAG), by the Office of Noise Abatement and Control (ONAC), U.S. Environmental Protection Agency (EPA), and the Federal Highway Administration (FHWA), U.S. Department of Transportation (DOT). The study was conducted by Wyle Laboratories under contract to FHWA Contract No. DOT-FH-11-9455. Wyle Research of El Segundo, California, and Wyle Research of Arlington, Virginia, performed the study.

The object of the study was to investigate and study the noise associated with highway construction activities. The study involved the identification and examination of: highway construction activities, noise characteristics associated with highway construction activities, availability of highway construction noise abatement measures, demonstration of construction site noise abatement measures, and development of a computer-based model for use as a tool to predict the noise impact of construction activities and to plan mitigation measures. The model was developed for use on the FHWA computer (IBM 360).

The principal project officers for Wyle Laboratories on this project were Mr. William Fuller of Wyle Research in El Segundo and Dr. Kenneth Plotkin of Wyle Research of Arlington, Virginia.

The government project managers for the study were Mr. Fred Romano of FHWA, and Mr. Roger Heymann of EPA/ONAC.

The various technical reports completed by Wyle under this contract and submitted to FHWA have been released for public distribution by EPA.

PREFACE

This study involved a comprehensive review of the environmental noise associated with highway construction activities. A total of seven reports have been released for public distribution. These reports are:

1. Analysis and Abatement of Highway Construction Noise, EPA 550/9-81-314-A, September 1981.
2. A Model for the Prediction of Highway Construction Noise, EPA 550/9-81-314-B, September 1981.
3. IBM 360/System Batch Version of Highway Construction Noise Model, EPA 550/9-81-314-C, September 1981.
4. Appendix A, Highway Construction Noise Field Measurements, Site 1: I-201 (California), EPA 550/9-81-314-D, September 1981.
5. Appendix B, Highway Construction Noise Field Measurements, Site 2: I-205 (Oregon), EPA 550/9-81-314-E, September 1981.
6. Appendix C, Highway Construction Noise Field Measurements, Site 3: I-95/I-395 (Maryland), EPA 550/9-81-314-F, September 1981.
7. Appendix D, Highway Construction Noise Field Measurements, Site 4: I-75 (Florida), EPA 550/9-81-314-G, September 1981.

The first two reports (Part A and Part B) might be considered the principal reports since they are relatively self-contained units on this study's efforts, the engineering studies and the computer model, respectively. In this regard, if there is to be a limited purchase of the reports, one might consider obtaining either or both of Part A and Part B, and obtaining the other reports as additional informational needs arise.

- The first report (Part A) contains all of the information from the engineering study phase of the project. It gives information on highway construction procedures, highway construction site noise characteristics, available abatement measures, and results from field demonstrations on noise abatement.

- The second report (Part B) presents a complete description of the highway noise prediction model. The report contains a description of the model's formulation and construction, a description of the program, and a user's manual.
- The third report (Part C) provides additional information to the Part B report on the highway construction noise model installed at DOT's Transportation Computer Center on an IBM 360 computer. It delineates the differences between the version of the model as installed on the IBM 360 and the two models (HINPUT and HICNOM) operating on the Wyle Computer (PDP-11). The report has additional user's manual information for use on the IBM 360, a programmer's manual describing changes in going from the PDP-11 to the IBM 360, and a maintenance manual.
- Reports 4, 5, 6, and 7 (Part D through Part G) contain field data gathered at the field demonstrations at highway construction sites in: Route I-201, California; I-205, Oregon; I-95/I-395, Maryland; and I-75, Florida. They contain noise data on single and multiple pieces of equipment, provide general description of highway site activities, and activity analyses of equipment.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

This report contains the documentation for HICNOM, the highway construction noise model developed by Wyle Laboratories for the Federal Highway Administration. This model permits the calculation of L_{eq} in the vicinity of highway construction sites. It is intended to be used by personnel involved in highway planning for the evaluation, analysis, and abatement of noise during the construction phase. The model is designed for users with minimal background in acoustics, and contains acoustic data for a large variety of construction equipment. However, it has options which permit the entry of user-supplied data bases and is therefore useful to acoustical specialists as well. While acoustic expertise is not required to operate this program, some knowledge of highway construction procedures and equipment is needed. Those desiring more background information on construction activities and noise, and on the field measurements used in the development of this model, should read the previous reports prepared under this contract, cited here as References 1 through 5.

The computer program as presented here was developed on a PDP 11V03 computer with 32K memory. It is run in two parts: HINPUT, an input and task activity module, and HICNOM, the main acoustical calculation. The memory size restriction permits the program to be run on small minicomputers or on larger microprocessor-based systems with FORTRAN capability. The programs were written in a way which makes them straightforward to combine for single-unit running on a larger machine. The program is written in a conversational mode, with data requested at the terminal as needed. This simplifies operation for users who may not be experienced in computer programming.

Section 2.0 of this report contains a description of the mathematical formulation of the model and the representation of construction activities. Section 3.0 is a description of the computer program. Section 4.0 is the user's manual, and gives complete instructions for using the program. Section 5.0 is the programmer's manual. It contains descriptions of all subroutines, data configurations, and programming conventions. Section 6.0 is the maintenance manual, and describes the main data blocks and procedures for updating them. A complete listing of the program is contained in Appendix A.

2.0 MODEL FORMULATION AND STRUCTURE

A construction site may be viewed as a collection of acoustic sources of various types, with the resultant noise levels outside the site being of interest for the present study. Where the noise is desired in terms of L_{eq} or L_{dn}, timing of equipment motion does not matter: only the energy-averaged noise emission and the area covered are required. Acoustically, a site may thus be modeled as a collection of fixed point, line, and area sources. The acoustic structure of the model has therefore been formulated as a general method of handling an arbitrary arrangement of these three source types. Leading into this acoustic structure is a task model and system section which transforms user inputs of construction activity into these three types of source. The model accepts data specifying equipment location and type of activity, and combines these with an equipment noise level and duty cycle data base. Algorithms are included which generate detailed motion parameters for mobile equipment (truck turnaround loops, acceleration profiles, etc.) and which balance activity levels between pieces of equipment working together.

2.1 Acoustical Formulation

The model considers noise from point, line, and area sources. Attenuation with distance consists of geometrical spreading plus a power law excess attenuation representing ground plus air absorption. The mathematical representations are described in the following subsections.

2.1.1 Point Sources

For a source and receiver separated by a distance r, the received sound pressure level is:

$$L = L_0 + 20 \log_{10} (d_0 / r)^{1+n} \qquad (1)$$

where L_0 is the sound level measured at a reference distance d_0, and n is a number between 0 and 1. A value of $n = 0$ corresponds to no excess attenuation, while $n = 1$ corresponds to the excess attenuation of 6 dB per doubling of distance which occurs for long-distance propagation at grazing angles over ground of finite impedance. If L_0 is the energy-average emission over some time period, then L in Equation (1) is L_{eq}.

2.1.2 Line Sources

Consider a finite line source segment as shown in Figure 1, which consists of point source elements moving at a uniform speed. L_{eq} is obtained by writing Equation (1) for

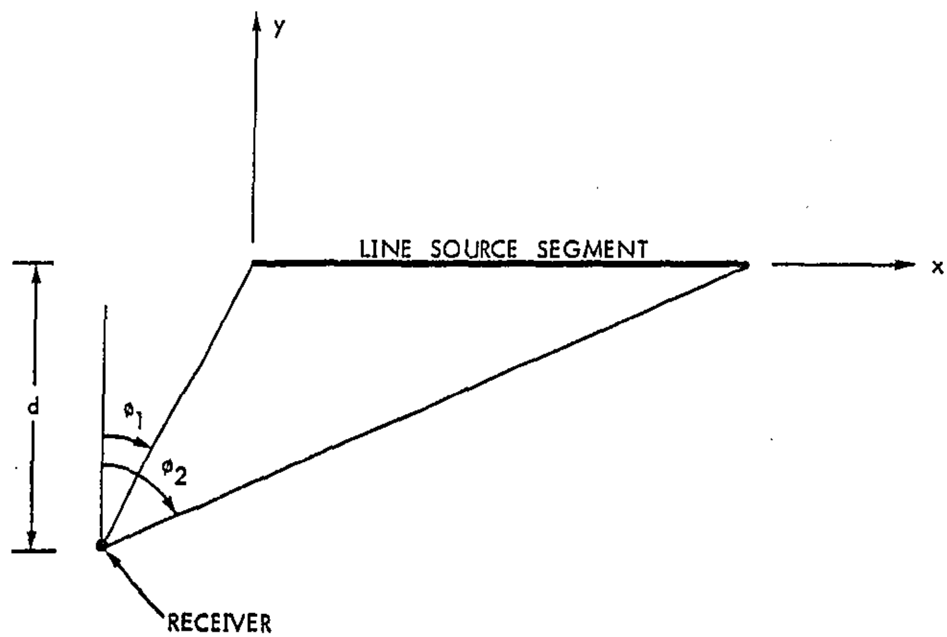

Figure 1. Geometry and Coordinates of Line Source Segment.

each point element, combining, and integrating over time. This integration has been well established in the highway noise literature. Following the general formulation in Reference 6, L_{eq} is given by

$$L_{eq} = L_0 + 10 \log_{10} d_0 N\pi + 10 \log_{10} (d_0 / d)^{1 + 2n} + 10 \log_{10} G(n, \phi_1, \phi_2) \qquad (2)$$

where L_0 is the energy-average passby level of the point source elements ($L_{eq(e)}$), N is the number of elements per unit length (equal to the number of sources passing per unit time times their speed), d is the sideline distance to the receiver, and G is a function given by:

$$G(n, \phi_1, \phi_2) = 1/\pi \int_{\tan \phi_1}^{\tan \phi_2} \frac{d\xi}{\left(1 + \xi^2\right)^{1+n}} \qquad (3)$$

where ϕ_1 and ϕ_2 are the angles to the end points of the line segment, as shown in Figure 1. Equation (3) was first derived in Reference 6 for an infinite length road, and closed-form solutions presented for arbitrary n. The generalization to a finite road was presented in Reference 7, together with extensive tables of numerical values for one value of n and a range of road lengths, in terms of the angles ϕ_1, ϕ_2. For $n = 0$, Equation (3) reduces to the factor $(\phi_2 - \phi_1)/\pi$ appearing in the commonly used finite road adjustment $10 \log_{10} (\Delta\phi/\pi)$,[8] where $\Delta\phi = \phi_2 - \phi_1$.

A closed-form representation of Equation (3) can be derived. The integral may be written in two parts:

$$G = 1/\pi \left[\int_0^{\tan \phi_2} \frac{d\xi}{\left(1 + \xi^2\right)^{1+n}} - \int_0^{\tan \phi_1} \frac{d\xi}{\left(1 + \xi^2\right)^{1+n}}\right] \qquad (4)$$

Each integral has form similar to the incomplete beta function:[9]

$$B_x(a, b) = \int_0^x t^{a-1} (1-t)^{b-1} \, dt \qquad (5)$$

Letting $t = \xi^2 / (1 + \xi^2)$,

$$B_x(a, b) = 2 \int_0^{\sqrt{x/(1-x)}} \xi^{2a-1} \frac{d\xi}{\left(1 + \xi^2\right)^{a+b}} \qquad (6)$$

The integral in Equation (6) matches those in Equation (4) providing that

$$\begin{aligned} x &= \sin^2 \phi \\ a &= 1/2 \\ b &= n + 1/2 \end{aligned} \tag{7}$$

Thus,

$$G(n, \phi_1, \phi_2) = (1/2\pi)\left[B_{\sin^2\phi_2}(1/2, n + 1/2) - B_{\sin^2\phi_1}(1/2, n + 1/2)\right] \tag{8}$$

The incomplete beta function may be evaluated from the following series:

$$B_x(a, b) = \frac{x^a (1-x)^b}{a}\left\{1 + \sum_{j=0}^{\infty}\left[\prod_{i=0}^{j}\left(\frac{a+b+i}{a+1+i}\right)\right] x^{j+1}\right\} \tag{9}$$

Equation (9) converges for $x < 1$, and poses no computational difficulties. To include the point $x = 1$, the following inversion relation may be used:

$$B_x(a, b) = B(a, b) - B_{1-x}(b, a) \tag{10}$$

where $B(a, b)$ is the beta function, and may be written in terms of the gamma function:[9]

$$B(a, b) = \frac{\Gamma(a)\,\Gamma(b)}{\Gamma(a, b)} \tag{11}$$

To minimize computation time, Equation (9) is used only for $x \leq 0.5$ (the series converges slowly near 1) and Equation (10) is used to obtain values in the range $0.5 < x \leq 1.0$.

Equations (2) and (8) through (11) provide a closed set of relations for computing L_{eq} from a straight line segment. Curved lines can be approximated by a series of short straight segments. It is applicable to any line where the average source distribution is uniform. In the model, line sources are input as a series of points defining a set of straight sections.

2.1.3 Area Sources

Areas are described to the model in terms of a series of points defining a piecewise straight centerline of the area, plus the width of the area at each point, as illustrated in Figure 2. In practice, areas tend to be quite regularly shaped, following the geometry of lanes under construction, so that this representation is very good for highway construction.

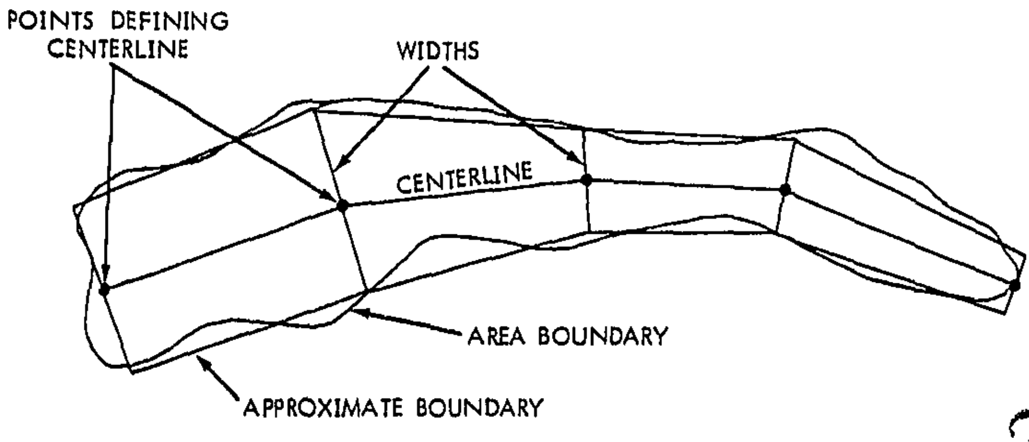

Figure 2. Representation of Area By Centerline And Widths.

No simple closed relationship exists for noise from an arbitrary area source with excess attenuation; some numerical integration is required. The approach taken here is to divide an area into strips, each strip sufficiently narrow relative to receiver distance that it may be modeled as a line. Each segment is divided into strips parallel to the centerline of that segment. One strip is the centerline, and the others are symmetrically defined on either side. The number of strips taken is based on the ratio between the strip width and the distance to the nearest receiver. The ratio is adjusted such that each strip can be approximated by a line with a worst-case error of less than 0.2 dB; this is essentially negligible. Following the construction of strips, the source density N is taken as the number of sources working the area segment divided by the total length of all strips in the segment. Noise is then computed by combining Equation (2) for all strips.

2.1.4 Barriers

Shielding by barriers is handled by Maekawa's formulation for screens.[10] Barriers are specified by providing a series of coordinates defining the top edge as a series of straight lines. Several barriers may be specified. The model checks each source/receiver pair for shielding by each barrier. Double shielding is not considered; once a barrier is found to shield a given path, other barriers are neglected for that path. When there is barrier shielding, ground attenuation is set equal to zero. In the event that this results in an amplification (as might happen for a low barrier), the barrier section is neglected and ground attenuation restored. This avoids a physically unrealistic case, and also permits the handling of barriers with a gap by specifying one section to have zero height.

For point sources, Maekawa's methodology is applied directly. Each source type has a nominal frequency assigned to it which is used for computing the Fresnel number. This procedure is a generalization of the use of 500 Hz to represent highway noise.[8] The nominal frequency for each type of equipment is the peak of the A-weighted spectrum. This reasonably approximates the shielding which would be obtained if an exact calculation were performed over the complete A-weighted spectrum.

Barrier shielding for line sources is computed by the method of Kurze and Anderson,[11] wherein point source shielding is integrated along the line. A numerical integration scheme is used in the program. An initial mesh consisting of the line end and center points is used. The mesh is successively doubled until the calculated shielding converges within about 0.4 dB. This shielding calculation is carried out for each section.

Barrier shielding for area sources is carried out by applying the line source methodology to each strip.

2.1.5 Source and Receiver Locations

Source and receiver locations are specified by the user as sets of cartesian coordinates x, y, and z. The x and y coordinates are in the horizontal plane, and z is vertical location. The origin and orientation of the coordinates may be set by the user in any convenient way.

For the noise calculations described in Sections 2.1.1 through 2.1.3, only the horizontal locations x, y are considered. The difference between true three-dimensional distance and the horizontal component is generally negligible. In the field program,[5] no elevation differences were found which would cause a difference of more than a fraction of a dB due to increased distance. The neglect of z in this analysis provided great simplification in the program.

Vertical distances are of critical importance when barriers are involved, so that true three-dimensional distances are used when computing the path length difference δ as described in Section 2.1.4. Coordinates specified for a barrier represent the spatial location of the top edge: the location of the ground at the barrier plus the height of the barrier above the ground. Similarly, receiver locations specified by the user must represent the actual height of the receiver.

Source elevations are specified somewhat differently. The height of the effective acoustic source location above the ground varies between various pieces of equipment, and forms part of the data base in the model. Source locations are therefore specified by the user as the ground-level location; the model adds the appropriate height to obtain the acoustic source location.

2.2 Construction Activities

As described in References 1 through 5, highway construction generally takes place in seven phases, beginning with mobilization and ending with finishing. Reference 4 contains a detailed list of activities and equipment involved in each of these phases. Review of this list indicates a considerable overlap in equipment between phases. The construction noise model is therefore organized such that the user specifies the equipment present, rather than the phase or task being performed. This greatly simplifies the operation of the model, and provides a direct means for the user to evaluate different approaches to a given task.

The following subsections describe the modeling of various equipment, their acoustic representation, and their coordination in multi-equipment tasks.

2.2.1 Construction Equipment

Table I lists equipment used in highway construction, organized by whether it is typically operated at a point, along a line, or over an area. Secondary geometry is indicated in parentheses for those pieces of equipment which may be operated in different ways. Also indicated is whether the equipment has a volume capacity associated with it. The equipment listing is based on References I through 5, with the categorization based on the field data reported in Reference 5.

Construction equipment can be divided into two categories: haul equipment (trucks and scrapers) and non-haul equipment. Haul equipment moves in an orderly manner along a specified path, and has definite positions at which it accelerates, decelerates, stops to load and dump, and cruises. These different modes must be accounted for; noise next to an acceleration section can be significantly louder than deceleration. The following model, developed and validated for trucks,[12] has been used to represent the noise from haul equipment:

$$L = \begin{cases} L_0 + \text{slope} \cdot \log_{10}(V/V_{ref}) & \text{Coast, Cruise, and Acceleration at } V > V_{crit} \\ L_0 & \text{Acceleration at } V < V_{crit} \end{cases} \qquad (12)$$

In Equation (12), L is $L_{eq(e)}$ at a reference distance, and L_0, slope, V_{ref}, and V_{crit} are empirical quantities. Specific values used in the model are based on the field data reported in Reference 5, and are discussed later.

Non-haul equipment does not have its operating modes strongly correlated with position; all modes occur more or less equally distributed along its operating area. The effective source level required by the program is the L_{eq}, averaged over several duty cycles, which would be measured by a microphone moving at a fixed reference distance from the equipment. In the program, this level is represented by

$$L = L_{max} - \Delta \qquad (13)$$

where L_{max} is the maximum level and Δ is the difference between L_{max} and L_{eq}. The noise is presented this way because L_{max} is much more widely reported in the literature (virtually all standard equipment noise measurement procedures obtain L_{max}), and Δ varies much less than L_{max} for a given type of equipment. Both L_{max} and Δ used in the noise model are derived from the field data.[5] The quantity Δ is equivalent to the "acoustic max factor" introduced in Reference 13, but in this project has been measured directly in the far field rather than from on-board instrumentation.

Table I

Construction Equipment

POINT SOURCES		CAPACITY
Backhoe	(Area)	Yes
Front End Loader	(Area)	Yes
Compressor		No
Pile Driver		No
Pump		No
Crane		Yes
Pavement Breaker	(Area)	No
Concrete Batch Plant		No
Concrete Pours		No
Hand-Held Equipment	(Area)	No
Power Shovel		Yes
LINE SOURCES		
Bulldozers	(Area)	No
Graders	(Area)	No
Trucks		Yes
Scrapers		Yes
AREA SOURCES		
Compactors		No
Spreaders		No
Paving Trains	(Line)	No

2.2.2 Task Models

A major section of the program converts the user inputs of equipment, locations, and activities into the acoustical quantities required for Equations (1) through (11) as described in Section 2.1. The user specifies the type and model of equipment and its location. The program assigns appropriate noise levels and other parameters, including the source density for extended sources. The program also computes production rates for earthmoving equipment where applicable. The rate is based on capacity per cycle and the duration of a cycle. Where equipment works together, such as a shovel loading trucks, the user may specify that the operation is coordinated. The program then provides the appropriate truck volume and/or shovel usage factor so as to match the production rate. This permits the user of the program to specify activity levels in terms of the quantity of work being performed.

As noted earlier, source data in the program are essentially for single pieces of equipment. Exceptions to this occur when a number of pieces of equipment act in a strictly coordinated manner, so that the entire task may be treated as a unit. In the field program,[5] this was found to occur less often than might be expected. Cement batch plants, concrete pours, and paving operations were found to be such tasks. The data for these operations in the program represent lumped task noise levels.

2.2.3 Haul Road Kinematics and Geometry

In most cases, source locations specified by the user are passed through the task models directly to the acoustic section. A major exception is in the case of haul roads. Based on the character of haul operations, detailed information is required as to the location of stopping points and the speeds during acceleration and deceleration. It is also quite common for trucks to turn on a loop through a loading/unloading area. The user can specify all of this information, but very often the labor involved is not justifiable in terms of the importance of truck noise or the lack of precision of knowledge of the haul road geometry. Haul roads are quite temporary in nature, and especially at loading points can change from day to day. Often all that is accurately known is the approach location of the road, the loading point, and the approximate radius of a turnaround loop. The program therefore contains routines which generate points defining a loop, and which generate speed profiles around stopping points. Each option may be independently specified. Figure 3 shows optional return loop shapes. A turning radius must be specified for these. Speed profiles are generated by constant acceleration and deceleration from zero at the stopping point to specified cruise values. The acceleration and deceleration values are data quantities; values of 0.1g are currently in the program.

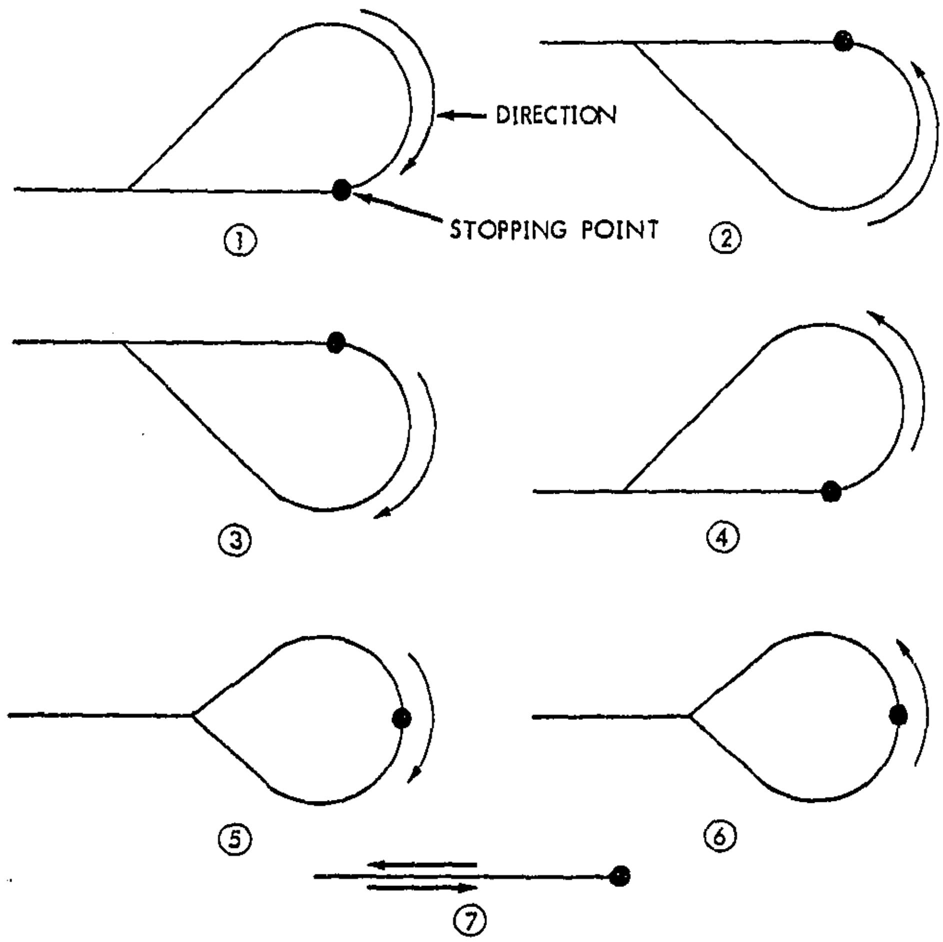

Figure 3. Haul Road Turnaround Loop Configurations.

2.3 Data Base

A key part of the model is the inclusion of an extensive data base of equipment noise levels and operating parameters. The bulk of the data incorporated into the model was gathered specifically for this purpose. The data collection program is described in Reference 5. Some additional data were obtained from a review of the literature,[3] but this was generally limited to maximum emission levels and manufacturer's specification of capacity.

Table 2 lists the equipment types and models contained in the model. Values of data quantities may be seen in the listing of block data program DATA2 in Appendix A. Table 2 is to be used for input selections when running the program using this data base. The model identification numbers used in the program are indicated in the table.

Each data item identified in Table 2 is an average of a number of similar pieces of equipment. Items marked "nominal" are from the literature; the remainder are from Reference 5. The descriptions in Table 2 range from quite specific to somewhat vague. They are presented in the most descriptive manner which does not lead to over-generalization. In some cases it will be necessary to examine the noise levels in the data base and make a judgment as to which model is appropriate. A wide variation in noise levels is not uncommon. It was found in the field program that nominally identical pieces of equipment could produce noise levels which differ by 10 dB or more. Construction sites generally have small numbers of equipment present (small in a statistical sense), so that caution must be exercised in any use of average levels for a specific case. The data in the program does provide a good selection of the range encountered, however, so that reasonable choices can be made in the absence of specific data.

Table 2

Equipment Types and Models

Type	Model No.	Description
Backhoe	1 2 3	Nominal Caterpiller, Koehring P&H
Loader	1 2 3 4 5	Nominal 3-yard capacity 5-yard capacity 7-yard capacity 10-yard capacity
Compressor	1 2 3 4	Nominal Standard Quiet, doors open Quiet, doors closed
Pile Driver	1 2	Nominal Current data
Pump	1 2	63 dB @ 50 feet 76 dB @ 50 feet
Crane	1 2 3 4	Nominal Quiet Medium Loud
Breaker	1 2 3	Rock Drill Standard Jackhammer Muffled Jackhammer
Concrete	1 2 3 4 5	Concrete Pour Nominal Batch Plant Patch Plant Pump Cement Mixer
Generator	1 2	Average Nominal
Miscellan	1 2 3 4	Grinder Concrete Saw Fan Welder

Table 2 (Concluded)

Bulldozer	1 2 3 4	Nominal Caterpiller D6, D7, D8 Caterpiller D9 D9 with muffler
Grader	1	Average
Trucks	1 2 3 4	10-yard dump, quiet 10-yard dump, noisy Dual 20-yard trailers Nominal
Scraper	1 2 3 4	Caterpiller 631, muffled Caterpiller 631, no muffler Caterpiller 623 Caterpiller 637
Compactor	1 2 3	Quiet Medium Loud
Paving	1 2 3	Nominal Concrete Paver Asphalt Paver

3.0 PROGRAM DESCRIPTION

3.1 Capabilities

The program has the following capabilities:

- Calculation of L_{eq} for construction activities representing various point, line, and area sources.
- Up to 10 receiver locations.
- Up to 10 point sources.
- Up to 6 line sources, each described by up to 20 points.
- Up to 5 area sources, each described by up to 10 centerline points and widths.
- Up to 3 barriers, each defined by up to 5 top edge points.
- Built-in data base for over 50 types and models of construction equipment.
- Easy user-specification of additional data.
- Activity levels can be automatically balanced between equipment working together.
- Excess attenuation specified by user.
- Automatic generation of haul road turnaround loops and acceleration/deceleration profiles.
- Diagnostic output identifying the contribution of each source to the overall noise.

The quantitative limits noted above are primarily due to the dimensioned size of arrays. They can be increased by changing appropriate dimensions, as discussed in Sections 5.0 and 6.0. The equipment data base can be increased to about 300 types and models within existing dimensions; only data statements described in Section 6.0 need be modified up to this limit.

3.2 Logical Structure

The program is divided into two main sections: an input and task module which accepts geometric inputs and task descriptions and generates acoustic source quantities, and the acoustical part which computes receiver noise levels. To fit within the 32K-word memory of the PDP 11 system, these two sections are run separately, under control of main programs HINPUT and HICNOM, respectively. HINPUT accepts user data conversationally from a terminal, prompting the user when various options are to be specified. The calculations described in Section 2.2 are performed by HINPUT, which then creates a data file to be read by HICNOM. HICNOM performs the noise calculations described in Section 2.1. The data file created by HINPUT is annotated so that the user can review it if desired.

The following subsections describe the logical structure of the two programs.

3.2.1 Input and Task Program HINPUT

Figure 4 is a flow chart of HINPUT. Indicated on the chart are the names of key subroutines used in each section. The subroutines are described in detail in Section 5.

The user first enters the coordinates of receiver locations and a value of excess attenuation. These values are passed directly through to the data file. Construction activities are specified by inputting the name of a particular piece of equipment. The subroutine DECODE checks the input name against a list of allowable names. An internal identification index is returned, together with an indicator as to whether the source is a point, line, or area. User-defined equipment may be specified. The input equipment type name is appended with a sequential number, e.g., the third backhoe input is called "BACKHOE 3", for identification in the data file.

Separate sections of code handle the remaining input and analysis for the three geometry types. The source location is input, then control is transferred to the appropriate task subroutine: PTTASK, LNTASK, or ARTASK. Each of these routines looks up the emission level of the equipment and adds its effective source height to the input location. If a user-defined piece of equipment is specified, the program requests the appropriate data. These data are stored so that this new equipment type may be referred to later in the same run. The task routines compute the production rate for appropriate equipment, and utilize this to obtain usage factors (fraction of time operating) when equipment is coordinated. The line and area sections divide the number of sources by total length or area to obtain source density.

The line task section (via subroutine LNTASK) additionally contains routines to generate turnaround loops and acceleration/deceleration profiles. The available options are described in Section 4, and details of the computation are in Section 5. Briefly, several options are available which result in the addition of a number of points to the input line geometry and which cause the calculation of average speed, source density, and emission level on each segment. These modified and/or computed values are returned to the main program for inclusion in the output data file.

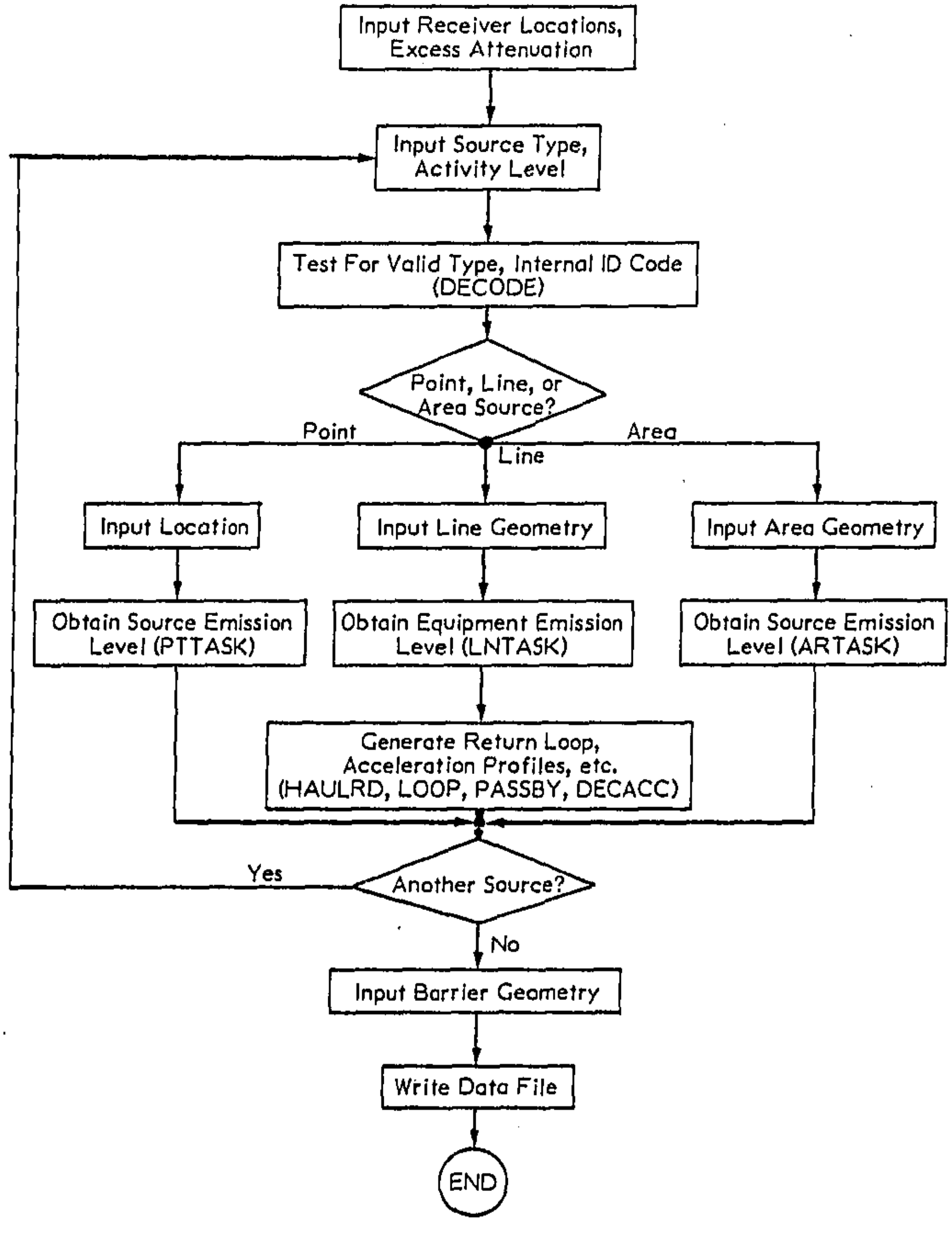

Figure 4. HINPUT Flow Chart.

Following input of source data, the geometries of any barriers are then input. These are passed through unaltered to the output data file. Writing the data file is the final step of this program.

3.2.2 Acoustical Computation Program HICNOM

Figure 5 is a flow chart of HICNOM. The key subroutines parenthetically noted are described in detail in Section 5.

Program structure is essentially linear. The data file created by HINPUT is read, then the noise at each receiver is computed for the point, line, and area sources. When barriers are present, the noise contribution from each source element is first computed without barriers, then with barriers. The smaller value is taken, as discussed in Section 2.1.4.

The point source calculation is quite straightforward, based on Equation (1), and the no-barrier case is handled entirely within the main program. The line and area cases are somewhat more complex. In order to standardize the line source calculation, Equation (2), each line segment is transformed to a local coordinate system oriented as shown in Figure 1. For a line between points i and $i + 1$, point i is placed at the origin and $i + 1$ on the positive x-axis. Each receiver location is transformed to these coordinates, with the y value taken as $-|y|$. The quantities d, ϕ_1, and ϕ_2 are then directly obtained for use in subroutine LINSRC, which performs the calculation of Equation (2). The coordinate transformation is performed by subroutine GEOM. GEOM is a general-use transformation routine which is used for a variety of purposes in both HICNOM and HINPUT.

The area calculation requires division of areas into strips, as described in Section 2.1.3. This task is performed by subroutine AREA. Each strip is treated as a separate line element, as are the segments of line sources.

Running indices are created in parallel to the main calculation. These keep the summation over subsources organized, and permit identification of subsource components. The indices and the key to subsources are described in the user's manual, Section 4.

3.3 Validation

Validation was performed at two levels: computational validation, where the computer code was checked out, and application validation, where predictions were compared with field data. The latter includes effects due to non-ideal data, and is discussed in the user's manual, Section 4, and in Reference 5.

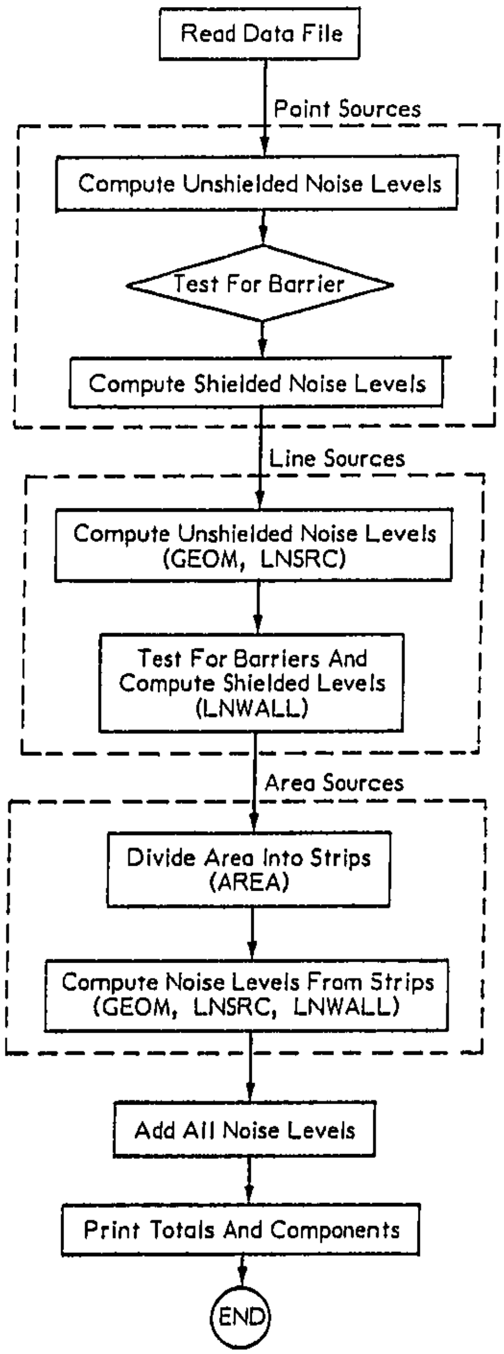

Figure 5. HICNOM Flow Chart.

Extensive validation of the computer code was carried out as the program was developed. The procedure for testing subroutines was to use special input data through the actual main program, rather than dummy driver routines. All options were exercised, and results checked against hand calculations and/or exact solutions. Particular attention was paid to limiting cases of the algorithms used; for example, coordinate transformation involving rotation angles of exact multiples of $\pi/2$. Temporary output statements were used to check key intermediate results. The precisions noted in Section 2 were found to be well satisfied.

4.0 USER'S MANUAL

4.1 Hardware Requirements

The program is written in FORTRAN IV and is configured to run on a DEC PDP 11V03 computer with 32K words of memory, dual floppy discs, Decwriter terminal, and RT 11 operating system. The operating system and run modules are on one disc in drive DX0. A second disc, for the data file created by HINPUT, is placed in DX1. Each program requires most of the 28K memory allowed for user programs. The terminal must be assigned to logical device 7. The program opens a file on logical device 1; this must not conflict with other peripheral devices. Eighty-character-wide paper is sufficient for all output formats.

4.2 Input Data Requirements

The following data are required to run the program:

- Cartesian coordinates of receiver locations.
- Excess attenuation, expressed as dB per doubling of distance.
- Names of the equipment at the site. Table 2 is a list of equipment types and models built into the program.
- Coordinates defining line and area source geometries.
- Coordinates defining barrier locations.

Additional information which may be needed is speed and volume data for haul operations, the nature of turnaround loops, activity level data, and source data for equipment defined by the user.

4.3 Running The Program

To run the programs, load the discs and bootstrap the system on DX0 as specified above. Enter the RUN command for each program.

4.3.1 Running HINPUT

All input data are prompted by requests from the program. The following is a list of prompts, in the order they appear, and the required data. Specific input and output phrases below are capitalized. Phrases printed by the computer are underlined. All data inputs are free format, separated by commas or carriage returns, and terminated by a carriage return. Some error checking is included in the program, but it is not foolproof. Restrictions are noted below which are the responsibility of the user.

Units are specified by the data base. The default units for input are distances in feet, speeds in mph, capacities in cubic yards, duty cycle time in hours, and frequency in Hz.

HOW MANY RECEIVERS? Enter the number of receivers, from 1 to 10.

INPUT X, Y, Z OF EACH RECEIVER Enter the coordinates of the receiver locations. The length of the list must be consistent with the number of receivers.

EXCESS GROUND ATTENUATION, DB/DD Enter the ground attenuation factor. A value of 1 to 1.5 was found to be typical at cleared construction sites.[5]

SOURCE TYPE? Enter a source type exactly as spelled in Table 2. Enter USER DEFINED to specify a new type. A blank (carriage return only) signifies no more sources will be entered. Any other entry causes the program to respond INVALID SOURCE TYPE. REENTER: until a valid type is entered.

MODEL NUMBER Enter a model number from the choices in Table 2. To create a new model, enter 0. To specify a previous user-created model, enter -1, -2, etc., to specify "last new model of this type", "next to last new model of this type", etc. The total number of models of a given type (highest value in Table 1 plus user-created models) must not exceed ten. Note that the model number for USER DEFINED is always 0 the first time, and zero or negative thereafter.

ENTER 1, 2, OR 3 FOR WORKING OVER POINT, LINE, OR AREA: Appears when there is a choice of geometries; enter the appropriate value. This always appears for USER DEFINED type.

HOURS WORKED Enter the number of hours worked per day. A full day is 8 hours. (This is a default value; it may be changed. See Section 6.) Working at less than full efficiency reduces this; for example, if there is a 75-percent use factor, enter 6. Entering a negative value indicates that the program is to compute the use factor so as to match the production rate to the last equipment with a production rate. Equipment for which a production rate exists are indicated in Table 1. To avoid errors, group equipment together. For example, in a load and haul operation, first specify the activity level for either the trucks or the loader, then enter the other with a negative number for hours worked. Note that a negative input can be used to specify separate activities with the same net production.

If the equipment specified operates at a point, the following is requested:

ENTER X, Y, Z OF SOURCE LOCATION Enter the appropriate coordinates.

If a new piece of equipment is being defined which operates at a point, the following is requested:

ENTER LMAX, DELTA, CAPACITY, CYCLE TIME, ACOUSTIC HEIGHT, AND FREQUENCY: Enter these data. LMAX and DELTA are L_{max} and Δ as defined in Section 2.2.1. CAPACITY is the capacity per cycle (cubic yards), CYCLE TIME is in hours. ACOUSTIC HEIGHT is as defined in Section 2.1.5. FREQUENCY is the effective frequency for barrier calculations. The last two quantities do not matter if there are no barriers.

If an area source has been specified, the following are requested:

HOW MANY POINTS DEFINE CENTERLINE? Enter the appropriate number. This is normally at least two. One is meaningless. Entering 0 or negative indicates that the last previous area is to be reused. This option is useful in cases where different equipment types use the same area, for example, compactors and a water truck in a fill area.

ENTER X, Y, Z AND WIDTH OF POINTS Enter values as defined in Section 2.1.3 and Figure 2.

ENTER LMAX, DELTA, CAPACITY, CYCLE TIME, ACOUSTIC HEIGHT, AND FREQUENCY: Requested if a new model has been specified. Input data are the same as described for point sources above.

HOW MANY? Enter the number of pieces of equipment operating in this area. It is common for several identical pieces to work in a given area.

If a line source has been specified, the following are requested:

HOW MANY POINTS IN LINE? Enter the number of points, usually two or more. Entering 0 or negative means use the last previous line, as describe above for areas.

ENTER X, Y, Z OF POINTS Enter the coordinates of up to 20 points defining the line. Unless a standard turnaround loop is later specified, motion of haul equipment is presumed to be one-way in the direction corresponding to the input order of the points. If a return loop is to be created, the last point is the loading point and the next to last point must be at least 2.5 times the loop radius away from it. Additional points are generated by this option. No more than 6 points should be specified if a loop is to be created by the program.

Line source equipment may be either haul or non-haul, as described in Section 2.2.1. If user-created non-haul equipment is specified, the following is requested:

<u>ENTER LMAX, DELTA, CAPACITY, SPEED, ACOUSTIC HEIGHT, AND FREQUENCY:</u> Enter the same data as described earlier, except that average speed (in mph) is required instead of cycle times. The cycle time is computed by the program from the speed and total path length.

If user-created haul equipment is specified, the following is requested:

<u>ENTER LMAX, REFSPD, SLOPE, VCRIT, CAPACITY, ACOUSTIC HEIGHT, AND FREQUENCY:</u> Enter data as described earlier. LMAX, REFSPD, SLOPE, and VCRIT correspond to L_0, V_{ref}, slope, and V_{crit} as defined in Equation (12).

The following is requested for haul equipment:

<u>ENTER SPEED ON ALL SEGMENTS</u> Enter the speed. If n line points were specified, n – 1 speeds will be required. These are the average speeds on each segment. If program-generated acceleration and deceleration profiles are to be specified, the speeds need only be approach and departure speeds.

<u>VEHICLES PER HOUR:</u> Requested if HOURS WORKED is positive. Enter the volume flow in one direction.

<u>TYPE AND RADIUS OF RETURN LOOP</u> Entering a value from 1 to 7 specifies that a loop as defined in Figure 3 is to be generated by the program. Enter the radius in feet; this is arbitrary for types other than 1 through 6. Entering a zero or negative type indicates straight-through traffic on the input line with no program-generated loop. A value greater than 7 will be treated as 7.

If a zero or negative loop type is specified, the following are requested:

<u>STOPPING AND DECELERATION POINTS</u> If the haul operation has a stopping point (usually for loading or unloading), specify the haul road point numbers (counting from the first) corresponding to the stopping point and the point where deceleration begins. The deceleration point must precede the stopping point, and by a distance consistent with the approach speed and deceleration rate. Specifying a zero or negative stopping point indicates no stopping point, and cruise is presumed at the speeds previously input. Specifying a stopping point and a zero or negative deceleration point indicates that the program is to generate acceleration and deceleration profiles which extend sufficiently far as to match the input speeds; new

speeds are created for the segments involved. Specifying positive deceleration and stopping points indicates that actual average speeds have been input, and no kinematics are to be generated by the program.

Following specification of all sources, the following are requested:

HOW MANY BARRIERS? Enter the number, up to 3.

For each barrier, the following are requested:

HOW MANY POINTS? Enter the number, up to 5.

ENTER X, Y, Z OF POINTS Enter the coordinates of the top edge of the barrier. It should be kept in mind that shielding is computed only for the first barrier encountered for each source/receiver pair; this should be considered when entering data for multiple barriers. The program was also designed to handle straightforward barrier arrangements. Complex shapes which intercept a particular line-of-sight more than once may cause erroneous results.

Following the source and barrier data, the following are requested:

ENTER TITLE Enter a title line, up to 80 characters long. The first character is non-printing, reserved for carriage control on line printers when the title is subsequently printed by HICNOM.

DATA FILE NAME – FILNAM.DAT Enter a name for the data file to be created. Any name and type allowed by RT 11 may be used, but it is recommended that a DAT type be specified.

Figures 6, 7, and 8 are sample HINPUT runs and the data files generated. Part a of each figure is the input at the terminal, and part b is the data file created by HINPUT.

Figure 6 corresponds to a case from the field program. It is a cut area with two identical load and haul operations. In each, two bulldozers push earth toward a loader, which loads tandem trailer dump trucks. Four receiver positions are specified. The first and third receivers' input were site boundary measurement points, while the second and fourth were each close to one of the loading operations. The equipment inputs are the loader, trucks, and bulldozers at one load operation, followed by the same at the other. Note that balance-to-the-last is specified (negative hours worked) for both loader and trucks at the second load operation. This is because both operations were known to have the same production rate, even though they were not directly connected. The intervening bulldozer data were irrelevant to the stored production rate, since bulldozers do not have production defined in this model.

```
.RUN HINPUT
HOW MANY RECEIVERS?
4
INPUT X,Y,Z OF EACH RECEIVER
-500,-610,0
-460,-300,0
-200,300,0
0,0,0
EXCESS ATTENUATION, DB/DD
1

SOURCE TYPE?
LOADER
MODEL NUMBER?
5
ENTER 1, 2, OR 3 FOR WORKING OVER POINT, LINE OR AREA:
1
HOURS WORKED:
8
ENTER X,Y,Z OF SOURCE LOCATION
-110,-65,0

SOURCE TYPE?
TRUCKS
MODEL NUMBER?
3
HOURS WORKED:
-1
HOW MANY POINTS IN LINE?
3
ENTER X,Y,Z OF POINTS
-900,-105,0
-450,-105,0
-120,-25,0
ENTER SPEED ON ALL SEGMENTS
35,35
TYPE AND RADIUS OF RETURN LOOP
1,100

SOURCE TYPE?
LOADER
MODEL NUMBER?
5
ENTER 1, 2, OR 3 FOR WORKING OVER POINT, LINE OR AREA:
1
HOURS WORKED:
-1
ENTER X,Y,Z OF SOURCE LOCATION
-390,-355,0
```

Figure 6. Example of HINPUT Run: Cut Area.
a. Data Input at Terminal.

```
SOURCE TYPE?
TRUCKS
MODEL NUMBER?
3
HOURS WORKED:
-1
HOW MANY POINTS IN LINE?
3
ENTER X,Y,Z OF POINTS
-1000,-105,0
-850,-105,0
-410,-340,0
ENTER SPEED ON ALL SEGMENTS
35,35
TYPE AND RADIUS OF RETURN LOOP
5,150

SOURCE TYPE?
BULLDOZER
MODEL NUMBER?
3
ENTER 1, 2, OR 3 FOR WORKING OVER POINT, LINE OR AREA:
2
HOURS WORKED:
8
HOW MANY POINTS IN LINE?
2
ENTER X,Y,Z OF POINTS
-30,-80,0
15,-200,0

SOURCE TYPE?
BULLDOZER
MODEL NUMBER?
3
ENTER 1, 2, OR 3 FOR WORKING OVER POINT, LINE OR AREA:
2
HOURS WORKED:
8
HOW MANY POINTS IN LINE?
2
ENTER X,Y,Z OF POINTS
-50,-80,0
-25,-220,0
```

Figure 6 a. (Continued)

```
SOURCE TYPE?
BULLDOZER
MODEL NUMBER?
3
ENTER 1, 2, OR 3 FOR WORKING OVER POINT, LINE OR AREA:
2
HOURS WORKED:
8
HOW MANY POINTS IN LINE?
2
ENTER X,Y,Z OF POINTS
-355,-400,0
-200,-490,0

SOURCE TYPE?
BULLDOZER
MODEL NUMBER?
3
ENTER 1, 2, OR 3 FOR WORKING OVER POINT, LINE OR AREA:
2
HOURS WORKED:
8
HOW MANY POINTS IN LINE?
2
ENTER X,Y,Z OF POINTS
-355,-410,0
-285,-550,0

SOURCE TYPE?

HOW MANY BARRIERS?
0

ENTER TITLE
 I-210 CUT AREA, RECEIVERS C3,C4,C6,C7
DATA FILE NAME - FILNAM.DAT
CUT210.DAT
```

Figure 6 a. (Continued)

```
.TYPE DX1:CUT210.DAT

 I-210 CUT AREA, RECEIVERS C3,C4,C6,C7
  4          RECEIVERS
   -500.00    -610.00       0.00
   -460.00    -300.00       0.00
   -200.00     300.00       0.00
      0.00       0.00       0.00
  2          POINT SOURCES
   -110.00     -65.00       6.00      80.00         500            LOADER  1
   -390.00    -355.00       6.00      80.00         500            LOADER  2
  6          LINE SOURCES
 10     500            TRUCKS  1
   -900.00    -105.00       6.00      86.00 0.0003241
   -450.00    -105.00       6.00      86.00 0.0003241
   -354.22     -81.78       6.00      86.00 0.0001624
   -229.29     124.46       6.00      85.23 0.0001771
   -167.12     169.37       6.00      84.13 0.0002010
    -91.29     157.91       6.00      82.64 0.0002385
    -46.37      95.75       6.00      80.34 0.0003108
    -57.83      19.91       6.00      72.69 0.0007502
   -120.00     -25.00       6.00      86.00 0.0004232
   -354.22     -81.78       6.00       0.00 0.0000000
 11     500            TRUCKS  2
  -1000.00    -105.00       6.00      86.00 0.0003241
   -850.00    -105.00       6.00      86.00 0.0003241
   -728.87    -169.69       6.00      86.00 0.0001652
   -586.08    -125.22       6.00      84.40 0.0001947
   -471.65    -137.02       6.00      82.10 0.0002537
   -398.20    -225.57       6.00      74.45 0.0006126
   -410.00    -340.00       6.00      86.00 0.0006126
   -498.54    -413.45       6.00      86.00 0.0002537
   -612.98    -401.65       6.00      86.00 0.0001947
   -686.43    -313.10       6.00      86.00 0.0001652
   -728.87    -169.69       6.00       0.00 0.0000000
  2     500            BULLDOZER    1
    -30.00     -80.00       6.00      83.00 0.0078027
     15.00    -200.00       6.00       0.00 0.0000000
  2     500            BULLDOZER    2
    -50.00     -80.00       6.00      83.00 0.0070316
    -25.00    -220.00       6.00       0.00 0.0000000
  2     500            BULLDOZER    3
   -355.00    -400.00       6.00      83.00 0.0055793
   -200.00    -490.00       6.00       0.00 0.0000000
  2     500            BULLDOZER    4
   -355.00    -410.00       6.00      83.00 0.0063888
   -285.00    -550.00       6.00       0.00 0.0000000
 0           AREA SOURCES
 0           BARRIERS
     1.00DB/DD EXCESS ATTENUATION
```

Figure 6. b. Data File Created By HINPUT.

```
.RUN HINPUT
HOW MANY RECEIVERS?
3
INPUT X,Y,Z OF EACH RECEIVER
0,100,4
0,-100,4
100,0,4
EXCESS ATTENUATION, DB/DD
1

SOURCE TYPE?
LOADER
MODEL NUMBER?
2
ENTER 1, 2, OR 3 FOR WORKING OVER POINT, LINE OR AREA:
1
HOURS WORKED:
8
ENTER X,Y,Z OF SOURCE LOCATION
0,0,0

SOURCE TYPE?
TRUCKS
MODEL NUMBER?
2
HOURS WORKED:
-1
HOW MANY POINTS IN LINE?
2
ENTER X,Y,Z OF POINTS
-500,0,0
0,0,0
ENTER SPEED ON ALL SEGMENTS
30
TYPE AND RADIUS OF RETURN LOOP
7,25

SOURCE TYPE?

HOW MANY BARRIERS?
2
HOW MANY POINTS?
2
ENTER X,Y,Z OF POINTS
-100,50,15
50,50,15
HOW MANY POINTS?
3
ENTER X,Y,Z OF POINTS
-100,-50,15
0,-50,15
50,-50,15

ENTER TITLE
 BARRIER EXAMPLE CASE
DATA FILE NAME - FILNAM.DAT
BAREX.DAT
```

Figure 7. Example of HINPUT Run: Fill Area.
a. Data Input at Terminals.

```
.TYPE DX1:BAREX.DAT

 BARRIER EXAMPLE CASE
  3          RECEIVERS
      0.00     100.00       4.00
      0.00    -100.00       4.00
    100.00       0.00       4.00
  1          POINT SOURCES
      0.00       0.00       6.00      76.00        500            LOADER  1
  1          LINE SOURCES
  3    500          TRUCKS  1
   -500.00       0.00       6.00      75.39 0.0003710
      0.00       0.00       6.00      81.00 0.0003710
   -500.00       0.00       6.00       0.00 0.0000000
  0          AREA SOURCES
  2          BARRIERS
  2          POINTS
   -100.00      50.00      15.00
     50.00      50.00      15.00
  3          POINTS
   -100.00     -50.00      15.00
      0.00     -50.00      15.00
     50.00     -50.00      15.00
      1.00DB/DD EXCESS ATTENUATION
```

Figure 7. b. Data File Created by HINPUT.

```
.RUN HINPUT
HOW MANY RECEIVERS?
3
INPUT X,Y,Z OF EACH RECEIVER
0,0,4
0,70,4
-50,340,4
EXCESS ATTENUATION, DB/DD
1

SOURCE TYPE?
TRUCKS
MODEL NUMBER?
3
HOURS WORKED:
8
HOW MANY POINTS IN LINE?
4
ENTER X,Y,Z OF POINTS
350,270,0
-340,260,0
-340,120,0
350,120,0
ENTER SPEED ON ALL SEGMENTS
30,20,10
VEHICLES PER HOUR:
55
TYPE AND RADIUS OF RETURN LOOP
0,0
STOPPING AND DECELERATION POINTS
0,0

SOURCE TYPE?
BULLDOZER
MODEL NUMBER?
3
ENTER 1, 2, OR 3 FOR WORKING OVER POINT, LINE OR AREA:
3
HOURS WORKED:
8
HOW MANY POINTS DEFINE CENTERLINE?
2
ENTER X,Y,Z AND WIDTH OF POINTS
-290,140,0,80
450,140,0,80
HOW MANY?
1

SOURCE TYPE?

HOW MANY BARRIERS?
0

ENTER TITLE
 I-210 FILL AREA, RECEIVERS F6,F7,F8
DATA FILE NAME - FILNAM.DAT
FIL210.DAT
```

Figure 8. Example of HINPUT Run With Barriers.
a. Data Input at Terminal.

```
.TYPE DX1:FIL210.DAT

 I-210 FILL AREA, RECEIVERS F6,F7,F8
  3            RECEIVERS
      0.00        0.00        4.00
      0.00       70.00        4.00
    -50.00      340.00        4.00
  0            POINT SOURCES
  1            LINE SOURCES
  4     500          TRUCKS  1
    350.00      270.00        6.00       86.00 0.0003464
   -340.00      260.00        6.00       86.00 0.0005197
   -340.00      120.00        6.00       86.00 0.0010393
    350.00      120.00        6.00        0.00 0.0000000
  1            AREA SOURCES
  2     500          BULLDOZER   1
   -290.00      140.00        6.00       80.00       83.00
    450.00      140.00        6.00       80.00       -0.00
  0            BARRIERS
     1.00DB/DD EXCESS ATTENUATION
```

Figure 8. b. Data File Created by HINPUT.

Program-generated return loops were specified for both haul operations. Three road points were sufficient to describe each approach, with the last corresponding to the loading point. The speeds input are the approach cruise speeds.

Figure 7 corresponds to the fill area for this same operation. The trucks passed to one side of the fill area, turned around, and returned parallel to the approach but about 140 feet from it. They slowed through the area. A bulldozer working over an 80-foot-wide area spread the dumped earth. The first two receivers are near the dump/spread operation. The third is near the haul road approach.

Figure 7 shows a hypothetical operation with barriers. The two barriers input are symmetric about the haul/load operation. They are physically identical, but are specified with different numbers of points. A straight in-and-out return loop is specified; the 25-foot radius given is irrelevant. Two of the receivers are shielded by the barriers, and one is not.

The data file contains the input data converted to acoustical quantities. It is presented in the order receiver data, point source data, line source data, area source data, barrier data, and excess attenuation. The quantities created and the FORTRAN formats for the data items are:

- Title line.
- Number of Receivers: I3.
- Receiver coordinates (X, Y, Z): 3F10.2 for each.
- Number of point sources: I3.
- Source location (X, Y, Z), effective emission level, and nominal frequency: 4F10.2, I10 for each. Note that the Z value has the effective acoustic height added to the input Z.
- Number of line sources: I3.
- For each line source, the following:
  - Number of points and nominal frequency: I3, I7.
  - Point coordinates (X, Y, Z), vehicle passby level ($L_{eq(3)}$), and source density (N): 4F10.2, F10.7 for each. The coordinate points printed in Figures 6b and 8b are the loop points created by the program, as requested in Figures 6 and 8a. The emission levels and source densities correspond to the segment beginning at the point on the same line; there is one fewer of these than there are points.

- Number of area sources: I3.
- For each area source, the following:
  - Number of points and nominal frequency: I3, I7.
  - Center point coordinates (X, Y, Z), width, and effective source emission level: 5F10.2. There is one fewer emission level than there are points.
- Number of barriers: I3.
- For each barrier, the following:
  - Number of points: I3.
  - Coordinates (X, Y, Z) of points: 3F10.2.
- Excess attenuation: F10.2.

Various identifying information is also printed in the data files. These labels are not read by HICNOM. Their formats may be seen in the program listings, Appendix A, and are available for use if it is desired to modify HICNOM to make use of them. Their main intent is to make the data file readable. Experienced users of this model may want to modify this file to make minor changes, rather than reenter all the data through HINPUT.

4.3.2 Running HICNOM

Following the creation of a data file by HINPUT, enter the command RUN HICNOM. HICNOM will request the data file name. It will then run and print the results. Figures 9, 10, and 11 are HICNOM outputs coresponding to the HINPUT runs of Figures 6, 7, and 8. The outputs consist of two parts: the total noise level at each receiver point, and the contribution of each source component to the noise at each receiver. Shown in parentheses next to the total level in Figures 9 and 10 are the measured levels. Agreement is generally quite good, within 3 dB for most points. Larger errors are present for one boundary point in the cut area, and for the receiver near the haul road approach in the fill. The first is most likely due to the less precise definition of activities with regard to more distinct receiver points; this is a general problem with any model of this type. The overprediction for the truck approach in the fill area is probably because the trucks were decelerating there, rather than cruising. Again, this is due to availability of less than fully complete data: attention in the field was concentrated on the more complex fill area itself.

The component contribution for each receiver consists of a source identifying index, an intensity quantity, and the sound pressure level due to that source. The intensity

```
.RUN HICNOM
FILE NAME - FILNAM.DAT
CUT210.DAT
I-210 CUT AREA, RECEIVERS C3,C4,C6,C7

RECEIVER NUMBER          LEQ


         1               71.7      (76.8)
         2               77.7      (75.2)
         3               67.7      (69.9)
         4               77.6      (75.4)

COMPONENT CONTRIBUTIONS FOR RECEIVER NUMBER:  1

INDEX    INTENSITY     LEVEL

    1    0.234334E+06  53.7
    2    0.183035E+07  62.6
  101    0.222040E+06  53.5
  201    0.524362E+05  47.2
  301    0.387578E+05  45.9
  401    0.673703E+04  38.3
  501    0.525163E+04  37.2
  601    0.447375E+04  36.5
  701    0.395766E+04  36.0
  801    0.203519E+04  33.1
  901    0.117194E+06  50.7
  102    0.471751E+05  46.7
  202    0.655366E+05  48.2
  302    0.477432E+05  46.8
  402    0.313780E+05  45.0
  502    0.308269E+05  44.9
  602    0.230731E+05  43.6
  702    0.764877E+06  58.8
  802    0.403026E+06  56.1
  902    0.150495E+06  51.8
 1002    0.700563E+05  48.5
  103    0.450208E+06  56.5
  104    0.498875E+06  57.0
  105    0.362092E+07  65.6
  106    0.590726E+07  67.7
```

Figure 9. HICNOM Output: Cut Area.

COMPONENT CONTRIBUTIONS FOR RECEIVER NUMBER: 2

INDEX	INTENSITY	LEVEL
1	0.691125E+06	58.4
2	0.260274E+08	74.2
101	0.116851E+07	60.7
201	0.424937E+06	56.3
301	0.177960E+06	52.5
401	0.194691E+05	42.9
501	0.136628E+05	41.4
601	0.110588E+05	40.4
701	0.991602E+04	40.0
801	0.562367E+04	37.5
901	0.538335E+06	57.3
102	0.901511E+05	49.5
202	0.176742E+06	52.5
302	0.233958E+06	53.7
402	0.300064E+06	54.8
502	0.623884E+06	58.0
602	0.114944E+07	60.6
702	0.104607E+08	70.2
802	0.100633E+07	60.0
902	0.354260E+06	55.5
1002	0.220828E+06	53.4
103	0.101192E+07	60.1
104	0.118429E+07	60.7
105	0.671521E+07	68.3
106	0.673944E+07	68.3

Figure 9 (Continued).

COMPONENT CONTRIBUTIONS FOR RECEIVER NUMBER: 3

INDEX	INTENSITY	LEVEL
1	0.902972E+06	59.6
2	0.224956E+06	53.5
101	0.172422E+06	52.4
201	0.789180E+05	49.0
301	0.312672E+06	55.0
401	0.335818E+06	55.3
501	0.293800E+06	54.7
601	0.113642E+06	50.6
701	0.453690E+05	46.6
801	0.137715E+05	41.4
901	0.408403E+06	56.1
102	0.276992E+05	44.4
202	0.335577E+05	45.3
302	0.260056E+05	44.2
402	0.237140E+05	43.8
502	0.186938E+05	42.7
602	0.562813E+04	37.5
702	0.553991E+05	47.4
802	0.184254E+05	42.7
902	0.139480E+05	41.4
1002	0.181719E+05	42.6
103	0.104114E+07	60.2
104	0.105542E+07	60.2
105	0.360114E+06	55.6
106	0.321589E+06	55.1

Figure 9 (Continued).

COMPONENT CONTRIBUTIONS FOR RECEIVER NUMBER: 4

INDEX	INTENSITY	LEVEL
1	0.112013E+08	70.5
2	0.409873E+06	56.1
101	0.150200E+06	51.8
201	0.943789E+05	49.7
301	0.249631E+06	54.0
401	0.108637E+06	50.4
501	0.144980E+06	51.6
601	0.310782E+06	54.9
701	0.965282E+06	59.8
801	0.319072E+06	55.0
901	0.146360E+07	61.7
102	0.212396E+05	43.3
202	0.274941E+05	44.4
302	0.231234E+05	43.6
402	0.237204E+05	43.8
502	0.251423E+05	44.0
602	0.945181E+04	39.8
702	0.895524E+05	49.5
802	0.256098E+05	44.1
902	0.164707E+05	42.2
1002	0.177854E+05	42.5
103	0.224285E+08	73.5
104	0.178385E+08	72.5
105	0.819356E+06	59.1
106	0.663551E+06	58.2

STOP --

Figure 9 (Concluded).

```
.RUN HICNOM
FILE NAME - FILNAM.DAT
FIL210.DAT
I-210 FILL AREA, RECEIVERS F6,F7,F8

RECEIVER NUMBER           LEQ


          1                73.8      (72.4)
          2                79.0      (76.8)
          3                72.9      (63)


COMPONENT CONTRIBUTIONS FOR RECEIVER NUMBER:   1

INDEX   INTENSITY    LEVEL

  101   0.130053E+07  61.1
  201   0.240830E+06  53.8
  301   0.144167E+08  71.6
10101   0.783708E+07  68.9


COMPONENT CONTRIBUTIONS FOR RECEIVER NUMBER:   2

INDEX   INTENSITY    LEVEL

  101   0.221696E+07  63.5
  201   0.286105E+06  54.6
  301   0.511811E+08  77.1
10101   0.256987E+08  74.1


COMPONENT CONTRIBUTIONS FOR RECEIVER NUMBER:   3

INDEX   INTENSITY    LEVEL

  101   0.944726E+07  69.8
  201   0.362711E+06  55.6
  301   0.536970E+07  67.3
10101   0.420221E+07  66.2
STOP --
```

Figure 10. HICNOM Output: Fill Area.

```
.RUN HICNOM
FILE NAME - FILNAM.DAT
BAREX.DAT
BARRIER EXAMPLE CASE

RECEIVER NUMBER          LEQ


          1              58.9
          2              58.9
          3              69.4


COMPONENT CONTRIBUTIONS FOR RECEIVER NUMBER:  1

INDEX    INTENSITY    LEVEL

    1    0.472863E+06  56.7
  101    0.638076E+05  48.0
  201    0.232206E+06  53.7

COMPONENT CONTRIBUTIONS FOR RECEIVER NUMBER:  2

INDEX    INTENSITY    LEVEL

    1    0.472863E+06  56.7
  101    0.638077E+05  48.0
  201    0.232206E+06  53.7

COMPONENT CONTRIBUTIONS FOR RECEIVER NUMBER:  3

INDEX    INTENSITY    LEVEL

    1    0.789945E+07  69.0
  101    0.173481E+06  52.4
  201    0.631323E+06  58.0
STOP --
```

Figure 11. HICNOM Output: Barrier Example.

quantity is $10^{L/10}$, where L is the sound level. The identifying index may be decoded as follows:

- Point sources have one or two digits, line sources three or four, and area sources five.
- The last two digits are the sequential number of the source, in the order it appears in the intermediate data file. Separate counts, from 1, exist for points, lines, and areas.
- For line and area sources, the next two digits identify the source segment, in the order they appear in the data file. Separate counts exist for each source. The level corresponds to that segment only.
- Area sources are identified by a 1 in the fifth digit (leading digit if present).

Note that the order of each source component list corresponds exactly to the order in which the components appear in the data file read by HICNOM.

5.0 PROGRAMMER'S MANUAL

5.1 Conventions and Annotation

The programs are written in FORTRAN IV. Standard default conventions apply to variable types. All real variables are single precision, R*4. All fixed-point variables are I*2. Two arrays in subroutine DECODE are declared LOGICAL*1 to simplify string manipulations. No proprietary routines, machine-dependent instructions, etc., are utilized. The program is fully transportable to other machines provided the data file specification in HINPUT and HICNOM is changed to conform with available hardware.

Extensive use is made of subroutines in order to provide reasonably well-structured code. Variable names are reasonably consistent between subroutines, although in many cases the mnemonics are similar but not identical. Executable statement labels begin with 1 in each program; format labels begin at 100. Variable names and formats are identical in HINPUT and HICNOM. This was done so as to facilitate future merging of the two programs for use on a larger machine. Executable statement labels were not made non-conflicting between the two; this is to discourage casual merging without a thorough review of the combined program. COMMON blocks are consistent throughout both programs.

All routines are heavily commented so as to facilitate review of code. A complete variable dictionary is included in each program.

5.2 Subroutines

Figures 12 and 13 are subroutine hierarchy charts for the two programs. The main programs are described in Section 3.2. The following subsections describe each subroutine, presented in the order they appear in the hierarchy charts. The descriptions here and in Section 3.2 are intended to be read together with the source listings in Appendix A if a detailed understanding of the code is desired.

5.2.1 DECODE (SRCNAM, IGEOM, INFO)

This subroutine checks an input name SRCNAM against a data list of allowable names. The corresponding type identification number is placed in INFO(2). The model number is requested and placed in INFO(1). IGEOM is set to indicate whether it is a point, line, or area source. The input name SRCNAM is appended with a sequential number.

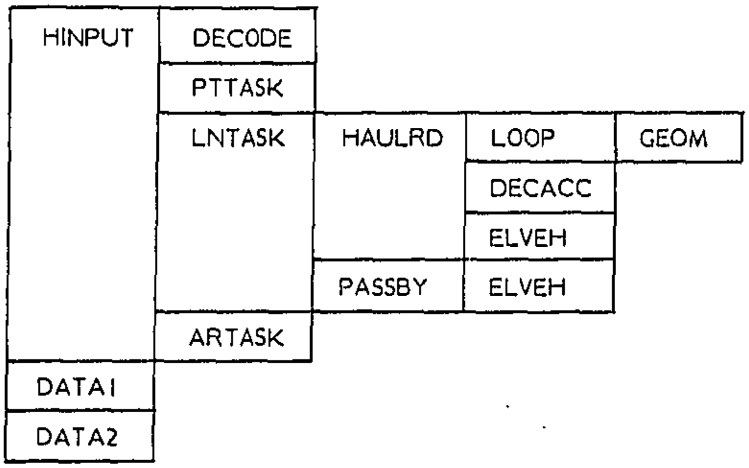

Figure 12. Subroutine Hierarchy Chart, HINPUT.

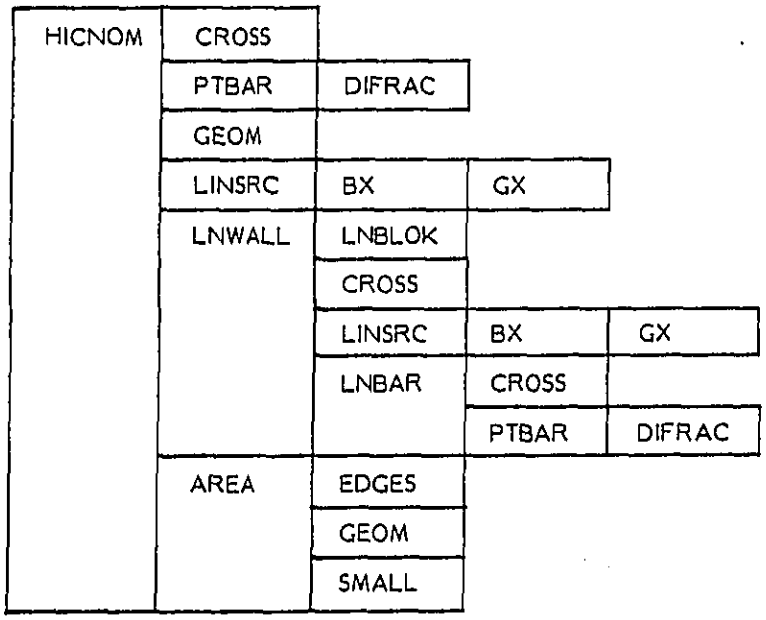

Figure 13. Subroutine Hierarchy Chart, HICNOM.

5.2.2 PTTASK (ELPT, Z, IFRQ)

This routine returns the L_{eq} emission level and effective source height (input height plus acoustic height) for point sources. These data are obtained from data files or input for newly created sources. The production rate per full workday is computed for appropriate pieces of equipment. Hours worked per day is then used to compute daily production. This is saved as PROD. If a "balance to last" case has been specified, equivalent hours worked per day are computed by matching production to the value of PROD left from the last piece of equipment. The matching procedure and value of PROD here are shared in common with LNTASK and ARTASK.

5.2.3 LNTASK (XLNSRC, NLNPTS, ELLN, EN, IFRQ)

This routine handles line tasks. It is divided into two parts. The first part handles non-haul equipment, and is logically similar to PTTASK. One piece of equipment is considered, moving at a single average speed, and production is based on the number of hours worked.

The second part of LNTASK handles haul operations. Production is related to number of vehicles per hour. Speeds can vary from segment to segment. If a loop is to be generated by the program, this is done by a call to HAULRD which returns an increased value of NLNPTS and an expanded array XLNPTS. Adding a loop can add 8 or more points to the array. To ensure that the limit of 20 points is not exceeded, the input line array should not exceed 6 points if a loop is to be generated.

If a loop is not to be created, the acceleration/deceleration profiles are handled by a call to PASSBY.

5.2.4 HAULRD (RDPTS, IRDPTS, RAD, ILOOP, SPEED, Q, IVEH, EN, ELL)

This routine is called when a turnaround loop is to be generated. The loop points are added to RDPTS by subroutine LOOP, making the last input point the loading/unloading point. Speed profiles are generated by DECACC, which may add additional points so as to complete the acceleration/deceleration profiles if they extend beyond the loop points. Emission levels are obtained from the function ELVEH, which accounts for operating mode and speed.

5.2.5 LOOP (RDPTS, IRDPTS, RAD, ITYPE, NRDPTS, NLOAD)

This routine generates turnaround loops as shown in Figure 3. Two basic geometries are stored: a half loop (types 1 through 4) and a full circle (types 5 and 6). The loop data are stored in the orientation shown in Figure 2. Subroutine GEOM is called to determine the angle of the last input road segment. The loop data are transformed through this angle, so as to be aligned with the last segment. They are multiplied by the input radius and then combined with RDPTS. Various details of adding the points, setting direction, etc., are described by comments in the listing.

5.2.6 GEOM (XI, X2, X0BS, D, PHI, ROTCS, OBSROT, RDLNTH, ISKIP)

This routine, used by both HINPUT and HICNOM, performs a coordinate transformation of a line segment defined by XI, X2. The transformation is that described in Section 3.2.2. Quantities returned are the cosine and sine of the transformation rotation angle, the transformed coordinates of one receiver position, d, ϕ_1, and ϕ_2 as defined in Figure 1, and the length of the line segment. A switch variable ISKIP permits the use of only parts of this routine.

5.2.7 DECACC (RDPTS, NRDPTS, IRDPTS, NLOAD, IDEC, SPEED, V)

This routine computes average speeds on a turnaround loop. Kinematics are based on constant deceleration and acceleration rates stored in the two-element array ACCRAT. Values of 0.1g are in data statements. The kinematic calculations begin at the stopping point and continues until the speed on a segment matches the input speed on that segment. If necessary, the segments outside the loop are split into separate acceleration and deceleration segments. The expanded point array defining the lines and the computed average speeds are returned.

5.2.8 ELVEH (IVEH, V, MODE)

This function subroutine computes the emission level of haul equipment from Equation (12).

5.2.9 PASSBY (XLNPTS, NLNPTS, ISTOP, IDEC, SPEED, Q, IVEH)

This routine prepares haul equipment speeds and emission levels when the haul road has been directly defined by the user, i.e., a loop is not generated by the program. Where kinematics are to be computed, the calculation is essentially the same as in DECACC.

5.2.10 ARTASK (CLPTS, WIDTH) NCLPTS, ELAR, IFRQ)

This routine computes the emission intensity in each section of an area. The logic is essentially the same as in PTTASK and the non-haul section of LNTASK. A number of pieces of equipment can be specified.

5.2.11 <u>DATA1</u>

This block data routine contains various data items described in Section 6.1, including unit declarations and reference distance.

5.2.12 <u>DATA2</u>

This block data routine contains all equipment data. The program is commented and tabulated so as to provide an easily read table of values. See Section 6.3 for a detailed description.

5.2.13 <u>CROSS (XOBS, XSRC, B1, B2, X, ICROSS)</u>

This subroutine determines if a line between a source and receiver intersects with a barrier segment. If so, the crossing point and the barrier height at that point are computed.

5.2.14 <u>PTBAR (XOBS, XSRC, SBAR, IFREQ, ATTEN)</u>

Given the source and receiver locations and the barrier height at the shielding point, this routine computes path length difference, then obtains barrier shielding from subroutine function DIFRAC.

5.2.15 <u>DIFRAC (IFREQ, DELTA)</u>

This function routine computes Fresnel number from path length difference and frequency, then obtains barrier shielding from Maekawa's curve. For Fresnel number between -0.3 and 1.0, a table look-up/interpolation scheme is used. Above 1.0, a logarithmic approximation is used.

5.2.16 <u>LINSRC (D, EYE, EN, PHI, GNDA, EYEEQ, OBS, RLEN)</u>

This routine computes L_{eq} from a single straight line segment, using Equations (2) and (8). The program contains an alternate expression for the case of a receiver directly in line with the line source, and contains some error checking of input geometry.

5.2.17 <u>BX (XIN, AIN, BIN)</u>

This function routine computes the incomplete beta function from the power series Equation (9), using the methodology described in Section 2.1.2. A convergence criterion of one part in 10^4 is specified as a data item.

5.2.18 <u>GX (XX, IER)</u>

This function subroutine computes the Gamma function for real arguments, using a polynomial approximation and a recursion.

5.2.19 LNWALL (OBSROD, XLNSRC, IFRQ, GNDA, EYESRC, EN, ROTCS, PHI, RDLNTH, BARPTS, NBAR, NBPTS, EYE)

This subroutine computes the contribution of a line source in the presence of barriers. The input parameters are one line source section, one receiver location, and the set of barrier coordinates. Subroutine LNBLOK is called to test whether a barrier shields the source/receiver combination. If so, the line segment is divided into three parts: a center shielded section and two unshielded ends. Any of these may have zero length. LINSRC is used to compute noise from the unshielded sections and LNBAR is used to compute noise from the shielded part. Logic is included to properly handle multiple-segment barriers. If no shielding is found, a very large number is returned to HICNOM. The HICNOM logic which takes the smaller of the shielded versus unshielded calculation then selects the unshielded result from LINSRC.

5.2.20 LNBLOK (OBSROD, PHI3, XI, BAR, IEND, ROTCS, BROT)

This subroutine tests whether any of the input barriers shield a given line source segment and receiver. If so, the line segment is divided into three segments as described in Section 5.2.19. Each segment is defined by a pair of angles ϕ_1, ϕ_2 as shown in Figure 1. Setting the angles equal to each other indicates no segment. The program contains logic to ensure that the sense of the angles corresponds to the convention following the order of line points.

5.2.21 LNBAR (D, PHI, Z, RDLNTH, OBS, B, IFRQ, ATTEN)

This routine computes the barrier shielding of a shielded line segment. A numerical integrating process is used, utilizing point source shielding computed by PTBAR and DIFRAC. This method is described in detail in Section 2.1.4.

5.2.22 AREA (NCLPTS, CLPTS, WIDTH, OBSPTS, OBSROT, NOBS, EYETOT, EYESTR, STRIPL, STRIPR, NSTR, CLLNTH)

This subroutine divides an area into strips, as described in Section 2.1.3. The widths are positions such that they bisect the normals to the centerline segments on either side. Subroutine EDGES is used to determine the coordinates of the resultant corners. Each area segment and all receivers are transformed into coordinates relative to the segment centerline. Each segment is then divided into strips consisting of the centerline plus 0 to 5 strips each to the right and left of the centerline. The average source density on the strips is obtained by dividing total strength in the segment by the length of all strips in the segment.

5.2.23 EDGES (NCLPTS, CLPTS, WIDTH)

This subroutine is used by AREA to find the coordinates of the edge points defined by the width lines, as described in Section 5.2.22.

5.2.24 SMALL (A, N, J)

This routine finds the smallest value in an array.

5.3 COMMON Blocks

The following COMMON blocks and their contents are used in the programs:

- /CONSTS/ PI, TWOPI, PIOV2
- /EDGE/ EDGEL (3, 10), EDGER (3, 10)
- /EQUIPT/ EQUIP (5, 10, 30), IFREQ (10, 30)
- /KINEM/ ACCRAT (2)
- /NMODLS/ NMODS (30), NVTYP
- /TSKARG/ INFO (2), HOURS, PRODUC
- /TYPES/ IPROD (30), IHAUL (10), IVEH (5, 5)
- /UNITS/ D0, D02
- /VEHLEV/ HAULEQ (6, 10)
- /WKDAY/ DAYHRS

The variable names vary somewhat between subroutines. In particular, EQUIP in block data routine DATA2 appears as a large number of smaller arrays, each named for the corresponding equipment.

Figure 14 is a chart showing the location of the COMMON blocks in the programs. An open circle indicates the COMMON is in the program. A solid circle indicates that some or all of the contents of the block are defined by data statements within that routine. Note that data statements for /UNITS/ are in both HINPUT and DATA1. If the programs are merged, one set of these data statements would have to be deleted.

	COMMON Block									
Program	CONSTS	EDGE	EQUIPT	KINEM	NMODLS	TSKARG	TYPES	UNITS	VEHLEV	WKDAY
HINPUT						O				
HICNOM								●		
DECODE					O					
PTTASK			O		O	O	O			O
LNTASK			O		O	O	O	O	O	O
HAULRD										
LOOP										
GEOM	●									
DECACC				O				O		
ELVEH								O	O	
PASSBY				O				O		
ARTASK			O		O	O	O			O
DATA1				●				●		●
DATA2			●		●		●		●	
CROSS										
PTBAR										
DIFRAC										
LINSRC	O							O		
BX										
GX										
LNWALL										
LNBLOK	O									
LNBAR										
AREA		O								
EDGES	O	O								
SMALL										

Figure 14. COMMON Block Locations.

6.0 MAINTENANCE MANUAL

Program maintenance consists of modifications to the data statements as additional data become available and/or to change units and constants. Data are contained in two block data programs, DATA1 and DATA2. Distance units for barrier calculations are set by a data item in DIFRAC. Additional equipment types may be defined, with names incorporated into DECODE.

6.1 BLOCK DATA DATA1

This routine contains several constants which implicitly set units. They are:

- D0 and D02, the sound level reference distance and distance squared. Current values are 50 and 2500, representing a 50-foot reference distance. Setting these to a reference distance, inputting coordinates in the same units, and providing appropriate levels in DATA2 (described in Section 6.3) automatically set the units for all non-barrier calculations.
- GRAV is the value of the acceleration of gravity in the units being used. A value of 32.2 has been used, for ft/sec^2. GRAV converts the values of ACCRAT from fractions of a g to physical units. If ACCRAT is left as fraction of a g (0.1 in the program presented here), GRAV should be changed to the appropriate value for other units. Alternatively, GRAV may be set equal to 1.0 and ACCRAT expressed in physical units.
- VELCON is a conversion factor from speed in units convenient for input to consistent physical units. A value of 1.47 has been used, converting speed in mph to feet per second. If metric units are adopted, km/hr for input speed and meters for distance, VELCON should be changed to 0.278.
- DAYHRS is the number of hours in a full work day. The program computes L_{eq} based on this time. A value of 8 hours is in the program.

6.2 SUBROUTINE DIFRAC

The Fresnel number is given by $2\delta f/c$, where c is the speed of sound. The program has a data item VOV2, representing c/2, which sets the units. A value of 580 has been used, setting the units as feet when frequency IFREQ is in Hz. To change units, replace VOV2 with c/2 in the appropriate units. For meters, this is 176.8.

6.3 BLOCK DATA DATA2

This data routine contains the equipment data base. Definitions may be found in the variable dictionaries in PTTASK, LNTASK, and ARTASK. The data statements are arranged and commented so as to form tables, with the main data table corresponding to Table 2. The annotation in columns 73 to 80 identifies the equipment. A * next to the annotation indicates the level was obtained from a review of the literature, Reference 3.

Modification of the basic data bases, HAULEQ and EQUIP (split into smaller individually named arrays in this program) is straightforward. Additional lines for new models may be added, up to a total of 10 for each type. The tabular format should be retained for ease in reading. When adding new lines, the zeros and the value of NMODS on the last continuation line of each set should be appropriately adjusted. New types may be defined, making room by reducing the size of the dummy arrays EMPTY1, EMPTY2, and EMPTY3. Corresponding new names must be added to DECODE, described below. Default geometries are point sources for the first 15 data blocks, line for the next 10, and area for the last 5, where the block number corresponds to the third index of EQUIPT. Alternate geometries may be specified in DECODE.

Haul equipment has its type numbers in sequence with non-haul, but the data are contained in the array HAULEQ. The array IVEH provides a mapping from EQUIP type and model numbers to HAULEQ; see the listing of LNTASK for the specific relationship. Up to 5 models are permitted for each type. Equipment types which are haul are listed in array IHAUL. This is a grocery list arrangement, with type numbers listed in any order. Array IPROD is a similar list which specifies which equipment have production rates associated with them.

The spectrum frequency data required by the barrier calculation is contained in array IFREQ, a two-dimensional array which serves as an additional column to EQUIP. Haul and non-haul data are in the array. The separate array – rather than another column on EQUIP – is used to permit these data to be in integer form. This will facilitate any future replacement of DIFRAC with a routine using exact calculations for particular spectra. IFREQ may then be treated as an index with no other recoding.

6.4 SUBROUTINE DECODE

This routine contains the list of equipment type names. Adding additional names is simply a matter of replacing the 'XXXX' blocks in the data statement for array ALNAMS. Three 4-character blocks are permitted. In addition to checking the name, an ordinal index from array NTH is appended. Its position is determined by data in arrays IPOS and

NPOS. They are set up so as to place the index in the thirteenth position of any added names. If it is desired to adjust the number of blanks (as has been done for the existing names), values of IPOS and NPOS should be changed. NPOS indicates in which four character block the the index is placed, and IPOS is the position.

As noted above, model types 1 to 15 are nominally point sources, 16 to 25 are line, and 26 to 30 are area. The array NGEOM is a grocery-list of models which may have alternate geometries. By adding a model number to the NGEOM data statement, the program will request the geometry from the user.

REFERENCES

1. "Definition of Typical Highway Types", Working Paper No. 1, DeLeuw, Cather & Company.

2. Barraza, J.A., "Definition of Highway and Bridge Construction Phases", Working Paper No. 2, DeLeuw, Cather & Company, January 1979.

3. Fuller, W.R., Brown, R., Marcus, W.E., and Plotkin, K.J., "Literature Review: Highway Construction Noise", Wyle Research Report WR 79-3, June 1979.

4. Brown, R., Fuller, W.R., and Plotkin, K.J., "Test Plan For Measurement and Analysis of Highway Construction Noise", Wyle Research Report WR 79-16, July 1979.

5. Fuller, W.R., "Analysis and Abatement of Highway Construction Noise", Wyle Research Report to be published.

6. Plotkin, K.J., "A Model for the Prediction of Highway Noise and Assessment of Strategies for its Abatement Through Vehicle Noise Control", Wyle Research Report WR 74-5, September 1974.

7. Barry, T.M., and Reagan, J.A., "FHWA Traffic Noise Prediction Model", FHWA-RD-77-108, December 1978.

8. Gordon, C.G., Galloway, W.J., Kugler, B.A., and Nelson, D.L., "Highway Noise – A Design Guide For Engineers", NCHRP Report 117, 1971.

9. Abramowitz, M., and Stegun, I.A. (eds.), Handbook of Mathematical Functions, Dover, New York, 1965.

10. Maekawa, Z.E., "Noise Reduction by Screens", Applied Acoustics, 1, pp. 157-173, 1971.

11. Kurze, U.J., and Anderson, G.S., "Sound Attenuation by Barriers", J.Applied Acoustics, 4, 1, pp. 35-53 (1971).

12. Plotkin, K.J., "Average Noise Levels For Highway Vehicles", Wyle Research Technical Note TN 79-2, September 1979.

13. Barnes, J.D., et al., "Power Plant Construction Noise Guide", prepared by BBN, Inc., for the Empire State Electric Energy Research Corporation, May 1977.

APPENDIX A

Computer Program Source Listings

```
      PROGRAM HINPUT
C THIS PROGRAM ACCEPTS HIGHWAY CONSTRUCTION DATA CONVERSATIONALLY AT
C THE TERMINAL AND PREPARES A DATA FILE TO BE READ BY 'HICNOM'.
C ALL VARIABLE NAMES, DIMENSIONS, AND FORMAT LABELS ARE COMPATABLE WITH
C HICNOM.  THIS WILL FACILITATE ANY FUTURE MERGING OF THE TWO PROGRAMS
C FOR USE ON A LARGER SYSTEM.
C
C DATA REQUIRED BY THE PROGRAM ARE REQUESTED AT THE TERMINAL.  TASK
C DATA ARE INPUT IN TERMS OF THE NAMES AND LOCATIONS OF EQUIPMENT INVOLVED.
C THE PROGRAM LOOKS UP NOISE AND OPERATIONAL DATA FROM A BUILT-IN DATA
C BASE.  THE END PRODUCT OF THIS PROGRAM IS AN ANNOTATED DATA FILE TO
C BE READ BY HICNOM, CONTAINING THE ACOUSTIC SOURCE EQUIVALENT OF THE
C CONSTRUCTION ACTIVITY DESCRIBED BY THE USER.
C
C VARIABLES:
C    ARNAM(4,5) = NAMES OF AREA SOURCES
C    BARPTS(3,5,3) = COORDINATES OF POINTS DEFINING BARRIERS
C    ELAR(10,5) = EMISSION LEVELS OF AREA SOURCES
C    ELLN(20,6) = EMISSION LEVELS OF LINE SOURCES
C    ELNAM(4,6) = NAMES OF LINE SOURCES
C    ELPT(10) = EMISSION LEVELS OF POINT SOURCES
C    EN(20,6) = VEHICLE DENSITY ON LINE SOURCES
C    FILNAM(4) = NAME OF OUTPUT FILE BEING PREPARED FOR 'HICNOM'
C    GNDA = EXCESS ATTENUATION FACTOR
C    HOURS = HOURS WORKED IN A TASK
C    I = DO LOOP INDEX
C    IARFRQ(5) = SPECTRUM IDENTIFIER FOR AREA SOURCE
C    ILNFRQ(6) = SPECTRUM IDENTIFIER FOR LINE SOURCE
C    IPTFRQ(10) = SPECTRUM IDENTIFIER FOR POINT SOURCE
C    IGEOM = INDEX DEFINING SOURCE GEOMETRY
C    II = DO LOOP INDEX
C    N= DO LOOP INDEX
C    NARSRC = NUMBER OF AREA SOURCES
C    NBAR = NUMBER OF BARRIERS
C    NBPTS(3) = NUMBER OF POINTS DEFINING BARRIER
C    NCLPTS(5) = NUMBER OF POINTS DEFINING CENTERLINE OF AREA SOURCES
C    NLNPTS(6) = NUMBER OF POINTS DEFINING LINE SOURCES
C    NLNSRC = NUMBER OF LINE SOURCES
C    NOBS = NUMBER OF RECEIVER LOCATIONS
C    NPTSRC = NUMBER OF POINT SOURCES
C    OBS(3,10) = COORDINATES OF OBSERVER LOCATIONS
C    PRODUC = DAILY PRODUCTION OF LAST TASK
C    PTNAM(4,10) = NAMES OF POINT SOURCES
C    SRCNAM(4) = INPUT SOURCE NAME
C    TITLE(20) = TITLE TO BE WRITTEN ONTO OUTPUT FILE
C    WIDTH(10,5) = WIDTH OF AREA SOURCE SEGMENTS
C    XCLPTS(3,10,5) = COORDINATES OF AREA SOURCE CENTER LINE POINTS
C    XLNSRC(20,6) = COORDINATES OF POINTS DEFINING LINE SOURCES
C    XPTSRC(3,10) = COORDINATES OF POINT SOURCE LOCATIONS
C
      DIMENSION TITLE(20),FILNAM(4),OBSPTS(3,10),XPTSRC(3,10),ELPT(10)
      DIMENSION NLNPTS(6),XLNSRC(3,20,6),ELLN(20,6),EN(20,6),NCLPTS(5)
      DIMENSION XCLPTS(3,10,5),WIDTH(10,5),ELAR(10,5),SRCNAM(4)
      DIMENSION PTNAM(4,10),ELNAM(4,6),ARNAM(4,5),FLNAM(3)
```

```
      DIMENSION NBPTS(3),BARPTS(3,5,3),IARFRQ(5),ILNFRQ(6),IPTFRQ(10)
      EQUIVALENCE(FILNAM(2),FLNAM(1))
      DATA FILNAM(1)/'DX1:'/
      COMMON /TSKARG/INFO(2),HOURS,PRODUC
C
C INPUT RECEIVER LOCATIONS
      WRITE(7,*)'HOW MANY RECEIVERS?'
      READ(7,*)NOBS
      WRITE(7,*)'INPUT X,Y,Z OF EACH RECEIVER'
      READ(7,*)((OBSPTS(J,N),J=1,3),N=1,NOBS)
C INPUT GROUND ATTENUATION
      WRITE(7,*)'EXCESS ATTENUATION, DB/DD'
      READ(7,*)EXATT
C SOURCE INPUT SECTION.  AN EQUIPMENT/TASK NAME IS REQUESTED FROM THE
C USER.  THIS NAME IS PASSED TO 'DECODE', WHICH RETURNS AN INDEX IGEOM
C SPECIFYING WHETHER IT IS A POINT (1), A LINE (2), OR AREA (3) SOURCE.
C A SMALL ARRAY, INFO, IS ALSO RETURNED, PROVIDING ADDITIONAL INFORMATION
C AS TO THE TYPE OF SOURCE.  THE NAME SUPPLIED BY THE USER IS RETURNED
C WITH A SEQUENTIAL INTEGER APPENDED TO IT.  THIS ID IS INCORPORATED
C INTO THE FINAL DATA FILE CREATED.
C
C INITIALIZE SOURCE TYPE COUNTS:
      NPTSRC=0
      NLNSRC=0
      NARSRC=0
C
C REQUEST SOURCE TYPE:
 7    WRITE(7,*)
      WRITE(7,*)'SOURCE TYPE?'
 5    READ(7,110)SRCNAM
      IF(SRCNAM(1).EQ.'    ')GO TO 10
      CALL DECODE(SRCNAM,IGEOM,INFO)
C TRAP FOR INVALID NAME, SIGNALLED BY IGEOM=0:
      IF(IGEOM.NE.0)GO TO 4
      WRITE(7,*)'INVALID SOURCE TYPE.  REENTER:'
      GO TO 5
 4    CONTINUE
C
C INPUT HOURS
      WRITE(7,*)'HOURS WORKED:'
      READ(7,*)HOURS
C TRANSFER TO APPROPRIATE GEOMETRY CASE SECTION:
      GO TO (1,2,3)IGEOM
C POINT SOURCE SECTION
 1    CONTINUE
C INCREMENT COUNTER AND SAVE NAME
      NPTSRC=NPTSRC+1
      DO 6 II=1,4
 6    PTNAM(II,NPTSRC)=SRCNAM(II)
C INPUT LOCATION
      WRITE(7,*)'ENTER X,Y,Z OF SOURCE LOCATION'
      READ(7,*)(XPTSRC(J,NPTSRC),J=1,3)
C GO TO SUBROUTINE 'PTTASK' FOR REST OF INFORMATION.  THIS SUBROUTINE
C CONTROLS ALL POINT SOURCE TASK MODELS.  IT INCLUDES RUN-TIME OPTION
```

```
C REQUESTS,OPTIONS SETTABLE THROUGH DATA STATEMENTS/CONTROL FILES, AND
C READ STATEMENTS CORRESPONDING TO DATA REQUIRED FOR THE SELECTED
C OPTION(S).  THE EFFECTIVE SOURCE HEIGHT IS ADDED TO THE INPUT Z.
      CALL PTTASK(ELPT(NPTSRC),XPTSRC(3,NPTSRC),IPTFRQ(NPTSRC))
      GO TO 7
C END OF POINT SOURCE SECTION
C
C LINE SOURCE SECTION
 2    CONTINUE
C INCREMENT COUNTER AND SAVE NAME
      NLNSRC=NLNSRC+1
      IF(NLNSRC.GT.6)WRITE(7,*)'TOO MANY LINE SOURCES -
     1 DIMENSIONS OVERRUN'
      DO 8 II=1,4
 8    ELNAM(II,NLNSRC)=SRCNAM(II)
C INPUT POINTS DEFINING LINE
      WRITE(7,*)'HOW MANY POINTS IN LINE?'
      READ(7,*)NLNPTS(NLNSRC)
C TEST FOR 'USE LAST' CASE
      IF(NLNPTS(NLNSRC).GT.0)GO TO 16
      NLNPTS(NLNSRC)=NLNPTS(NLNSRC-1)
      GO TO 22
 16   CONTINUE
      WRITE(7,*)'ENTER X,Y,Z OF POINTS'
      READ(7,*)((XLNSRC(J,I,NLNSRC),J=1,3),I=1,NLNPTS(NLNSRC))
C STORE IN NLNSRC+1 IN CASE NEXT TASK USES SAME GEOMETRY.  THIS WILL
C BE OVERWRITTEN IF NEW COORDINATES ARE INPUT.
 22   CONTINUE
      IF(NLNSRC.EQ.6)GO TO 17
      DO 18 I=1,NLNPTS(NLNSRC)
      DO 18 J=1,3
      XLNSRC(J,I,NLNSRC+1)=XLNSRC(J,I,NLNSRC)
 18   CONTINUE
 17   CONTINUE
C GO TO SUBROUTINE 'LNTASK' FOR REST OF INFORMATION
      CALL LNTASK(XLNSRC(1,1,NLNSRC),NLNPTS(NLNSRC),ELLN
     1(1,NLNSRC),EN(1,NLNSRC),ILNFRQ(NLNSRC))
      GO TO 7
C END OF LINE SOURCE SECTION
C
C AREA SOURCE SECTION
 3    CONTINUE
C INCREMENT COUNTER AND SAVE NAME
      NARSRC=NARSRC+1
      IF(NARSRC.GT.5)WRITE(7,*)'TOO MANY AREA SOURCES -
     1DIMENSIONS OVERRUN'
      DO 9 II=1,4
 9    ARNAM(II,NARSRC)=SRCNAM(II)
C INPUT GEOMETRY
      WRITE(7,*)'HOW MANY POINTS DEFINE CENTERLINE?'
      READ(7,*)NCLPTS(NARSRC)
C TEST FOR 'USE LAST' CASE
      IF(NCLPTS(NARSRC).GT.0)GO TO 19
      NCLPTS(NARSRC)=NCLPTS(NARSRC-1)
```

```
      GO TO 23
 19   CONTINUE
      WRITE(7,*)'ENTER X,Y,Z AND WIDTH OF POINTS'
      READ(7,*)((XCLPTS(J,I,NARSRC),J=1,3),WIDTH(I,NARSRC),
     1I=1,NCLPTS(NARSRC))
C STORE IN NARSRC+1 IN CASE NEXT TASK USES SAME GEOMETRY
 23   CONTINUE
      IF(NARSRC.EQ.5)GO TO 20
      DO 21 I=1,NCLPTS(NARSRC)
      WIDTH(I,NARSRC+1)=WIDTH(I,NARSRC)
      DO 21 J=1,3
      XCLPTS(J,I,NARSRC+1)=XCLPTS(J,I,NARSRC)
 21   CONTINUE
 20   CONTINUE
C GO TO SUBROUTINE 'ARTASK' FOR REST OF INFORMATION
      CALL ARTASK(XCLPTS(1,1,NARSRC),WIDTH(1,NARSRC),NCLPTS(
     1NARSRC),ELAR(1,NARSRC),IARFRQ(NARSRC))
      GO TO 7
C END OF AREA SOURCE CASE
C
C BARRIER DATA SECTION
 10   CONTINUE
      WRITE(7,*)'HOW MANY BARRIERS?'
      READ(7,*)NBAR
      IF(NBAR.EQ.0)GO TO 24
      DO 25 N=1,NBAR
      WRITE(7,*)'HOW MANY POINTS?'
      READ(7,*)NBPTS(N)
      WRITE(7,*)'ENTER X,Y,Z OF POINTS'
      READ(7,*)((BARPTS(J,I,N),J=1,3),I=1,NBPTS(N))
 25   CONTINUE
C
 24   CONTINUE
C CREATE A DATA FILE FOR HICNOM.  FIRST REQUEST A TITLE AND DATA FILE
C NAME.  THEN WRITE FILE USING SAME WRITE FORMATS,AS HICNOM'S READ
C FORMATS.
      WRITE(7,*)
      WRITE(7,*)'ENTER TITLE'
      READ(7,100)TITLE
      WRITE(7,*)'DATA FILE NAME - FILNAM.DAT'
      READ(7,110)FLNAM
      OPEN(UNIT=1,NAME=FILNAM,TYPE='NEW')
      REWIND 1
C NOW WRITE ONTO FILE, FOLLOWING HICNOM DATA FORMATS EXACTLY.  SOURCE
C NAMES HAVE BEEN ADDED TO RIGHT OF DATA FILES HERE; HICNOM FORMATS
C WILL BE ADJUSTED TO READ THESE AS WELL FOR FINAL PRINTOUT PURPOSES.
      WRITE(1,100)TITLE
      WRITE(1,112)NOBS,'RECE','IVER','S   '
      WRITE(1,102)((OBSPTS(J,N),J=1,3),N=1,NOBS)
      WRITE(1,112)NPTSRC,'POIN','T SO','URCE','S   '
      IF(NPTSRC.EQ.0)GO TO 11
      WRITE(1,111)((XPTSRC(J,N),J=1,3),ELPT(N),IPTFRQ(N),(PTNAM(II,N),
     1II=1,4),N=1,NPTSRC)
 11   CONTINUE
```

```
      WRITE(1,112)NLNSRC,'LINE',' SOU','RCES'
      IF(NLNSRC.EQ.0)GO TO 12
      DO 13 N=1,NLNSRC
      WRITE(1,101)NLNPTS(N),ILNFRQ(N),(ELNAM(II,N),II=1,4)
      WRITE(1,104)(((XLNSRC(J,I,N),J=1,3),ELLN(I,N),EN(I,N)),
     1I=1,NLNPTS(N))
13    CONTINUE
12    CONTINUE
      WRITE(1,112)NARSRC,'AREA',' SOU','RCES'
      IF(NARSRC.EQ.0)GO TO 14
      DO 15 N=1,NARSRC
      WRITE(1,101)NCLPTS(N),IARFRQ(N),(ARNAM(II,N),II=1,4)
      WRITE(1,103)(((XCLPTS(J,I,N),J=1,3),WIDTH(I,N),ELAR(I,N)),
     1I=1,NCLPTS(N))
15    CONTINUE
14    CONTINUE
      WRITE(1,112)NBAR,'BARR','IERS'
      IF(NBAR.EQ.0)GO TO 26
      DO 27 N=1,NBAR
      WRITE(1,112)NBPTS(N),'POIN','TS  '
      WRITE(1,102)((BARPTS(J,I,N),J=1,3),I=1,NBPTS(N))
27    CONTINUE
26    CONTINUE
      WRITE(1,105)EXATT
C FILE COMPLETE - CLOSE IT AND STOP
      CLOSE(UNIT=1,DISPOSE='SAVE')
C FORMATS
100   FORMAT(20A4)
101   FORMAT(I3,I7,10X,4A4)
102   FORMAT(3F10.2)
103   FORMAT(5F10.2)
104   FORMAT(4F10.2,F10.7)
105   FORMAT(F10.2,'DB/DD EXCESS ATTENUATION')
110   FORMAT(4A4)
111   FORMAT(4F10.2,I10,10X,4A4)
112   FORMAT(I3,10X,4A4)
      CALL EXIT
      END
```

```
      PROGRAM HICNOM
C
C THIS COMPUTATIONAL MODEL READS SOURCE LEVELS, NUMBERS, AND GEOMETRIES
C FROM A DATA FILE PREPARED BY HINPUT, THE INPUT AND SOURCE MODULE.
C ALL VARIABLE NAMES, DIMENSIONS, AND FORMAT LABELS ARE COMPATABLE WITH
C HINPUT. THIS WILL FACILITATE ANY FUTURE MERGING OF THE TWO PROGRAMS
C FOR USE ON A LARGER SYSTEM.
C
C THE NOISE CALCULATION CONSIDERS THREE TYPES OF SOURCE GEOMETRY:
C POINT, LINE, AND AREA.  POINT SOURCES ARE HANDLED BY SIMPLE GEOMETRIC
C SPREADING PLUS POWER LAW EXCESS ATTENUATION.  LINE SOURCES ARE
C HANDLED BY AN ANLYTIC ALGORITHM FOR ARBITRARY LENGTH LINE SOURCES WITH
C POWER LAW EXCESS ATTENUATION; THIS CALCULATION IS CONTAINED IN
C THE SUBROUTINE 'LINSRC'.  AREA SOURCES ARE TREATED BY DIVIDING THEM
C INTO A SERIES OF PARALLEL STRIPS, THEN TREATING EACH STRIP AS A LINE
C SOURCE, USING 'LINSRC'.  THE NUMBER OF STRIPS IS DECIDED BY THE PROGRAM,
C BASED ON THE PRECISION AS A FUNCTION OF RATIO OF AREA WIDTH TO
C RECEIVER DISTANCE.
C BARRIER SHIELDING IS HANDLED BY MAEKAWA'S FORMULATION, NUMERICALLY
C INTEGRATING THIS ALONG LINE SOURCES.  THE DIFFRACTION CURVES FOR
C VARIOUS SOURCES, OBTAINED BY INTEGRATING MAEKAWA'S OVER NOMINAL
C SPECTRA, ARE CONTAINED IN A SUBROUTINE AND ARE IDENTIFIED
C BY AN INDEX FOR EACH SOURCE
C
C INPUT VARIABLES:
C     BARPTS(3,5,3) = COORDINATES OF POINTS DEFINING BARRIERS
C     ELAR(10,5)= TOTAL EMISSION LEVEL OF SOURCE IN ONE AREA SEGMENT
C     ELLN(20,6) = EMISSION LEVEL OF LINE SOURCE, PER VEHICLE
C     ELPT(10) = EMISSION LEVEL OF POINT SOURCE
C     EN(20,6) = SOURCE DENSITY (NUMBER/UNIT LENGTH) ON LINE SEGMENT
C     EXATT = EXCESS ATTENUATION DUE TO GROUND AND AIR, DB/DOUBLING OF DISTANCE
C     FLNAM(3) = NAME OF DATA FILE PREPARED BY HINPUT
C     IARFRQ(5) = SPECTRUM IDENTIFIER FOR AREA SOURCE
C     ILNFRQ(6) = SPECTRUM IDENTIFIER FOR LINE SOURCE
C     IPTFRQ(10) = SPECTRUM IDENTIFIER FOR POINT SOURCE
C     NARSRC = NUMBER OF AREA SOURCES
C     NBAR = NUMBER OF BARRIERS,UP TO 3
C     NBPTS(3) = NUMBER OF POINTS DEFINING BARRIERS, UP TO 5
C     NCLPTS(5) = NUMBER OF POINTS DEFINING CENTERLINE OF AREA SOURCES
C     NLNPTS(6) = NUMBER OF POINTS DEFINING LINE SOURCES
C     NLNSRC = NUMBER OF LINE SOURCES
C     NOBS = NUMBER OF RECEIVER POINTS
C     NPTSRC = NUMBER OF POINT SOURCES
C     OBSPTS(3,10) = COORDINATES OF RECEIVER POINTS
C     TITLE(20) = 80 CHARACTER TITLE STRING TO HEAD OUTPUT
C     WIDTH(10,5) = WIDTH OF AREA SOURCE AT EACH CENTERLINE POINT
C     XCLPTS(3,10,5) = COORDINATES OF AREA SOURCE CENTERLINES
C     XLNSRC(3,20,6) = COORDINATES DEFINING LINE SOURCE
C     XPTSRC(3,10) = COORDINATES OF POINT SOURCES
C OUTPUT VARIABLES
C     ELEQ(10) = EQUIVALENT SOUND LEVEL AT RECEIVER POSITIONS
C     ELVEL = COMPONENT LEQ, CORRESPONDING TO EYECMP
C     EYECMP(180,10) = INTENSITY CONTRIBUTION OF EACH SOURCE AT EACH RECEIVER
C     INDEX(180) = CODE IDENTIFYING SOURCE/RECEIVER
```

```
C OTHER VARIABLES
C     ATTEN = BARRIER SHIELDING FACTOR
C     CLLNTH(10) = LENGTHS OF AREA CENTERLINE SEGMENTS
C     D = DISTANCE FROM RECEIVER TO LINE SOURCE
C     DO = EMISSION LEVEL REFERENCE DISTANCE
C     DO2 = DO**2
C     DX,DY = DIFFERENCES BETWEEN COORDINATES
C     EXPON = EITHER 1.0 OR GPLUS1 (SEE BELOW)
C     EYE(10) = INTENSITY OF TOTAL RECEIVED SOUND, CORRESPONDING TO ELEQ
C     EYEAR(10,5) = INTENSITY OF EMISSION FROM AN AREA SEGMENT
C     EYEBAR = COMPONENT SOURCE CONTRIBUTION WITH BARRIER SHIELDING
C     EYELN(20,6) = INTENSITY EMISSION PER VEHICLE ON LINE SOURCE
C     EYEPT(10) = INTENSITY EMISSION OF POINT SOURCE
C     EYESTR(10) = INTENSITY EMISSION DENSITY ON AREA STRIPS
C     EYETEM = INTENSITY CONTRIBUTION OF A STRIP
C     FILNAM(4) = DEVICE AND NAME OF INPUT FILE CREATED BY HINPUT
C     GNDA = ATTENUATION EXPONENT = EXATT/6.
C     GPLUS1 = GNDA+1.
C     ICROSS = TEST VARIABLE TO SIGNAL WHETHER A BARRIER BLOCKS A SOURCE
C     ISKIP = OPTION PARAMETER FOR SUBROUTINE 'GEOM'; SEE GEOM
C     NN = RUNNING INDEX OF SOURCE COMPONENTS
C     NSTR(20) = NUMBER OF STRIPS EACH AREA SEGMENT IS DIVIDED INTO
C     OBSROD(3) = RECEIVER COORDINATES TRANSFORMED RE: LINE OR STRIP SEGMENT
C     OBSROT(3,10,10) = RECEIVER COORDINATES TRANSFORMED RE: AREA CL SEGMENT
C     PHI(2) = ANGLES FROM RECEIVER TO ENDS OF LINE SEGMENT
C     RDLNTH = LENGTH OF LINE SEGMENT
C     RDLNTL = LENGTH OF STRIP SEGMENT, LEFT SIDE
C     RDLNTR = LENGHT OF STRIP SEGMENT, RIGHT SIDE
C     RLENT(3) = ARRAY DEFINING SEGMENT END IN SEGMENT ORIENTED COORDINATES
C     ROTCS(2) = COSINE AND SINE OF TRANSFORMATION ROTATION ANGLE
C     ROTNO(2) = ROTCS FOR NO ROTATION
C     R2 = SQUARE OF SOURCE-RECEIVER DISTANCE
C     STRIPL(3,2,5,10) = END POINT COORDINATES OF LEFT SIDE STRIPS
C     STRIPR(3,2,5,10) = END POINT COORDINATES OF RIGHT SIDE STRIPS
C     XCROSS(3) = COORDINATES OF BARRIER/LINE OF SIGHT CROSSING
C     ZERO(3) = COORDINATES OF SEGMENT END IN SEGMENT-ORIENTED COORDINATES
C DO LOOP INDICES
C     I = SEGMENT ALONG LINE OR AREA CENTER LINE
C     IBAR = SEGMENT ALONG A BARRIER
C     J = RECEIVER NUMBER
C     JBAR = BARRIER NUMBER
C     K = X,Y,Z INDEX
C     L = INDEX OF STRIP IN AREA SEGMENT
C     N = NUMBER OF SOURCE
C
      DIMENSION TITLE(20),FILNAM(4),OBSPTS(3,10),XPTSRC(3,10),ELPT(10)
      DIMENSION EYEPT(10),NLNPTS(6),XLNSRC(3,20,6),ELLN(20,6),FLNAM(3)
      DIMENSION  EYELN(20,6),EN(20,6),NCLPTS(5),XCLPTS(3,10,5)
      DIMENSION WIDTH(10,5),ELAR(10,5),EYEAR(10,5),INDEX(180),ROTNO(2)
      DIMENSION EYECMP(180,10),PHI(2),ROTCS(2),OBSROD(3),OBSROT(3,10,10)
      DIMENSION EYESTR(10),STRIPL(3,2,5,10),STRIPR(3,2,5,10),NSTR(10)
      DIMENSION CLLNTH(10),ZERO(3),RLENT(3),EYE(10),ELEQ(10),XCROSS(3)
      DIMENSION BARPTS(3,5,3),IPTFRQ(10),ILNFRQ(6),IARFRQ(5),NBPTS(3)
      EQUIVALENCE (FILNAM(2),FLNAM(1))
```

```
C
      COMMON /UNITS/DO,DO2
C
      DATA FILNAM(1)/'DX1:'/
      DATA DO,DO2/50.,2500./
      DATA ZERO,RLENT/6*0./
      DATA ROTNO/1.,0./
C
      WRITE(7,109)
      READ(7,110)FLNAM
      OPEN(UNIT=1,NAME=FILNAM,TYPE='OLD')
      REWIND 1
      READ(1,100)TITLE
C RECEIVER POSITIONS.  UP TO 10, LOCATED AT OBSPTS.
      READ(1,101) NOBS
      READ (1,102)((OBSPTS(K,J),K=1,3),J=1,NOBS)
C POINT SOURCES.  UP TO 10.
      READ(1,101)NPTSRC
      IF(NPTSRC.EQ.0)GO TO 2
      DO 1 N=1,NPTSRC
      READ(1,111)(XPTSRC(K,N),K=1,3),ELPT(N),IPTFRQ(N)
C CONVERT LEVEL TO INTENSITY
 1    EYEPT(N)=10.**(ELPT(N)/10.)
 2    CONTINUE
C LINE SOURCES.  UP TO 10 SOURCES, EACH DEFINED BY UP TO 20 POINTS.
C NUMBER OF LINE SOURCES:
      READ(1,101)NLNSRC
      IF(NLNSRC.EQ.0)GO TO 4
      DO 3 N=1,NLNSRC
      READ(1,101)NLNPTS(N),ILNFRQ(N)
      DO 3 I=1,NLNPTS(N)
      READ(1,103)(XLNSRC(K,I,N),K=1,3),ELLN(I,N),EN(I,N)
 3    EYELN(I,N)=10.**(ELLN(I,N)/10.)
 4    CONTINUE
C AREA SOURCES.  UP TO 5, EACH DEFINED BY UP TO 10 CENTERLINE POINTS
C AND WIDTHS.
C NUMBER OF AREA SOURCES:
      READ(1,101)NARSRC
      IF(NARSRC.EQ.0) GO TO 6
      DO 5 N=1,NARSRC
      READ(1,101)NCLPTS(N),IARFRQ(N)
      DO 5 I=1,NCLPTS(N)
      READ(1,103)(XCLPTS(K,I,N),K=1,3),WIDTH(I,N),ELAR(I,N)
 5    EYEAR(I,N)=10.**(ELAR(I,N)/10.)
 6    CONTINUE
C BARRIER INPUT SECTION.  UP TO 3, EACH DEFINED BY UP TO 5 POINTS
C NUMBER OF BARRIERS:
      READ(1,101)NBAR
      IF(NBAR.EQ.0)GO TO 21
      DO 22 N=1,NBAR
      READ(1,101)NBPTS(N)
      READ(1,102)((BARPTS(K,J,N),K=1,3),J=1,NBPTS(N))
 22   CONTINUE
 21   CONTINUE
```

```
C
C READ PROPAGATION EXPONENT.  SECTION WILL BE EXPANDED LATER TO
C ALLOW MORE THAN ONE VALUE.
      READ(1,102)EXATT
      GNDA=EXATT/6.
      GPLUS1=1.+GNDA
C
      CLOSE(UNIT=1,DISPOSE='SAVE')
C
C CALCULATION OF SOURCE CONTRIBUTIONS.  SUBSOURCE COMPONENTS ARE
C STORED SEQUENTIALLY IN EYECMP(NN,J).  KEY BACK TO ORIGINAL SOURCES
C IS THROUGH INDEX(NN) WHICH CONTAINS PACKED VALUES OF I (SEGMENT
C NUMBER) AND N (SOURCE NUMBER) IN THE FORMAT N + 100*I (+10000
C IF AREA SOURCE).
C
C INITIALIZE INDEX NN.  IT IS INCREMENTED BEFORE ADDING EACH SOURCE
      NN=0
C
C POINT SOURCE SECTION
C
      IF(NPTSRC.EQ.0) GO TO 15
      DO 7 N=1,NPTSRC
      NN=NN+1
      INDEX(NN)=N
      DO 7 J=1,NOBS
C COMPUTE UBSHIELDED NOISE
      DX=OBSPTS(1,J)-XPTSRC(1,N)
      DY=OBSPTS(2,J)-XPTSRC(2,N)
      R2=DX**2+DY**2
      EYECMP(NN,J)=EYEPT(N)*(DO2/R2)**GPLUS1
      IF(NBAR.EQ.0)GO TO 7
C BARRIER SECTION.  TEST FOR ANY BARRIER SHIELDING THE LINE OF SIGHT.
C ONLY ONE BARRIER - THE FIRST ENCOUNTERED - IS CONSIDERED.
      DO 19 JBAR=1,NBAR
      DO 19 IBAR=1,NBPTS(JBAR)-1
      CALL CROSS(OBSPTS(1,J),XPTSRC(1,N),BARPTS(1,IBAR,JBAR),BARPTS
     1(1,IBAR+1,JBAR),XCROSS,ICROSS)
      IF(ICROSS.EQ.1)GO TO 20
 19   CONTINUE
C NO BARRIER SHIELDS THIS SOURCE
      GO TO 7
 20   CONTINUE
C A SHIELDING BARRIER HAS BEEN FOUND - OBTAIN BARRIER SHIELDING 'ATTEN'
      CALL PTBAR(OBSPTS(1,J),XPTSRC(1,N),XCROSS,IPTFRQ(N),ATTEN)
C TAKE SMALLER RESULT
      EYECMP(NN,J)=AMIN1(EYEPT(N)*ATTEN*(DO2/R2),EYECMP(NN,J))
 7    CONTINUE
 15   CONTINUE
C
C LINE SOURCE SECTION
C
      IF(NLNSRC.EQ.0) GO TO 16
      DO 8 N=1,NLNSRC
      DO 8 I=1,NLNPTS(N)-1
```

```
      NN=NN+1
      INDEX(NN)=100*I+N
      ISKIP=0
      DO 8 J=1,NOBS
C OBTAIN GEOMETRIC PARAMETERS FOR SEGMENT I AND RECEIVER J
      CALL GEOM(XLNSRC(1,I,N),XLNSRC(1,I+1,N),OBSPTS(1,J),D,PHI,
     1ROTCS,OBSROD,RDLNTH,ISKIP)
C SET ISKIP TO USE SAME ROTANG AND RDLNTH UNTIL NEXT I
      ISKIP=2
C OBTAIN LINE SOURCE LEQ
      CALL LINSRC(D,EYELN(I,N),EN(I,N),PHI,GNDA,EYECMP(NN,J)
     1,OBSROD(1),RDLNTH)
C TEST FOR ANY BARRIERS
      IF(NBAR.EQ.0)GO TO 8
C GET BARRIER LEQ
      CALL LNWALL(OBSROD,XLNSRC(1,I,N),ILNFRQ(N),GNDA,EYELN(I,N),
     1EN(I,N),ROTCS,PHI,RDLNTH,BARPTS,NBAR,NBPTS,EYEBAR)
C TAKE THE SMALLER - BARRIERS ARE PRESUMED NOT TO AMPLIFY
      EYECMP(NN,J)=AMIN1(EYECMP(NN,J),EYEBAR)
 8    CONTINUE
16    CONTINUE
C
C AREA SOURCE SECTION
C
      IF(NARSRC.EQ.0)GO TO 17
      DO 9 N=1,NARSRC
      CALL AREA(NCLPTS(N),XCLPTS(1,1,N),WIDTH(1,N),OBSPTS,OBSROT,
     1NOBS,EYEAR,EYESTR,STRIPL,STRIPR,NSTR,CLLNTH)
C ROTANG IS ZERO RE: TRANSFORMED COORDINATES
      ROTCS(1)=1.
      ROTCS(2)=0.
C CALCULATE AND ADD STRIP CONTRIBUTIONS IN ONE SEGMENT AT A TIME
      DO 9 I=1,NCLPTS(N)-1
C INCREMENT NN
      NN=NN+1
      INDEX(NN)=10000 + 100*I + N
C OBTAIN CONTRIBUTIONS OF CENTERLINES
C SET UP (RDLNTH,0) END OF CENTERLINE
      RDLNTH=CLLNTH(I)
      RLENT(1)=RDLNTH
C LOOP THROUGH RECEIVERS
      DO 10 J=1,NOBS
C CL END POINTS ARE AT (0,0) AND (RDLNTH,0); RDLNTH FIXED THROUGH J LOOP
C ROTNO IS USED IN THIS CALL TO GEOM BECAUSE OBSROT ARE ALREADY ROTATED;
C GEOM NEED ONLY SHIFT THESE (A ZERO SHIFT HERE) TO GET OBSROD
      CALL GEOM(ZERO,RLENT,OBSROT(1,J,I),D,PHI,ROTNO,OBSROD,RDLNTH,2)
C OBTAIN LINE SOURCE LEQ
      CALL LINSRC(D,EYESTR(I),1.,PHI,GNDA,EYECMP(NN,J),OBSROD(1),RDLNTH)
C TEST FOR BARRIERS
      IF(NBAR.EQ.0)GO TO 10
C OBTAIN BARRIER EYEEQ, AND TAKE SMALLER VALUE
      CALL LNWALL(OBSROD,XCLPTS(1,I,N),IARFRQ(N),GNDA,EYESTR(I),1.,
     1ROTCS,PHI,RDLNTH,BARPTS,NBAR,NBPTS,EYEBAR)
      EYECMP(NN,J)=AMIN1(EYECMP(NN,J),EYEBAR)
```

```
 10    CONTINUE
C
C OBTAIN CONTRIBUTIONS OF OTHER STRIPS
C NOTE: LOOP NESTING IN THIS SECTION (USING TWO J LOOPS) IS DESIGNED TO
C PERMIT MAXIMUM USE OF ISKIP=2 ENTRY TO GEOM, AND TO FACILITATE
C SKIPPING L LOOP IF NSTR(I)=0
C
      IF(NSTR(I).EQ.0)GO TO 9
C
C LOOP THROUGH STRIPS
      DO 11 L=1,NSTR(I)
C SET ISKIP TO COMPUTE NEW RDLNTHS FIRST TIME THROUGH J LOOP
      ISKIP=1
C LOOP THROUGH RECEIVERS
      DO 11 J=1,NOBS
C LEFT STRIPS
C USE ROTNO HERE BECAUSE STRIPS AND OBSROT ARE ROTATED ALREADY; ONLY
C A SHIFT RE:STRIPL IS NEEDED
      CALL GEOM(STRIPL(1,1,L,I),STRIPL(1,2,L,I),OBSROT(1,J,I)
     1,D,PHI,ROTNO,OBSROD,RDLNTL,ISKIP)
      CALL LINSRC(D,EYESTR(I),1.,PHI,GNDA,EYETEM,OBSROD(1),RDLNTL)
C TEST FOR BARRIERS
      IF(NBAR.EQ.0)GO TO 23
      CALL LNWALL(OBSROD,STRIPL(1,1,L,I),IARFRQ(N),GNDA,EYESTR(I),1.,
     1ROTCS,PHI,RDLNTL,BARPTS,NBPTS,EYEBAR)
      EYETEM=AMIN1(EYETEM,EYEBAR)
 23   CONTINUE
C ADD TO TOTAL
      EYECMP(NN,J)=EYECMP(NN,J)+EYETEM
C RIGHT STRIP
      CALL GEOM(STRIPR(1,1,L,I),STRIPR(1,2,L,I),OBSROT(1,J,I),
     1D,PHI,ROTNO,OBSROD,RDLNTR,ISKIP)
      CALL LINSRC(D,EYESTR(I),1.,PHI,GNDA,EYETEM,OBSROD(1),RDLNTR)
C TEST FOR BARRIERS
      IF(NBAR.EQ.0)GO TO 24
C OBTAIN BARRIER EYEEQ
      CALL LNWALL(OBSROD,STRIPR(1,1,L,I),IARFRQ(N),GNDA,EYESTR(I),1.,
     1ROTCS,PHI,RDLNTR,BARPTS,NBPTS,EYEBAR)
      EYETEM=AMIN1(EYETEM,EYEBAR)
 24   CONTINUE
C ADD TO TOTAL
      EYECMP(NN,J)=EYECMP(NN,J)+EYETEM
C SET ISKIP TO USE SAME RDLNTHS IN REST OF J LOOP
      ISKIP=2
 11   CONTINUE
 9    CONTINUE
 17   CONTINUE
C
C ALL SOURCE INTENSITIES AT ALL RECEIVER LOCATIONS ARE NOW COMPUTED
C SUMMATION/OUTPUT SECTION BEGINS HERE
C
      DO 12 J=1,NOBS
      EYE(J)=0.
      DO 13 N=1,NN
```

```
 13   EYE(J)=EYE(J)+EYECMP(N,J)
      IF(EYE(J).LE.0.)EYE(J)=10.
      ELEQ(J)=10.*ALOG10(EYE(J))
 12   CONTINUE
C OUTPUT SECTION
      WRITE(7,104)TITLE
      WRITE(7,105)(J,ELEQ(J),J=1,NOBS)
      DO 14 J=1,NOBS
      WRITE(7,106)J
      WRITE(7,107)
      DO 14 N=1,NN
      IF(EYECMP(N,J).LE.0.)EYECMP(N,J)=10.
      ELVEL=10.*ALOG10(EYECMP(N,J))
      WRITE(7,108)INDEX(N),EYECMP(N,J),ELVEL
 14   CONTINUE
      STOP
 100  FORMAT(20A4)
 101  FORMAT(I3,I7)
 102  FORMAT(3F10.2)
 103  FORMAT(5F10.2)
 104  FORMAT(20A4//,' RECEIVER NUMBER',10X,'LEQ'//)
 105  FORMAT(I12,13X,F6.1)
 106  FORMAT(//' COMPONENT CONTRIBUTIONS FOR RECEIVER NUMBER:',I3)
 107  FORMAT(/' INDEX',3X,'INTENSITY',4X,'LEVEL'/)
 108  FORMAT(I6,E15.6,F6.1)
 109  FORMAT(' FILE NAME - FILNAM.DAT')
 110  FORMAT(4A4)
 111  FORMAT(4F10.2,I10)
      END
```

```
      SUBROUTINE DECODE(SRCNAM,IGEOM,INFO)
C THIS SUBROUTINE ACCEPTS 'SRCNAM', THE NAME OF THE EQUIPMENT OR OPER-
C ATION.  THE NAME IS COMPARED TO A LIST OF AVAILABLE NAMES, 'ALNAMS'.
C THE LIST ALLOWS FOR UP TO 15 POINT SOURCE TYPES, 10 LINE TYPES, AND
C 5 AREA TYPES.  THE CORRESPONDING ID NUMBER IS OUTPUT AS 'INFO(1)'.  'IGEOM'
C IS SET TO 1, 2 OR 3 TO DENOTE WHETHER THIS IS A POINT, LINE OR AREA
C SOURCE.  A DEFAULT VALUE OF IGEOM FOR EACH SOURCE TYPE MAY BE OVERRIDDEN
C FOR A DESIGNATED LIST OF TYPES WHICH CAN HAVE ALTERNATES.  ONE OF THE
C POINT TYPES ALLOWED, ITYPE=15, IS 'USER DEFINED'.  ALTERNATE GEOMETRY
C MAY BE SPECIFIED, AND THE MODEL TYPE MUST ALWAYS BE ZERO OR LESS.  'INFO(2)'
C HAS A MODEL TYPE (INCLUDING TO-BE-DEFINED FLAG) STORED IN IT.
C THE INPUT 'SRCNAM' HAS AN INTEGER APPENDED TO IT TO DISTINGUISH IT
C FROM OTHERS OF THE SAME TYPE; THIS APPENDED NAME IS USED FOR ALL
C FUTURE IDENTIFICATION OF THE SOURCE IN OUTPUT COMMUNICATIONS.
C
C INPUT AND OUTPUT VARIABLES:
C    SRCNAM(4) = SOURCE TYPE NAME
C    IGEOM = 1,2,3 FOR POINT, LINE, AREA SOURCE
C    INFO(2) = MODEL AND ITYPE
C OTHER VARIABLES:
C    ALNAMS(3,30) = LIST OF ALLOWABLE SOURCE TYPE NAMES
C    COPY = WORD FOR TRANSFERRING 'NTH' TO 'SRCNAM'
C    I = DO LOOP INDEX
C    ICOPY(4) = BYTE ARRAY, EQUIVALENCED TO 'COPY', FOR POSITIONING 'NTH'
C    IPOS(30) = POSITION IN 'COPY'/'ICOPY' WHERE 'NTH' IS TO BE PUT
C    ITYPE = TASK TYPE
C    J = DO LOOP INDEX
C    MODEL = MODEL NUMBER OF SOURCE
C    NGEOM(30) = ARRAY CARRYING LIST OF 'ITYPE'S WITH ALTERNATE GEOMETRIES
C    NPOS(30) = POSITION IN SRCNAM INTO WHICH 'COPY' IS PLACED
C    NTH(10) = LITERAL VALUES OF INTEGERS
C    NUMBER(30) = COUNT OF HOW MANY SOURCES OF EACH 'ITYPE'
C    NMODS(30) = NUMBER OF MODELS OF EACH TYPE IN DATA BASE
C
      DIMENSION SRCNAM(4),ALNAMS(3,30),NUMBER(30),NTH(10),ICOPY(4)
      DIMENSION IPOS(30),NPOS(30),NGEOM(30),INFO(2)
      COMMON /NMODLS/NMODS(30)
      LOGICAL*1 NTH,ICOPY
      EQUIVALENCE (ICOPY(1),COPY)
      DATA NTH/'1','2','3','4','5','6','7','8','9','0'/
      DATA NUMBER/30*0/
      DATA NGEOM/1,2,8,15,16,17,27,23*0/
      DATA IPOS/2,1,1,2,3,1,1,2,3,6*1,1,1,1,2,6*1,1,1,3*1/
      DATA NPOS/3,3,4,4,2,3,3,3,3,6*4,4,3,3,3,6*4,4,3,3*4/
      DATA ALNAMS          /'BACK','HOE ','    ','LOAD','ER  ','    ',
     1'COMP','RESS','OR  ','PILE',' DRI','VER ','PUMP','    ','    ',
     2'CRAN','E   ','    ','SHOV','EL  ','    ','BREA','KER ','    ',
     3'CONC','RETE','    ','GENE','RATO','R   ','MISC','ELLA','N   ',
     4   9*'XXXX',          'USER',' DEF','INED',
     5'BULL','DOZE','R   ','GRAD','ER  ','    ','TRUC','KS  ','    ',
     6'SCRA','PER ','    ',
     7 18*'XXXX',
     8'COMP','ACTE','R   ','PAVI','NG  ','    ',
     9 9*'XXXX'/
```

```
C SEARCH ALNAMS FOR NAME
      DO 1 ITYPE=1,30
      DO 2 J=1,3
      IF(SRCNAM(J).NE.ALNAMS(J,ITYPE)) GO TO 1
 2    CONTINUE
      GO TO 3
 1    CONTINUE
      IGEOM=0
      RETURN
C INCREMENT NUMBER COUNT FOR SOURCE
 3    NUMBER(ITYPE)=NUMBER(ITYPE)+1
C SET IGEOM TO NOMINAL VALUE, BASED ON ITYPE
      IGEOM=1
      IF(ITYPE.GT.15)IGEOM=2
      IF(ITYPE.GT.25.)IGEOM=3
C APPEND NAME
      COPY='    '
      ICOPY(IPOS(ITYPE))=NTH(NUMBER(ITYPE))
      SRCNAM(NPOS(ITYPE))=COPY
C DETAILED SOURCE TYPE IS IDENTIFIED HERE.  'MODEL' IS INPUT.  IF THIS
C IS AN INTEGER GREATER THAN ZERO, THIS CORRESPONDS TO A SPECIFIC
C CATALOGED MODEL.  IF ZERO IS INPUT, THEN A NEW MODEL WILL BE CREATED
C IN THE APPROPRIATE EQUIPMENT SUBROUTINE, REQUESTING LEVEL AND
C CAPACITY DATA FROM THE USER.  ENTERING -1,-2, ETC., SIGNIFIES 'SAME
C AS LAST NEW MODEL OF THIS TYPE', 'SAME AS SECOND FROM LAST NEW MODEL
C OF THIS TYPE', ETC.  NOTE THAT ITYPE=15 IS ALWAYS ACCOMPANIED BY A ZERO
C MODEL NUMBER THE FIRST TIME IT IS SPECIFIED, AND ZERO OR NEGATIVE THEREAFTER
      WRITE(7,*)'MODEL NUMBER?'
      READ(7,*)MODEL
C SET UP TRUE MODEL NUMBER IF 'SAME AS LAST' CASE:
      IF(MODEL.LT.0)MODEL=NMODS(ITYPE)+1+MODEL
C SAVE ITYPE AND MODEL
      INFO(1)=MODEL
      INFO(2)=ITYPE
C TEST IF THERE IS A CHOICE OF GEOMETRIES
      DO 4 I=1,30
      IF(ITYPE.EQ.NGEOM(I))GO TO 5
 4    CONTINUE
      RETURN
 5    WRITE(7,*)'ENTER 1, 2, OR 3 FOR WORKING OVER POINT, LINE OR AREA:'
      READ(7,*)IGEOM
      RETURN
      END
```

```
      SUBROUTINE PTTASK(ELPT,Z,IFRQ)
C THIS SUBROUTINE ACCEPTS THE TASK DESCRIPTION 'INFO' (TYPE AND
C MODEL OF EQUIPMENT) AND RETURNS THE LEQ SOURCE LEVEL AT THE REFE-
C RENCE DISTANCE.  THE TIME WORKED EACH DAY - 'HOURS' - IS
C COMPARED TO THE WORKDAY TO ADJUST LEQ TO THE FULL WORK DAY.
C THE DAILY PRODUCTION RATE 'PRODUC' IS ALSO COMPUTED FOR USE IN
C BALANCING EQUIPMENT USAGE WITHIN A TASK.  EMISSION LEVELS,
C DUTY CYCLES, AND CAPACITIES ARE LOOKED UP FROM
C DATA FILES FOR SPECIFIC PIECES OF EQUIPMENT; THE USER MAY DEFINE NEW
C TYPES.  THE DATA FILES ARE MAINTAINED IN A SEPARATE BLOCK DATA ROUTINE.
C
C INPUT VARIABLE:
C    Z = INPUT SOURCE HEIGHT. RAISED BY ACOUSTIC HEIGHT FOR RETURN
C OUTPUT VARIABLE:
C    ELPT = NOISE LEVEL OF TASK
C OTHER VARIABLES:
C    DAYHRS = WORK DAY LENGTH    /WKDAY/
C    EQUIP(5,10,30) = EQUIPMENT DATA BASE    /EQUIPT/
C       EQUIP(1,M,I) = LMAX
C       EQUIP(2,M,I) = LMAX-LEQ(CYCLE)
C       EQUIP(3,M,I) = CAPACITY
C       EQUIP(4,M,I) = CYCLE TIME
C       EQUIP(5,M,I) = EFFECTIVE ACOUSTIC SOURCE HEIGHT
C    HOURS = HOURS WORKED IN TASK    /TSKARG/
C    IFRQ = SPECTRUM IDENTIFIER OF SOURCE
C    IFREQ(10,30) = EQUIPMENT SPECTRUM DATA BASE    /EQUIPT/
C    INFO(2) = MODEL AND TYPE
C    IPROD(30) = LIST OF TYPES WHICH HAVE PRODUCTION RATES    /TYPES/
C    J = DO LOOP INDEX
C    K = DO LOOP INDEX
C    M = MODEL NUMBER
C    NMODS(30) = NUMBER OF MODELS OF EACH TYPE    /NMODLS/
C    PROD = PRODUCTION RATE
C    PRODUC = DAILY PRODUCTION, SAVED FOR NEXT TASK    /TSKARG/
C
      COMMON /WKDAY/DAYHRS
      COMMON /TYPES/IPROD(30),IHAUL(10),IVEH(5,5)
      COMMON /NMODLS/NMODS(30),NVTYP
      COMMON /EQUIPT/EQUIP(5,10,30),IFREQ(10,30)
      COMMON /TSKARG/INFO(2),HOURS,PRODUC
C
      M=INFO(1)
      I=INFO(2)
      IF(M.GT.0)GO TO 1
C SECTION TO ADD NEW MODEL DATA
C INCREMENT NMODS
      NMODS(I)=NMODS(I)+1
      M=NMODS(I)
      WRITE(7,*)'ENTER LMAX,DELTA,CAPACITY,CYCLE TIME, ACOUSTIC
     1 HEIGHT, AND FREQUENCY:'
      READ(7,*)(EQUIP(K,M,I),K=1,5),IFREQ(M,I)
 1    CONTINUE
C CALCULATE LEQ(CYCLE)
      ELPT=EQUIP(1,M,I)-EQUIP(2,M,I)
```

```
C TEST TO SEE IF A PRODUCTION RATE IS ASSOCIATED WITH THIS EQUIPMENT TYPE
      DO 2 J=1,30
      IF(IPROD(J).EQ.1)GO TO 3
 2    CONTINUE
      GO TO 4
 3    CONTINUE
C CALCULATE DAILY PRODUCTION
      PROD=EQUIP(3,M,I)*DAYHRS/EQUIP(4,M,I)
C COMPUTE EFFECTIVE 'HOURS' IF BALANCED TO LAST:
      IF(HOURS.LT.0.)HOURS=DAYHRS*PRODUC/PROD
C COMPUTE DAILY PRODUCTION FROM HOURS AND DAILY RATE - NOTE REDUNDANCY
C OF THIS CALCULATION FOR 'BALANCE TO LAST' CASE
      PRODUC=PROD*HOURS/DAYHRS
 4    CONTINUE
C ADJUST LEVEL BY USE FACTOR
      ELPT=ELPT+10.*ALOG10(HOURS/DAYHRS)
C ADJUST Z TO ACOUSTIC HEIGHT
      Z=Z+EQUIP(5,M,I)
C GET FREQUENCY IDENTIFIER
      IFRQ=IFREQ(M,I)
      RETURN
      END
```

```
      SUBROUTINE LNTASK(XLNSRC,NLNPTS,ELLN,EN,IFRQ)
C     THIS SUBROUTINE ACCEPTS EQUIPMENT DESCRIPTION 'INFO' AND
C LINE SOURCE GEOMETRY 'XLNSRC'.  IT RETURNS 'ELLN' AND 'EN', THE CYCLE
C AVERAGED EMISSION LEVEL AND NUMBER OF VEHICLES PER UNIT LENGTH.  FOR
C MOST PIECES OF EQUIPMENT THE CALCULATION IS ESSENTIALLY THE SAME AS
C IN 'PTTASK', EXCEPT THAT THE SOURCE IS DISTRIBUTED OVER A LINE.
C ONE PIECE OF EQUIPMENT OPERATES ON THE LINE.  FOR HAULING OPERATIONS
C (HAUL ROADS, SCRAPERS, ETC.) THE CALCULATION IS SOMEWHAT DIFFERENT.
C THE TASK IS SET UP WITH A HIGHWAY-LIKE FORMULATION, AND MULTIPLE
C VEHICLES ARE EITHER SPECIFIED BY THE USER OR INFERRED FROM PROD-
C UCTION RATES.  TRAVEL SPEED AND ACCELERATION/DECELERATION AT LOAD/DUMP
C POINTS ARE ACCOUNTED FOR.  RETURN LOOP GEOMETRY AND SPEEDS ON ACCEL-
C ERATION/DECELERATION SEGMENTS ARE GENERATED AS AN OPTION, USING THE
C 'HAULRD' SET OF ROUTINES.
C
C INPUT VARIABLES:
C   XLNSRC(3,20) = COORDINATES OF LINE
C   NLNPTS = NUMBER OF POINTS DEFINING LINE
C OUTPUT VARIABLES:
C   ELLN(19) = EQUIPMENT EMISSION LEVEL ON EACH LINE SEGMENT
C   EN(19) = EFFECTIVE NUMBER OF VEHICLES PER UNIT LENGTH ON EACH SEGMENT
C  (THE CONTENTS OF COMMON BLOCK /TSKARG/ ARE ALSO TREATED AS I/O )
C OTHER VARIABLES:
C   CYCLE = EQUIPMENT CYCLE TIME
C   DAYHRS = LENGTH OF WORK DAY   /WKDAY/
C   DX,DY = X AND Y EXTENT OF LINE SEGMENT
C   DS = LENGTH OF LINE SEGMENT
C   EL = NOISE LEVEL
C   EQUIP(5,10,30) = DATA BASE FOR NON-HAUL EQUIPMENT   /EQUIPT/
C       EQUIP(1,M,I) = LMAX
C       EQUIP(2,M,I) = LMAX-LEQ(CYCLE)
C       EQUIP(3,M,I) = CAPACITY PER CYCLE
C       EQUIP(4,M,I) = CYCLE DURATION
C       EQUIP(5,M,I) = EFFECTIVE ACOUSTIC HEIGHT
C   HAULEQ(6,10) = DATA BASE FOR HAUL EQUIPMENT   /VEHLEV/
C       HAULEQ(1,IVEH) = EMISSION LEVEL
C       HAULEQ(2,IVEH) = REFERENCE SPEED
C       HAULEQ(3,IVEH) = SPEED DEPENDENCE SLOPE
C       HAULEQ(4,IVEH) = CRITICAL SPEED
C       HAULEQ(5,IVEH) = CAPACITY
C       HAULEQ(6,IVEH) = EFFECTIVE ACOUSTIC HEIGHT
C   HOURS = TIME WORKED FOR TASK   /TSKARG/
C   I = EQUIPMENT OR TASK TYPE NUMBER
C   IFRQ = SPECTRUM IDENTIFIER OF SOURCE
C   IFREQ(10,30) = EQUIPMENT SPECTRUM DATA BASE   /EQUIPT/
C   II = I-15
C   IHAUL(10) = LIST OF HAUL TASK TYPE NUMBERS   /TYPES/
C   ILOOP = TYPE OF RETURN LOOP FOR HAUL TASKS
C   INFO(2) = MODEL AND TYPE   /TASKARG/
C   IPROD(30) = LIST OF EQUIPMENT/TASK TYPES WITH PRODUCTION RATES  /TYPES/
C   IVEH(5,5) = TABLE OF HAUL TASK TYPE/MODEL NUMBERS, IVEHM,II)   /TYPES/
C   J = DO LOOP INDEX
C   K = DO LOOP INDEX
C   M = MODEL NUMBER
```

```
C    N = DO LOOP INDEX
C    NMODS(30) = NUMBER OF MODELS OF EACH TYPE   /NMODLS/
C    NVTYP = NUMBER OF HAUL TASK TYPES   /NMODLS/
C    PROD = PRODUCTION RATE, FULL WORK DAY
C    PRODUC = DAILY PRODUCTION, SAVED FOR NEXT TASK   /TSKARG/
C    RAD = RADIUS OF RETURN LOOP
C    SPEED(19) = SPEED OF VEHICLES ON EACH SEGMENT, INPUT UNITS
C    SUM = TOTAL LENGTH OF LINE
C    SUMINV = 1./SUM
C    VELCON = VELOCITY CONVERSION BETWEEN INPUT AND CONSISTENT UNITS
C    VELOC = EQUIPMENT SPEED, CONSISTENT UNITS
C    VOL = ONE-WAY VOLUME, VEHICLES PER HOUR, ON HAUL ROAD
C
      DIMENSION XLNSRC(3,20),ELLN(19),EN(19),SPEED(19)
      COMMON /EQUIPT/EQUIP(5,10,30),IFREQ(10,30)
      COMMON /NMODLS/NMODS(30),NVTYP
      COMMON /TSKARG/INFO(2),HOURS,PRODUC
      COMMON /TYPES/IPROD(30),IHAUL(10),IVEH(5,5)
      COMMON /WKDAY/DAYHRS
      COMMON /UNITS/DO,DO2,GRAV,VELCON
      COMMON /VEHLEV/HAULEQ(6,10)
      M=INFO(1)
      I=INFO(2)
      II=I-15
C TEST FOR HAUL OR NOT.  HAUL SECTION STARTS AT LABEL 2, AND IS ESSEN-
C TIALLY A SEPARATE BLOCK.
      DO 1 K=1,10
      IF(I.EQ.IHAUL(K))GO TO 2
 1    CONTINUE
C
C NON-HAUL ROAD SECTION - LOGIC PARALLELS 'PTTASK'
      IF(M.GT.0)GO TO 3
C SECTION TO INPUT NEW NON-HAUL DATA
      NMODS(I)=NMODS(I)+1
      M=NMODS(I)
      WRITE(7,*)'ENTER LMAX,DELTA,CAPACITY,SPEED, ACOUSTIC HEIGHT,
     1 AND FREQUENCY:'
C EQUIP(4,M,I) IS SPEED, IN INPUT UNITS.  CYCLE TIME IS COMPUTED FROM
C THIS AND TWICE THE TOTAL PATH LENGTH, PRESUMING A CYCLE TO BE A
C ROUND TRIP
      READ(7,*)(EQUIP(K,M,I),K=1,5),IFREQ(M,I)
 3    CONTINUE
C CALCULATE LENGTH OF PATH - EN IS THE RECIPROCAL OF THIS
      SUM=0.
      DO 4 N=1,NLNPTS-1
      DX=XLNSRC(1,N+1)-XLNSRC(1,N)
      DY=XLNSRC(2,N+1)-XLNSRC(2,N)
      DS=SQRT(DX**2+DY**2)
      SUM=SUM+DS
 4    CONTINUE
      SUMINV=1./SUM
      DO 5 N=1,NLNPTS-1
 5    EN(N)=SUMINV
      EN(NLNPTS)=0.
```

```
C TEST TO SEE IF PRODUCTION RATE IS ASSOCIATED WITH THIS EQUIPMENT TYPE
      DO 6 J=1,30
      IF(I.EQ.IPROD(J))GO TO 7
 6    CONTINUE
      GO TO 8
 7    CONTINUE
C CALCULATE DAILY PRODUCTION
      VELOC=EQUIP(4,M,I)*VELCON
      CYCLE=SUM/(VELOC*1800.)
      PROD=EQUIP(3,M,I)*DAYHRS/CYCLE
C EFFECTIVE HOURS IF BALANCED TO LAST:
      IF(HOURS.LT.0.)HOURS=DAYHRS*PRODUC/PROD
C COMPUTE PRODUCTION:
      PRODUC=PROD*HOURS/DAYHRS
 8    CONTINUE
C COMPUTE LEVEL AND ADD ACOUSTIC SOURCE HEIGHT:
      Z=EQUIP(5,M,I)
      EL=EQUIP(1,M,I)-EQUIP(2,M,I)+10.*ALOG10(HOURS/DAYHRS)
      DO 9 N=1,NLNPTS-1
      XLNSRC(3,N)=XLNSRC(3,N)+Z
 9    ELLN(N)=EL
      XLNSRC(3,NLNPTS)=XLNSRC(3,NLNPTS)+Z
      ELLN(NLNPTS)=0.
C GET SPECTRUM IDENTIFIER
      IFRQ=IFREQ(M,I)
C END OF NON-HAUL SECTION
      RETURN
C
C HAUL OPERATION SECTION.  TASK IS VEHICLES MOVING ON A ROAD.  PROD-
C UCTION IS ACCOUNTED FOR BY VEHICLE VOLUME, NOT WORK PERIOD.  VARYING
C SPEED IS PERMITTED.  RETURN LOOP GEOMETRY AND ACCELERATION/DECELERATION
C DETAILS MAY BE COMPUTED BY THE PROGRAM, AS OPTIONS UNDER SUBROUTINE
C 'HAULRD'.  VEHICLE EMISSION LEVELS ARE PROVIDED BY FUNCTION 'ELVEH',
C RATHER THAN DATA ARRAY 'EQUIP'
C
 2    CONTINUE
      IF(M.GT.0)GO TO 10
C SECTION TO INPUT NEW HAUL DATA
      NVTYP=NVTYP+1
      NMODS(I)=NMODS(I)+1
      M=NMODS(I)
      IVEH(M,II)=NVTYP
      WRITE(7,*)'ENTER LMAX,REFSPD,SLOPE,VCRIT,CAPACITY, ACOUSTIC
     1 HEIGHT, AND FREQUENCY:'
      READ(7,*)(HAULEQ(K,NVTYP),K=1,6),IFREQ(M,I)
 10   CONTINUE
C ADD ACOUSTIC SOURCE HEIGHT
      Z=HAULEQ(6,NVTYP)
C GET SPECTRUM IDENTIFIER
      IFRQ=IFREQ(M,I)
      DO 13 N=1,NLNPTS
 13   XLNSRC(3,N)=XLNSRC(3,N)+Z
      WRITE(7,*)'ENTER SPEED ON ALL SEGMENTS'
      READ(7,*)(SPEED(N),N=1,NLNPTS-1)
```

```
C NEGATIVE HOURS INDICATES BALANCE TO LAST - NO NEED TO READ VOL
      IF(HOURS.LT.0.)GO TO 11
      WRITE(7,*)'VEHICLES PER HOUR:'
      READ(7,*)VOL
 11   CONTINUE
      WRITE(7,*)'TYPE AND RADIUS OF RETURN LOOP'
      READ(7,*)ILOOP,RAD
C
C COMPUTE PRODUCTION
      IF(HOURS.LT.0.)VOL=PRODUC/(HAULEQ(5,IVEH(M,II))*DAYHRS)
      PRODUC=VOL*HAULEQ(5,IVEH(M,II))*DAYHRS
C SCALE TO HOURS WORKED
      IF(HOURS.GT.0.)PRODUC=PRODUC*HOURS/DAYHRS
      Q=VOL/3600.
C CALL HAULRD OR PASSBY, AS INDICATED BY 'ILOOP'
      IF(ILOOP.LE.0)GO TO 12
      CALL HAULRD(XLNSRC,NLNPTS,RAD,ILOOP,SPEED,Q,IVEH(M,II),EN,ELLN)
      RETURN
 12   CONTINUE
      WRITE(7,*)'STOPPING AND DECELERATION POINTS'
      READ(7,*)ISTOP,IDEC
      CALL PASSBY(XLNSRC,NLNPTS,ISTOP,IDEC,SPEED,Q,IVEH(M,II),
     1EN,ELLN)
      RETURN
      END
```

```
      SUBROUTINE HAULRD(RDPTS,IRDPTS,RAD,ILOOP,SPEED,Q,
     1IVEH,EN,ELL)
C PROGRAM TO CONVERT INPUT ROAD POINTS, SPEED, AND VOLUME TO ROAD
C WITH RETURN LOOP AND ACOUSTIC SOURCE QUANTITIESTHE INPUT RDPTS
C AND IRDPTS ARE MODIFIED. OTHER QUANTITIES ARE NEWLY CREATED.
C
C INPUT VARIABLES:
C    RDPTS(3,20) = COORDINATES OF POINTS DEFINING ROAD
C    IRDPTS = TOTAL NUMBER OF POINTS IN ROAD.  AFTER 'LOOP' AND
C              BEFORE RETURN, DENOTES INDEX OF LOOP BRANCH POINT.
C    RAD = RADIUS OF TURNAROUND LOOP, IF ONE IS SPECIFIED
C    ILOOP = TYPE OF TURNAROUND LOOP; SEE 'LOOP' FOR OPTIONS
C    SPEED(19) = SPEED OF VEHICLES, IN INPUT UNITS
C    Q = ONE-WAY VOLUME OF VEHICLES PER SECOND ON ROAD
C    IVEH = INDEX IDENTIFYING TYPE OF VEHICLE
C OUTPUT VARIABLES:
C    EN(19) = NUMBER OF VEHICLES PER UNIT DISTANCE ON EACH SEGMENT
C    ELL(19) = EFFECTIVE LEQ(E) OF VEHICLES ON EACH SEGMENT
C    RDPTS AND IRDPTS ARE MODIFIED TO INCLUDE LOOP POINTS
C OTHER VARIABLES:
C    I= ROAD SEGMENT INDEX IN DO LOOPS
C    IDEC = INDEX OF DECELERATION POINT, CALCULATED IN 'DECACC'
C    NLOAD = INDEX OF LOADING POINT, CALCULATED IN 'LOOP'
C    NRDPTS = NEW TOTAL NUMBER OF POINTS, AS EXTENDED BY 'LOOP'
C    V(19) = SPEED OF VEHICLES ON EACH SEGMENT, IN CONSISTENT UNITS
C
      DIMENSION RDPTS(20),SPEED(19),EN(19),ELL(19),V(19)
      CALL LOOP (RDPTS,IRDPTS,RAD,ILOOP,NRDPTS,NLOAD)
      CALL DECACC(RDPTS,NRDPTS,IRDPTS,NLOAD,IDEC,SPEED,V)
C NOW HAVE RDPTS WITH LOOP WHOSE START, LOAD, AND END POINTS ARE
C DEFINED BY IRDPTS, NLOAD, AND NRDPTS.  DECELERATION STARTS AT IDECEL.
C COMPUTE EN. NOTE THAT THE ACTUAL VOLUME IS TWICE Q BEFORE THE LOOP,
C SINCE Q IS THE NUMBER OF VEHICLES LOADED PER HOUR.
      DO 1 I=1,IRDPTS-1
 1    EN(I)=2.*Q/V(I)
      DO 2 I=IRDPTS,NRDPTS-1
 2    EN(I)=Q/V(I)
C OBTAIN VEHICLE LEVEL FOR EACH SEGMENT.
C THE THIRD ARGUMENT OF THE FUNCTION SUBROUTINE IS OPERATING
C MODE: 1=CRUISE AND ACCEL, 2=DECEL.
      DO 3 I=1,IDEC-1
 3    ELL(I)=ELVEH(IVEH,V(I),1)
      DO 4 I=IDEC,NLOAD-1
 4    ELL(I)=ELVEH(IVEH,V(I),2)
      DO 5 I=NLOAD,NRDPTS-1
 5    ELL(I)=ELVEH(IVEH,V(I),1)
C PUT ZEROES INTO EN AND ELL OF LAST POINT
      ELL(NRDPTS)=0.
      EN(NRDPTS)=0.
C COPY NRDPTS, NUMBER OF ROAD POINTS WITH LOOP EXTENSION,
C INTO IRDPTS:
      IRDPTS=NRDPTS
      RETURN
      END
```

```
      SUBROUTINE LOOP(RDPTS,IRDPTS,RAD,ITYPE,NRDPTS,NLOAD)
C PROGRAM TO PROVIDE END LOOP FOR A HAUL ROAD. THE LAST ROAD POINT,
C RDPTS(NRDPTS), IS THE STOPPING POINT. SIX TYPES OF END LOOPS ARE
C DEFINED: 1,2=LOOP ON ARRIVAL, 3,4= LOOP ON DEPARTURE, 5,6= LOOP ON
C BOTH, ODD=CLOCKWISE, EVEN=COUNTERCLOCKWISE. A 7 CONFIGURATION IS
C STRAIGHT IN AND OUT, NO LOOP.
C
C INPUT VARIABLES:
C    RDPTS(3,20) = COORDINATES OF POINTS DEFINING ROAD
C    IRDPTS = NUMBER OF INPUT ROAD POINTS
C    ITYPE = TYPE OF TURNAROUND LOOP; SEE DEFINITIONS ABOVE
C    RAD = RADIUS OF TURNAROUND LOOP
C OUTPUT VARIABLES:
C    NRDPTS = NUMBER OF POINTS IN ROAD EXTENDED BY LOOP
C    NLOAD = INDEX OF LOADING POINT; CORRESPONDS TO ORIGINAL LAST POINT
C    RDPTS IS ALSO EXTENDED/MODIFIED TO INCLUDE LOOP POINTS
C    IRDPTS IS MODIFIED TO IDENTIFY BRANCH POINT
C OTHER VARIABLES:
C    I = ROAD POINT/SEGMENT INDEX IN DO LOOPS
C    J = X,Y,Z INDEX IN DO LOOPS
C    XLOOP(2,14) = X,Y COORDINATES DEFINING LOOP SHAPES; TWO SHAPES
C                  ARE STORED, EACH ALIGNED FOR APPROACH ON X AXIS
C    IBOT,ITOP = PAIR OF  INDICES (EITHER1,6 OR 7,14) IDENTIFYING XLOOP
C                DATA BEING USED
C    I1 = IRDPTS-IBOT+1 ; SET UP SO THAT I1+I STARTS AT IRDPTS+1
C         WHEN I STARTS AT IBOT
C    X0,Y0,Z0 = TEMPORARY STORAGE OF LAST ORIGINAL ROAD POINT AT IRDPTS
C    X,Y = TEMPORARY STORAGE OF LOOP POINTS WHEN TRANSFORMING FROM LOOP
C          SHAPE COORDINATES TO ALIGNMENT WITH LAST INPUT ROAD SEGMENT
C    RDLNTH = LENGTH OF ROAD SEGMENT
C    ROTCS(2) = COSINE AND SINE OF ANGLE BETWEEN LAST ROAD SEGMENT AND
C               X AXIS
C    SIGN(2) =1,1 TO USE LOOP SHAPE AS STORED; 1,-1 TO USE MIRROR IMAGE
C    TEMP(2) = TEMPORARY STORAGE OF LOOP POINTS WHEN INVERTING ORDER FOR
C              LOOP ON APPROACH CASE
C    DUMMY(3) = DUMMY VARIABLE TO FILL UNUSED RETURN ARGUMENTS OF GEOM
C
      DIMENSION RDPTS(3,20),XLOOP(2,14),DUMMY(3)
      DIMENSION ROTCS(2),SIGN(2),TEMP(2)
      DATA XLOOP/.71,-.29,1.,-1.,.71,-1.71,0.,-2.,-.71,-1.71,-2.41,0.,
     1-1.71,.71,  -1.,1.,  -.29,.71,  0.,0.,  -.29,-.71,
     2  -1.,-1.,  -1.71,-.71,  -2.41,0./
      NRDPTS=IRDPTS
      NLOAD=IRDPTS
C TEST FOR LOOP OR NOT LOOP CASE
      IF(ITYPE.GE.1.AND.ITYPE.LE.6)GO TO 8
C STRAIGHT IN AND OUT CASE: BIFURCATE LAST SEGMENT
      NRDPTS=NRDPTS+1
      IRDPTS=IRDPTS-1
      DO 9 J=1,3
9     RDPTS(J,NRDPTS)=RDPTS(J,IRDPTS)
      RETURN
8     CONTINUE
      IF(ITYPE.LE.0.OR.ITYPE.GT.6) RETURN
```

```
C      SELECT SINGLE OR DOUBLE SIDED LOOP
       IBOT=1
       ITOP=6
       IF(ITYPE.LT.5) GO TO 1
       IBOT=7
       ITOP=14
 1     CONTINUE
C XLOOP DATA CORRESPOND TO CASES 3 AND 5. SET UP TO CHANGE SIGN OF Y
C FOR CASES 1,4 AND 6.  CASES 1 AND 2 WILL REQUIRE INVERSION OF ORDER
C AS WELL, SO THAT ARRIVAL POINTS COME BEFORE DEPARTURE.  INVERSION
C IS DONE AT END OF THE SUBROUTINE.
       SIGN(1)=1.
       SIGN(2)=1.
       IF(ITYPE.EQ.1.OR.ITYPE.EQ.4.OR.ITYPE.EQ.6)SIGN(2)=-1.
       I1=IRDPTS-IBOT+1
C      MULTIPLY XLOOP BY RAD
       DO 2 J=1,2
       DO 2 I=IBOT,ITOP
       RDPTS(J,I1+I)=RAD*XLOOP(J,I)*SIGN(J)
 2     CONTINUE
C      OBTAIN ROTATION ANGLE OF LAST SEGMENT
       CALL GEOM(RDPTS(1,IRDPTS-1),RDPTS(1,IRDPTS),DUMMY,
      1DUMMY,DUMMY,ROTCS,DUMMY,RDLNTH,3)
       IF(RDLNTH.LT.(2.5*RAD))WRITE(7,*)'LAST SEGMENT TOO SHORT
      1FOR SPECIFIED END LOOP'
C      ROTATE LOOP POINTS AND PLACE RE: END POINT
       X0=RDPTS(1,IRDPTS)
       Y0=RDPTS(2,IRDPTS)
       Z0=RDPTS(3,IRDPTS)
       DO 3 I=IBOT,ITOP
       X=RDPTS(1,I1+I)
       Y=RDPTS(2,I1+I)
       RDPTS(1,I1+I)=X*ROTCS(1)-Y*ROTCS(2)+X0
       RDPTS(2,I1+I)=Y*ROTCS(1)+X*ROTCS(2)+Y0
       RDPTS(3,I1+I)=Z0
 3     CONTINUE
C      ADJUST Z OF LAST NEW POINT
       RDPTS(3,I1+ITOP)=(2.41*RAD/RDLNTH)*(RDPTS(3,IRDPTS-1)-Z0)+Z0
C SHIFT ALL LOOP POINTS UP ONE TO MAKE ROOM FOR INSERTED POINT IN
C HALF-LOOP CASE.  SKIP THIS FOR FULL CIRCLE CASE.
       IF(IBOT.NE.1)GOTO 4
       DO 5 J=1,3
       DO 5 I=7,1,-1
 5     RDPTS(J,IRDPTS+I)=RDPTS(J,IRDPTS+I-1)
C      INCREASE ITOP TO MATCH
       ITOP=ITOP+1
 4     CONTINUE
C      INSERT NEW POINT/MOVE OLD LAST POINT
       DO 6 J=1,3
 6     RDPTS(J,IRDPTS)=RDPTS(J,I1+ITOP)
C      DEFINE NEW NUMBER OF POINTS
       NRDPTS=I1+ITOP
C      IDENTIFY LOADING POINT
       NLOAD=NRDPTS-6+(IBOT/3)
```

```
C     FINISHED EXCEPT FOR CASES 1 AND 2
      IF(ITYPE.GT.2)RETURN
C
C INVERT ORDER OF LOOP POINTS SO THAT LOOP COMES FIRST. THIS INVOLVES
C POINTS IRDPTS+1 THROUGH NRDPTS-1 (=IRDPTS+6).  ONLY X AND Y NEED
C BE SHIFTED; Z IS THE SAME FOR ALL OF THESE.
      DO 7 I=1,3
      DO 7 J=1,2
      TEMP(J)=RDPTS(J,IRDPTS+I)
      RDPTS(J,IRDPTS+I)=RDPTS(J,NRDPTS-I)
      RDPTS(J,NRDPTS-I)=TEMP(J)
 7    CONTINUE
C     CHANGE LOADING POINT INDEX
      NLOAD=NRDPTS-1
      RETURN
      END
```

```
      SUBROUTINE GEOM(X1,X2,XOBS,D,PHI,ROTCS,OBSROT,RDLNTH,ISKIP)
C THIS SUBROUTINE PERFORMS A COORDINATE TRANSFORMATION FROM THE INPUT
C COORDINATES TO COORDINATES RELATIVE TO A LINE DEFINED BY TWO POINTS.
C THE TRANSFORMED COORDINATES ARE CENTERED ON THE FIRST END POINT OF
C THE LINE AND ARE ROTATED SO THAT THE SECOND END LIES ON THE NEW
C X-AXIS.  THE SUBROUTINE RETURNS THE COSINE AND SINE OF THE ROTATION ANGLE,
C TRANSFORMED COORDINATES OF THE RECEIVER POSITION, ANGLES FROM THE RECEIVER
C TO THE LINE END POINTS, PLUS SEVERAL OTHER PERTINENT PARAMETERS.  A
C SWITCH 'ISKIP' PERMITS REUSING CERTAIN PARAMETERS, RATHER THAN CALCULATING
C THEM, WHEN THERE ARE REPETITIVE CALLS TO GEOM.  THE TRANSFORMATION TAKES
C PLACE IN THE X, Y PLANE; Z FROM THE RECEIVER POSITION IS TRANSFERRED TO THE
C TRANSFORMED COORDINTES.
C
C INPUT VARIABLES:
C     ISKIP = SWITCH VARIABLE.  0: COMPUTE ALL PARAMETERS.  1: USE INPUT
C             VALUES OF ROTCS.  2: USE INPUT VALUES OF ROTCS AND RDLNTH.
C             3: COMPUTE ROTCS AND RDLNTH ONLY.
C     XOBS(3) = RECEIVER LOCATION
C     X1(2),X2(2) = FIRST AND SECOND END POINTS OF LINE
C OUTPUT VARIABLES:
C     D = NORMAL DISTANCE FROM RECEIVER TO LINE
C     OBSROT(3) = TRANSFORMED COORDINATES OF RECEIVER
C     PHI(2) = ANGLES (RE: NORMAL) FROM RECEIVER TO ENDS OF LINE
C     RDLNTH = LENGTH OF LINE SEGMENT (INPUT IF ISKIP = 2)
C     ROTCS(2) = COSINE AND SINE OF TRANSFORMATION ANGLE (INPUT IF ISKIP=1,2)
C OTHER VARIABLES:
C     DELTX,DELTY = X AND Y DISPLACEMENTS BETWEEN X1 AND X2
C     ROTANG = TRANSFORMATION ROTATION ANGLE
C
      DIMENSION X1(2),X2(2),XOBS(3),PHI(2),OBSROT(3),ROTCS(2)
      COMMON /CONSTS/PI,TWOPI,PIOV2
      DATA PI,TWOPI,PIOV2/3.141592654,6.28318531,1.570796327/
C ISKIP=1 : USE LAST ROTANG AND ROTCS
C ISKIP=2 : USE LAST ROTANG,ROTCS, AND RDLNTH
      IF(ISKIP.EQ.1)GO TO 4
      IF(ISKIP.EQ.2)GO TO 5
C COMPUTE ANGLE TO END OF ROAD
      DELTX=X2(1)-X1(1)
      DELTY=X2(2)-X1(2)
C TEST FOR 90 DEGREE CASE
      IF(DELTX.NE.0.)GO TO 2
      ROTANG=PIOV2
C TEST FOR -90 DEGREE CASE
      IF(DELTY.LT.0.)ROTANG=ROTANG+PI
      GO TO 3
 2    CONTINUE
C PRINCIPLE VALUE
      ROTANG=ATAN(DELTY/DELTX)
C TEST FOR SECOND OR THIRD QUADRANTS
 3    CONTINUE
      IF(DELTX.LT.0.)ROTANG=ROTANG+PI
C COSINE AND SINE
      ROTCS(1)=COS(ROTANG)
      ROTCS(2)=SIN(ROTANG)
```

```
 4     CONTINUE
C LENGTH OF ROAD
      RDLNTH=(X2(1)-X1(1))*ROTCS(1)+(X2(2)-X1(2))*ROTCS(2)
      IF(ISKIP.EQ.3)RETURN
 5    CONTINUE
C TRANSFORM RECEIVER POSITION
      DELTX=XOBS(1)-X1(1)
      DELTY=XOBS(2)-X1(2)
      OBSROT(1)=DELTX*ROTCS(1)+DELTY*ROTCS(2)
      OBSROT(2)=DELTY*ROTCS(1)-DELTX*ROTCS(2)
C ANGLES PHI
C FIRST COMPUTE MAGNITUDE
C TEST FOR IN-LINE CASE
      IF(OBSROT(2).EQ.0.)GO TO 6
      PHI(1)=ATAN(ABS(OBSROT(1)/OBSROT(2)))
      PHI(2)=ATAN(ABS((OBSROT(1)-RDLNTH)/OBSROT(2)))
      GO TO 7
C IN-LINE CASE
 6    PHI(1)=PIOV2
      PHI(2)=PIOV2
 7    CONTINUE
C ASSIGN CORRECT SIGNS
      IF(OBSROT(1).GT.0.)PHI(1)=-PHI(1)
      IF(OBSROT(1).GT.RDLNTH)PHI(2)=-PHI(2)
C MAGNITUDE OF DISTANCE
      D=ABS(OBSROT(2))
C TRANSFER Z TO OBSROT
      OBSROT(3)=XOBS(3)
      RETURN
      END
```

```
      SUBROUTINE DECACC(RDPTS,NRDPTS,IRDPTS,NLOAD,IDEC,SPEED,V)
C PROGRAM TO PROVIDE AVERAGE SPEEDS ON HAUL ROAD ACCELERATION AND
C DECELERATION SEGMENTS.  SPEEDS ARE CALCULATED FOR CONSTANT ACCEL-
C ERATION, BEGINNING AT THE LOADING POINT.  THE CONSTANT-A
C SPEED PROFILE IS CONTINUED UNTIL THE SPEED AT THE FAR END
C OF THE SEGMENT EQUALS OR EXCEEDS THE APPROACH SPEED VTERM.
C VTERM IS INITIALLY SET AS THE INPUT SPEED AT THE LAST
C ROAD SEGMENT.  IF THE LAST SEGMENT (OR THE SEGMENTS FROM THE
C LOADING POINT TO THE END OF THE LOOP) IS NOT LONG ENOUGH TO
C REACH VTERM, THE LOOP SECTION IS EXTENDED BY A BIFURCA-
C TION OF THE APPROACHING SEGMENTS.  VTERM IS THEN TAKEN AS
C THE SPEED ON THE EXTENDED SEGMENT.  APPROACH AND DEPARTURE
C SEGMENTS ARE EXTENDED EQUALLY, SO THAT THE DIFFERENCES IN SOURCE
C LEVELS CAN BE PROPERLY HANDLED.  VTERM IS MODIFIED (IF
C NEEDED) SEPARATELY FOR EACH.  SPEEDS ARE FILLED IN WITH VTERM
C IF THE LOOP IS MORE THAN LONG ENOUGH.
C
C INPUT VARIABLES:
C   RDPTS(3,20) = COORDINATES OF ROAD POINTS
C   NRDPTS = NUMBER OF ROAD POINTS
C   IRDPTS = INDEX OF LOOP BRANCH POINT
C   NLOAD = INDEX OF LOADING POINT
C   SPEED(19) = SPEED ON SEGMENTS, IN INPUT UNITS
C OUTPUT VARIABLES:
C   IDEC = INDEX OF POINT WHERE DECELERATION BEGINS
C   V(19) = SPEED ON SEGMENTS, IN CONSISTENT UNITS
C   IRDPTS, NRDPTS ARE MODIFIED TO INDICATE NEW INDICES IF EXTENSION
C                   TO LOOP WAS NECESSARY TO ACCOMODATE ACCELERATION OR
C                   DECELERATION
C   RDPTS IS MODIFIED TO INCLUDE ADDED POINTS IF LOOP WAS EXTENDED
C OTHER VARIABLES:
C   ACCEL(2) = DECELERATION/ACCELERATION IN CONSISTENT UNITS
C   ACCRAT(2) = DECELERATION/ACCELERATION, IN G
C   I = ROAD POINT/SEGMENT INDEX IN DO LOOPS
C   J = X,Y,Z INDEX IN DO LOOPS
C   DX,DY = X AND Y SPANS OF SEGMENT
C   S = LENGTH OF SEGMENT
C   V1,V2 = SPEEDS AT BEGINNING AND END OF SEGMENT UNDER CONSTANT A
C   VTERM = APPROACH SPEED, AS DESCRIBED ABOVE
C   IACCEL, IDECEL = INDICES MARKING END POINTS OF ACCELERATION AND
C                    DECELERATION BRANCHES.  THESE ARE TEMPORARY
C                    WORKING VALUES INCREMENTED ONE STEP AT A TIME
C                    AND TRANSFERRED INTO NRDPTS AND IRDPTS ON RETURN
C   VBUF(19) = TEMPORARY STORAGE OF CALCULATED SPEEDS.  TRANSFERRED
C              INTO V BEFORE RETURNING
C
      DIMENSION ACCEL(2),SPEED(19),VBUF(19)
      DIMENSION RDPTS(3,20),V(19)
      COMMON/UNITS/DO,DO2,GRAV,VELCON
      COMMON /KINEM/ACCRAT(2)
C CONVERT ACCELERATIONS FROM G'S TO CONSISTENT UNITS
      ACCEL(1)=ACCRAT(1)*GRAV
      ACCEL(2)=ACCRAT(2)*GRAV
C MAKE DECEL AND ACCEL POINTS CORRESPOND TO LOOP ENDS
```

```
      IACCEL=NRDPTS
      IDECEL=IRDPTS
C CONVERT SPEED TO CONSISTENT UNITS
      DO 1 I=1,NRDPTS-1
      V(I)=SPEED(I)*VELCON
 1    CONTINUE
C
C SPEED CALCULATION SECTION
C
      VTERM=V(IRDPTS-1)
      IF(IRDPTS.EQ.1)VTERM=V(1)
C
C DO DECELERATION LOOP FIRST
      V1=0.
      DO 2 I=NLOAD-1,IRDPTS,-1
C TEST FOR V ALREADY PAST VTERM
      IF(V1.GE.VTERM) GO TO 3
C TAKE IDEC AS I; THE LAST ONE THROUGH WILL BE LEFT AS IDEC
      IDEC=I
C LENGTH OF SEGMENT
      DX=RDPTS(1,I+1)-RDPTS(1,I)
      DY=RDPTS(2,I+1)-RDPTS(2,I)
      S=SQRT(DX**2+DY**2)
C SPEED AT END
      V2=SQRT(V1**2+2.*S*ACCEL(1))
C TEST FOR MERGE WITH VTERM
      IF(V2.GE.VTERM) GO TO 4
C AVERAGE SPEED
      VBUF(I)=(V1+V2)/2.
      V1=V2
      GO TO 2
 4    CONTINUE
C EXPRESSION FOR SECTION WHERE V MERGES INTO VTERM
      VBUF(I)=(VTERM*V2-(VTERM**2+V1**2)*.5)/(V2-V1)
C DE-DECREMENT IDEC IF MOST OF THIS SEGMENT IS CRUISE
      IF((V2-VTERM).GT.(VTERM-V1))IDEC=IDEC+1
      V1=V2
      GO TO 2
 3    CONTINUE
C FILL VTERM INTO REST OF  LOOP
      VBUF(I)=VTERM
 2    CONTINUE
C DECELERATION VELOCITIES ARE COMPLETE WITHIN LOOP. THE FOLLOWING
C SUBSECTION CONTINUES THE PROCESS IF THE APPROACH VELOCITY HAS NOT
C YET BEEN MATCHED.  SUCCESSIVE ELEMENTS BEFORE THE LOOP BRANCH ARE
C SPLIT AND INCORPORATED INTO THE LOOP.
C
C SKIP IF VTERM HAS BEEN MET OR NO MORE SEGMENTS
      IF(V1.GE.VTERM.OR.IRDPTS.EQ.1)GO TO 6
      DO 7 I=IRDPTS-1,1,-1
      VTERM=V(I)
C TEST FOR COMPLETION
      IF(V1.GE.VTERM) GO TO 6
      IDEC=I
```

```
C LENGTH OF SEGMENT
      DX=RDPTS(1,I+1)-RDPTS(1,I)
      DY=RDPTS(2,I+1)-RDPTS(2,I)
      S=SQRT(DX**2+DY**2)
C SPEED AT END
      V2=SQRT(V1**2+2.*S*ACCEL(1))
C TEST FOR MERGE WITH VTERM
      IF(V2.GE.VTERM) GO TO 9
C AVERAGE SPEED
      VBUF(I)=(V1+V2)/2.
      V1=V2
      GO TO 10
 9    CONTINUE
C MERGE SECTION EXPRESSION
      VBUF(I)=(VTERM*V2-(VTERM**2+V1**2)*.5)/(V2-V1)
C DE-DECREMENT IDEC IF MOST OF THIS SEGMENT IS CRUISE
      IF((V2-VTERM).GT.(VTERM-V1))IDEC=IDEC+1
      V1=V2
 10   CONTINUE
C EXTEND LOOP SECTION
      IDECEL=IDECEL-1
      IACCEL=IACCEL+1
      DO 11 J=1,3
 11   RDPTS(J,IACCEL)=RDPTS(J,IDECEL)
      V(IACCEL-1)=V(IDECEL)
 7    CONTINUE
 6    CONTINUE
C
C DECELERATION VELOCITIES ARE NOW COMPLETED. THE LOOP SEGMENT IS FROM
C IDECEL TO IACCEL, WHICH WERE INCREASED (IF NECESSARY) FROM IRDPTS
C AND NRDPTS.  VELOCITIES ARE STORED IN VBUF. INPUT VELOCITIES,
C V, REMAIN INTACT.  THE START OF DECELERATION IS RECORDED AS IDEC.
C
C ACCELERATION VELOCITIES WILL NOW BE COMPUTED, USING ESSENTIALLY THE
C SAME LOGIC.
C
      VTERM=V(IRDPTS-1)
      IF(IRDPTS.EQ.1)VTERM=V(1)
      V1=0.
      DO 12 I=NLOAD,IACCEL-1
      IF(I.GT.NRDPTS)VTERM=V(I)
      IF(V1.GE.VTERM)GO TO 13
      DX=RDPTS(1,I+1)-RDPTS(1,I)
      DY=RDPTS(2,I+1)-RDPTS(2,I)
      S=SQRT(DX**2+DY**2)
      V2=SQRT(V1**2+2.*S*ACCEL(2))
      IF(V2.GE.VTERM) GO TO 14
      VBUF(I)=(V1+V2)/2.
      V1=V2
      GO TO 12
 14   CONTINUE
      VBUF(I)=(VTERM*V2-(VTERM**2+V1**2)*.5)/(V2-V1)
      V1=V2
      GO TO 12
```

```
 13   CONTINUE
      VBUF(I)=VTERM
 12   CONTINUE
C ACCELERATION COMPLETE WITHIN LOOP AS EXTENDED BY DECELERATION.
C EXTEND FURTHER IF NEEDED, AS BEFORE.
      IF(V1.GE.VTERM) GO TO 16
      DO 17 I=IACCEL,19
      VTERM=V(I)
      IF(V1.GE.VTERM.OR.IDECEL.EQ.1) GO TO 16
      DX=RDPTS(1,I+1)-RDPTS(1,I)
      DY=RDPTS(2,I+1)-RDPTS(2,I)
      S=SQRT(DX**2+DY**2)
      V2=SQRT(V1**2+2.*S*ACCEL(2))
      VBUF(I)=(V1+V2)/2.
      IF(V2.GE.VTERM) GO TO 19
      V1=V2
      GO TO 20
 19   CONTINUE
      VBUF(I)=(VTERM*V2-(VTERM**2+V1**2)*.5)/(V2-V1)
      V2=V1
 20   CONTINUE
      IACCEL=IACCEL+1
      IDECEL=IDECEL-1
      DO 21 J=1,3
 21   RDPTS(J,IACCEL)=RDPTS(J,IDECEL)
      V(IACCEL-1)=V(IDECEL)
 17   CONTINUE
 16   CONTINUE
C
C CALCULATIONS COMPLETED. TRANSFER IACCEL AND IDECEL TO NRDPTS AND
C IRDPTS TO DEFINE TOTAL NUMBER OF POINTS AND LOCATION OF BRANCH
C POINTS, COPY VBUF INTO V, AND RETURN.
      IRDPTS=IDECEL
      NRDPTS=IACCEL
      DO 22 I=IDECEL,IACCEL-1
 22   V(I)=VBUF(I)
      RETURN
      END
```

```
      FUNCTION ELVEH(IVEH,V,MODE)
C     THIS FUNCTION SUBROUTINE COMPUTES SOUND PRESSURE LEVEL
C     AT DISTANCE DO FOR VEHICLE TYPE IVEH AT
C     SPEED V, DURING MODE 1(CRUISE AND ACCEL) OR 2 (DECEL)
C     A LOG SPL RELATION IS FOLLOWED, WITH A PLATEAU
C     BELOW SPEED VCRIT FOR MODE 1. PARAMETERS
C     DEFINING THE FUNCTION - SLOPE,LEVEL AT VREF,
C     AND VCRIT - ARE CONTAINED IN THE BLOCK DATA PROGRAM.
      COMMON /UNITS/DO,DO2,GRAV,VELCON
      COMMON /VEHLEV/HAULEQ(6,10)
C   HAULEQ(1,I) = EMISSION LEVEL AT DO
C   HAULEQ(2,I) = REFERENCE SPEED
C   HAULEQ(3,I) = SLOPE
C   HAULEQ(4,I) = VCRIT
C   HAULEQ(5,I) = CAPACITY
C   HAULEQ(6,I) = EFFECTIVE ACOUSTIC HEIGHT
C     DATA IN COMMON BLOCK ARE IN INPUT UNITS.
C     INPUT V IS IN CONSISTENT UNITS.
      VEE=V/VELCON
      IF(MODE.EQ.1)VEE=AMAX1(VEE,HAULEQ(4,IVEH))
      ELVEH=HAULEQ(1,IVEH)+HAULEQ(3,IVEH)*ALOG10(VEE/HAULEQ(2,IVEH))
      RETURN
      END
```

```
      SUBROUTINE PASSBY(XLNPTS,NLNPTS,ISTOP,IDEC,SPEED,Q,IVEH
     1,EN,ELL)
C PROGRAM TO COMPUTE LINE SOURCE QUANTITIES EN AND ELL FOR A STRAIGHT-
C THROUGH HAUL OPERATION, WITH NO MODIFICATIONS TO XLNPTS.  A USER-DEFINED
C LOOP MAY BE CREATED BY USING THIS OPTION WITH APPROPRIATE INPUT
C VALUES OF XLNPTS.
C
C ACCELERATION/DECELERATION SECTIONS MAY BE INPUT BY THE USER OR
C COMPUTED AS AN OPTION.  A STOPPING POINT IS INDICATED BY A POSITIVE VALUE
C OF 'ISTOP'; A NEGATIVE VALUE INDICATES CRUISE ON ALL SEGMENTS.  ONLY
C ONE STOPPING POINT MAY BE SPECIFIED.  WHEN A STOPPING POINT IS SPECIFIED,
C 'IDEC' DEFINES THE POINT WHERE DECELERATION BEGINS.  PROVIDING A
C POSITIVE VALUE OF 'IDEC' INDICATES THAT APPROPRIATE AVERAGE SPEEDS
C ON THE DECELERATION/ACCELERATION SEGMENTS HAVE BEEN PROVIDED AS
C INPUT DATA.  A NEGATIVE VALUE OF 'IDEC' INDICATES THAT THE PROGRAM IS
C TO COMPUTE THESE.  CONSTANT-ACCELERATION SPEED PROFILES ARE COMPUTED,
C USING THE SAME ALGORITHMS AS IN 'DECACC'.  THE PROFILES ARE CONTINUED
C UNTIL THE SPEED MATCHES THE INPUT VALUE OF SPEED ON THE SEGMENT BEING
C CONSIDERED, OR UNTIL THE END OF THE ROAD IS REACHED, WHICHEVER COMES
C FIRST.  THE SPEEDS INPUT BY THE USER SHOULD BE THE APPROACH AND DEPARTURE
C CRUISE SPEEDS.
C
C INPUT VARIABLES:
C     IDEC = INDEX OF DECELERATION POINT (MAY BE OUTPUT; SEE ABOVE)
C     ISTOP = INDEX OF STOPPING POINT
C     IVEH = IDENTIFYING INDEX OF VEHICLE
      DIMENSION XLNPTS(3,20),SPEED(19),EN(19),ELL(19),V(19)
      DIMENSION ACCEL(2)
      COMMON /UNITS/DO,DO2,GRAV,VELCON
      COMMON /KINEM/ACCRAT(2)
      ACCEL(1)=ACCRAT(1)*GRAV
      ACCEL(2)=ACCRAT(2)*GRAV
C CONVERT SPEED TO CONSISTENT UNITS
      DO 1 I=1,NLNPTS
1     V(I)=SPEED(I)*VELCON
C TRANSFER TO EN/ELL CALCULATION IF NO STOPPING POINT
      IF(ISTOP.LE.0)GO TO 2
C ALSO TRANSFER IF DECELERATION POINT IS SPECIFIED
      IF(IDEC.GE.1)GO TO 2
C COMPUTE DECELERATION VELOCITIES
      V1=0.
      DO 3 I=ISTOP-1,1,-1
      VTERM=V(I)
C IF NEW V(I) IS SLOWER THAN END OF LAST SEGMENT, DECELERATION FINISHED.
      IF(V1.GE.VTERM)GO TO 4
C SET IDEC
      IDEC=I
C LENGTH OF SEGMENT
      DX=XLNPTS(1,I+1)-XLNPTS(1,I)
      DY=XLNPTS(2,I+1)-XLNPTS(2,I)
      DS=SQRT(DX**2+DY**2)
C SPEED AT END
      V2=SQRT(V1**2+2.*DS*ACCEL(1))
C TEST FOR MERGE WITH VTERM
```

```
      IF(V2.GE.VTERM)GO TO 5
C AVERAGE SPEED
      V(I)=(V1+V2)/2.
      V1=V2
      GO TO 3
 5    CONTINUE
C MERGE SECTION EXPRESSION
      V(I)=(VTERM*V2-(VTERM**2+V1**2)*.5)/(V2-V1)
      IF((V2-VTERM).GT.(VTERM-V1))IDEC=IDEC+1
      V1=V2
 3    CONTINUE
 4    CONTINUE
C DECELERATION CALCULATION COMPLETE.  CALCULATE ACCELERATION SEGMENT
C SPEEDS, USING THE SAME LOGIC.
      V1=0.
      DO 6 I=ISTOP,NLNPTS-1
      VTERM=V(I)
      IF(V1.GE.VTERM)GO TO 7
      DX=XLNPTS(1,I+1)-XLNPTS(1,I)
      DY=XLNPTS(2,I+1)-XLNPTS(2,I)
      DS=SQRT(DX**2+DY**2)
      V2=SQRT(V1**2+2.*DS*ACCEL(2))
      IF(V2.GE.VTERM)GO TO 8
      V(I)=(V1+V2)/2.
      V1=V2
      GO TO 6
 8    CONTINUE
      V(I)=(VTERM*V2-(VTERM**2+V1**2)*.5)/(V2-V1)
      V1=V2
 6    CONTINUE
 7    CONTINUE
C ALL ACCELERATION/DECELERATION SPEEDS ARE COMPLETE
C
 2    CONTINUE
C COMPUTE EN
      DO 9 I=1,NLNPTS-1
 9    EN(I)=Q/V(I)
      EN(NLNPTS)=0.
C OBTAIN VEHICLE LEVEL FOR EACH SEGMENT.  THERE ARE THREE DOMAINS:
C 1 TO IDEC-1, IDEC TO ISTOP-1, AND ISTOP TO NLNPTS-1.  SET UP DUMMY
C LIMITS IF IDEC AND/OR ISTOP ARE NEGATIVE.
      IDEC=MAX0(IDEC,1)
      ISTOP=MAX0(ISTOP,1)
C IF ONE OR BOTH OF THESE ARE SET TO 1, ELL(1) COMPUTED IN 10 AND/OR
C 11 LOOP WILL BE OVERWRITTEN IN SUBSEQUENT LOOP(S)
C
      DO 10 I=1,IDEC-1
 10   ELL(I)=ELVEH(IVEH,V(I),1)
      DO 11 I=IDEC,ISTOP-1
 11   ELL(I)=ELVEH(IVEH,V(I),2)
      DO 12 I=ISTOP,NLNPTS-1
 12   ELL(I)=ELVEH(IVEH,V(I),1)
      ELL(NLNPTS)=0.
      RETURN
      END
```

```
      SUBROUTINE ARTASK(CLPTS,WIDTH,NCLPTS,ELAR,IFRQ)
C THIS SUBROUTINE ACCEPTS THE TASK DESCRIPTION - GEOMETRY AND 'INFO' -
C AND RETURNS AN ARRAY 'ELAR' OF LEVELS IN EACH SEGMENT.  EQUIPMENT
C LOOK-UP AND PRODUCTION MATCHING ARE HANDLED EXACTLY AS IN PTTASK.
C MULTIPLE SOURCES ARE PERMITTED, AND TOTAL NOISE EMISSION IS
C APPORTIONED ALONG THE SEGMENTS BY AREA.
C
C IN THE FOLLOWING VARIABLE DICTIONARY, NOTE THAT SOME DATA ARE PASSED TO
C AND FROM OTHER ROUTINES THROUGH COMMON BLOCKS.  ALL DEFINED VARIABLES WITHIN
C COMMONS ARE INPUT OR OUTPUT; THEIR ROLE VARIES DEPENDING ON WHETHER
C NEW INFORMATION IS BEING CREATED FOR A PARTICULAR TASK
C
C INPUT VARIABLES:
C     CLPTS(3,20) = COORDINATES OF CENTERLINE POINTS
C     NCLPTS = NUMBER OF CENTERLINE POINTS
C     WIDTH(20) = WIDTH OF AREA AT EACH CENTERLINE POINT
C DATA INPUTS:
C     EQUIP(5,10,30) = EQUIPMENT NOISE AND PRODUCTION PARAMETERS  /EQUIPT/
C     EN = NUMBER OF PIECES OF EQUIPMENT IN AREA
C     IFREQ(10,30) = EQUIPMENT SPECTRUM DATA BASE
C OUTPUT VARIABLES:
C     ELAR(19) = TOTAL EMISSION LEVEL OF EQUIPMENT IN AREA SEGMENTS
C     IFRQ = SPECTRUM IDENTIFIER OF SOURCE
C OTHER VARIABLES:
C     AREA(19) = AREA OF EACH AREA SEGMENT
C     AREAT = TOTAL AREA OVER ALL SEGMENTS
C     DAYHRS = NUMBER OF HOURS IN FULL WORK DAY  /WKDAY/
C     DS = LENGTH OF CENTERLINE SEGMENT
C     DX,DY = X AND Y DIFFERENCES BETWEEN CENTERLINE POINTS
C     EL = TOTAL EMISSION LEVEL OF ALL PIECES OF EQUIPMENT IN AREA
C     HOURS = EQUIVALENT FULL TIME HOURS WORKED PER DAY  /TSKARG/
C     I = EQUIPMENT TYPE INDEX
C     IHAUL(10),IVEH(5,5) = (NOT USED HERE)  /TYPES/
C     INFO(2) = EQUIPMENT MODEL AND TYPE (M,I)    /TSKARG/
C     IPROD(30) = VALUE OF 1 INDICATES A PRODUCTION RATE EXISTS   /TYPES/
C     J,K = DO LOOP INDICES
C     M = MODEL NUMBER OF EQUIPMENT
C     NMODS(30) = NUMBER OF DEFINED MODELS OF EACH EQUIPMENT TYPE  /NMODLS/
C     PROD = PRODUCTION RATE PER FULL WORKDAY
C     PRODUC = DAILY PRODUCTION, ACTUAL ACTIVITY LEVEL  /TSKARG/
C
      DIMENSION CLPTS(3,20),WIDTH(20),AREA(19),ELAR(19)
      COMMON /WKDAY/DAYHRS
      COMMON /TYPES/IPROD(30),IHAUL(10),IVEH(5,5)
      COMMON /EQUIPT/EQUIP(5,10,30),IFREQ(10,30)
      COMMON /NMODLS/NMODS(30),NVTYP
      COMMON /TSKARG/INFO(2),HOURS,PRODUC
C
      M=INFO(1)
      I=INFO(2)
      IF(M.GT.0)GO TO 1
C SECTION TO ADD NEW MODEL DATA
C INCREMENT NMODS
      NMODS(I)=NMODS(I)+1
```

```
      M=NMODS(I)
C READ NEW DATA
      WRITE(7,*)'ENTER LMAX,DELTA,CAPACITY,CYCLE TIME, ACOUSTIC
     1 HEIGHT, AND FREQUENCY:'
      READ(7,*)(EQUIP(K,M,I),K=1,5),IFREQ(M,I)
 1    CONTINUE
C CALCULATE LEQ(CYCLE)
      EL=EQUIP(1,M,I)-EQUIP(2,M,I)
C TEST TO SEE IF THERE IS A PRODUCTION RATE
      DO 2 J=1,30
      IF(IPROD(J).EQ.I)GO TO 3
 2    CONTINUE
      GO TO 4
 3    CONTINUE
C CALCULATE DAILY PRODUCTION
      PROD=EQUIP(3,M,I)*DAYHRS/EQUIP(4,M,I)
C COMPUTE EFFECTIVE 'HOURS' IF BALANCED TO LAST
      IF(HOURS.LT.0)HOURS=DAYHRS*PRODUC/PROD
C COMPUTE PRODUC
      PRODUC=PROD*HOURS/DAYHRS
 4    CONTINUE
C
      WRITE(7,*)'HOW MANY?'
      READ(7,*)EN
      EL=EL+10.*ALOG10(EN*HOURS/DAYHRS)
C
C COMPUTE AREAS FOR APPORTIONMENT
      AREAT=0.
      DO 5 J=1,NCLPTS-1
      DX=CLPTS(1,J+1)-CLPTS(1,J)
      DY=CLPTS(2,J+1)-CLPTS(2,J)
      DS=SQRT(DX**2+DY**2)
      AREA(J)=DS*(WIDTH(J+1)+WIDTH(J))/2.
      AREAT=AREAT+AREA(J)
 5    CONTINUE
C APPORTION EL AND ADD ACOUSTIC HEIGHT TO Z
      Z=EQUIP(5,M,I)
      DO 6 J=1,NCLPTS-1
      ELAR(J)=EL+10.*ALOG10(AREA(J)/AREAT)
      CLPTS(3,J)=CLPTS(3,J)+Z
 6    CONTINUE
      CLPTS(3,NCLPTS)=CLPTS(3,NCLPTS)+Z
C GET SPECTRUM IDENTIFIER
      IFRQ=IFREQ(M,I)
      RETURN
      END
```

```
      BLOCK DATA DATA1
C THIS DATA BLOCK CONTAINS VARIOUS MISCELLANEOUS DATA ITEMS SUCH AS
C UNIT DECLARATIONS, WORKDAY LENGTH, ETC.
      COMMON /KINEM/ACCRAT(2)
      COMMON /UNITS/DO,DO2,GRAV,VELCON
      COMMON /WKDAY/ DAYHRS
      DATA ACCRAT/.1,.1/
      DATA DO,DO2,GRAV,VELCON/50.,2500.,32.2,1.47/
      DATA DAYHRS/8./
      END
```

```
      BLOCK DATA DATA2
C THIS DATA BLOCK CONTAINS ALL EQUIPMENT AND TASK DATA - NOISE LEVELS,
C CAPACITIES, ETC - REQUIRED BY THE TASK MODELS.  EMPTY SPACES ARE
C LEFT IN THE ARRAYS FOR FUTURE EXPANSION.  NOTE THAT WHEN MORE DATA
C ARE INSERTED, THE COUNTS FOR EACH MODEL (NMODS AND NVTYP) MUST BE
C ADJUSTED AS WELL.
C
      COMMON/EQUIPT/BACKHO(5,10),FLOADE(5,10),COMPRE(5,10),PILEDR(5,10),
     1PUMP(5,10),CRANE(5,10),SHOVEL(5,10),BREAKE(5,10),
     2CONCRE(5,10),GENERA(5,10),FMISCE(5,10),EMPTY1(5,10,4),
     3BULLDO(5,10),GRADER(5,10),TRUCKS(5,10),SCRAPE(5,10),
     4EMPTY2(5,10,6),COMPAC(5,10),PAVING(5,10),EMPTY3(5,10,3)
     5,IFREQ(10,30)
C
      COMMON /VEHLEV/HAULEQ(6,10)
      COMMON /NMODLS/NMODS(30),NVTYP
      COMMON /TYPES/IPROD(30),IHAUL(10),IVEH(5,5)
C
C HAUL EQUIPMENT   LEVEL    VREF     SLOPE    VCRIT     CAP   HEIGHT
      DATA HAULEQ/  76.   ,  35.   ,  20.   ,  35.   ,  10.  ,   8.   ,     10YTRUCK
     1              81.   ,  35.   ,  20.   ,  35.   ,  10.  ,   8.   ,     10YTRUCK
     2              86.   ,  35.   ,  20.   ,  35.   ,  40.  ,   8.   ,     2*20Y TR
     3              90.7  ,  35.   ,  20.   ,  35.   ,  10.  ,   8.   ,     NOM TR *
     4              84.   ,  30.   ,   0.   ,  30.   ,  25.  ,   6.   ,     CAT631M
     5              95.   ,  30.   ,   0.   ,  30.   ,  25.  ,   6.   ,     CAT631NM
     6              90.   ,  30.   ,   0.   ,  30.   ,  17.  ,   6.   ,     CAT623
     7              81.   ,  30.   ,   0.   ,  30.   ,  25.  ,   6.   ,     CAT
     8 12*0./
      DATA NVTYP/8/
C
C EQUIPMENT DATA   LMAX     DELTA     CAP     CYCLE  HEIGHT
      DATA BACKHO/  86.5 ,    3.   ,   1.   , .00833,   6.   ,                NOMINAL*
     1              88.  ,    3.   ,   1.   , .00833,   6.   ,                CAT,KOEH
     2              92.  ,    3.   ,   1.   , .00833,   6.   ,                PH
     3                                                 35*0./,NMODS(1)/3/
      DATA FLOADE/  89.  ,    5.   ,   5.   , .00833,   6.   ,                NOMINAL*
     1              81.  ,    5.   ,   3.   , .00833,   6.   ,                3YD
     2              82.  ,    5.   ,   5.   , .00833,   6.   ,                5YD
     3              83.  ,    5.   ,   7.   , .00833,   6.   ,                7YD
     4              85.  ,    5.   ,  10.   , .00833,   6.   ,                10YD
     5                                                 25*0./,NMODS(2)/5/
      DATA COMPRE/  91.3 ,    2.   ,   0.   ,   0.  ,   4.   ,                NOMINAL*
     1              88.  ,    2.   ,   0.   ,   0.  ,   4.   ,                STANDARD
     2              77.  ,    2.   ,   0.   ,   0.  ,   4.   ,                QUIET,DC
     3              67.  ,    2.   ,   0.   ,   0.  ,   4.   ,                QUIET,DC
     4                                                 30*0./,NMODS(3)/4/
      DATA PILEDR/  97.3 ,    6.   ,   0.   ,   0.  ,  20.   ,                NOMINAL*
     1             103.  ,    6.   ,   0.   ,   0.  ,  20.   ,                PILEDR
     2                                                 40*0./,NMODS(4)/2/
      DATA PUMP  /  63.  ,    0.   ,   0.   ,   0.  ,   4.   ,                PUMP1
     1              76.  ,    0.   ,   0.   ,   0.  ,   4.   ,                PUMP2
     2                                                 40*0./,NMODS(5)/2/
      DATA CRANE /  89.  ,   7.5   ,   0.   ,   0.  ,  15.   ,                NOMINAL*
     1              73.  ,   7.5   ,   0.   ,   0.  ,  15.   ,                QUIE
```

```
     2                 81.5 ,  7.5  ,   0.  ,   0.  ,  15.  ,              MEDIUM
     3                 85.  ,  7.5  ,   0.  ,   0.  ,  15.  ,              LOUD
     4                                              30*0./,NMODS(6)/4/
      DATA BREAKE/     96.  ,   7.  ,   0.  ,   0.  ,   2.  ,              RK DRL *
     1                 87.  ,   7.  ,   0.  ,   0.  ,   2.  ,              JACKHAM
     2                 76.  ,   7.  ,   0.  ,   0.  ,   2.  ,              MUFJHAM
     3                                              35*0./,NMODS(8)/3/
      DATA CONCRE/     78.  ,   5.  ,   0.  ,   0.  ,  10.  ,              POUR1
     1                 90.  ,   0.  ,   0.  ,   0.  ,  10.  ,              BATCH*
     2                 82.  ,   0.  ,   0.  ,   0.  ,  10.  ,              BATCH1
     3                 88.  ,   0.  ,   0.  ,   0.  ,   6.  ,              PUMP
     4                 82.8 ,   0.  ,   0.  ,   0.  ,   8.  ,              MIXER*
     5                                              25*0./,NMODS(9)/5/
      DATA GENERA/     73.5 ,   0.  ,   0.  ,   0.  ,   4.  ,              AVERAGE
     1                 81.  ,   0.  ,   0.  ,   0.  ,   4.  ,              NOMINAL*
     2                                              40*0./,NMODS(10)/2/
      DATA FMISCE/     71.  ,   1.  ,   0.  ,   0.  ,   2.  ,              GRINDER
     1                 88.  ,   0.  ,   0.  ,   0.  ,   1.  ,              CONC SAW
     2                 83.  ,   0.  ,   0.  ,   0.  ,   4.  ,              FAN
     3                 71.  ,   0.  ,   0.  ,   0.  ,   4.  ,              WELDER*
     4                                              30*0./,NMODS(11)/4/
      DATA BULLDO/     90.1 ,   2.  ,   0.  ,   0.  ,   6.  ,              NOMINAL*
     1                 80.  ,   2.  ,   0.  ,   0.  ,   6.  ,              D6,7,8
     2                 85.  ,   2.  ,   0.  ,   0.  ,   6.  ,              D9,MUF
     3                 96.  ,   2.  ,   0.  ,   0.  ,   6.  ,              D9,NMUF
     4                                              30*0./,NMODS(16)/4/
      DATA GRADER/     83.  ,   0.  ,   0.  ,   0.  ,   8.  ,              GRADER
     1                                              45*0./,NMODS(17)/1/
      DATA COMPAC/     80.  ,   0.  ,   0.  ,   0.  ,   8.  ,              QUIET
     1                 86.  ,   0.  ,   0.  ,   0.  ,   8.  ,              MEDIUM
     2                 93.  ,   0.  ,   0.  ,   0.  ,   8.  ,              LOUD
     3                                              35*0./,NMODS(26)/3/
      DATA PAVING/     83.8 ,   0.  ,   0.  ,   0.  ,   4.  ,              NOMINAL*
     1                 82.8 ,   0.  ,   0.  ,   0.  ,   4.  ,              CONCRETE
     2                 82.5 ,   0.  ,   0.  ,   0.  ,   4.  ,              ASPHALT
     3                                              35*0./,NMODS(27)/3/
C
C FREQUENCY DATA
      DATA IFREQ/20*500,10*1000,10*1500,10*800,
     1           20*500,10*1500,10*500,20*1200,
     2           40*0,40*500,60*0,10*630,10*500,30*0/
C
      DATA IPROD/1,2,6,7,9,18,19,23*0/
      DATA IHAUL/18,19,8*0/
      DATA IVEH/5*0,5*0,1,2,3,4,0,5,6,7,8,0,5*0/
      END
```

```
      SUBROUTINE CROSS(XOBS,XSRC,B1,B2,X,ICROSS)
C PROGRAM TO FIND THE INTERSECTION, X, BETWEEN A BARRIER B1-B2 AND THE
C SOURCE TO OBSERVER LINE OF SIGHT XOBS-XSRC.  IF SHIELDING OCCURS,
C ICROSS IS SET EQUAL TO 1; IF NOT, IT IS SET EQUAL TO ZERO.  ICROSS
C IS INITIALIZED AT 0, AND RETURNS OCCUR EACH TIME A NECESSARY CONDITION
C IS NOT SATISFIED.  THE CROSSING POINT IS FIRST LOCATED IN THE X-Y
C PLANE.  ONCE FOUND, ITS Z VALUE IS TAKEN AS THE BARRIER
C HEIGHT AT THAT POINT
C
C INPUT VARIABLES:
C     B1(3),B2(3) = COORDINATES OF BARRIER SECTION END POINTS
C     XOBS(3) = COORDINATES OF RECEIVER LOCATION
C     XSRC(3) = COORDINATES OF SOURCE LOCATION
C OUTPUT VARIABLES:
C     ICROSS = TEST FLAG; =1 IF A CROSSING POINT HAS BEEN FOUND, 0 IF NOT
C     X(3) = COORDINATES OF CROSSING POINT; X(3) IS ON BARRIER
C OTHER VARIABLES:
C     CRSO = RIGHT HAND SIDE OF RECEIVER-SOURCE EQUATION
C     CR21 = RIGHT HAND SIDE OF BARRIER LINE EQUATION
C     DELSO(2) = DELTAX AND DELTAY OF RECEIVER-SOURCE LINE
C     DEL21(2) = DELTAX AND DELTAY OF BARRIER LINE
C     DET = DETERMINANT OF THE TWO LINE EQUATIONS
C     DO(2) = DISTANCE FROM RECEIVER TO CROSSING POINT X
C     DS(2) = DISTANCE FROM SOURCE TO CROSSING POINT X
C     D1(2),D2(2) = DISTANCES FROM BARRIER END POINTS B1,B2 TO X
C     K = INDEX OF DO LOOPS
C
      DIMENSION XOBS(3),XSRC(3),B1(3),B2(3),DELSO(2),DEL21(2)
      DIMENSION X(3),DO(2),DS(2),D1(2),D2(2)
      ICROSS=0
C SET UP COEFFICIENTS FOR EQUATIONS OF THE TWO LINES.  EQUATIONS ARE
C OF THE FORM   X*DELTAY - Y*DELTAX = X1*Y2 - Y1*X2
      DO 1 K=1,2
      DELSO(K)=XSRC(K)-XOBS(K)
      DEL21(K)=B2(K)-B1(K)
 1    CONTINUE
      CRSO=XOBS(1)*XSRC(2)-XOBS(2)*XSRC(1)
      CR21=B1(1)*B2(2)-B2(1)*B1(2)
C COMPUTE DETERMINANT, TEST FOR ZERO
      DET=DEL21(2)*DELSO(1)-DELSO(2)*DEL21(1)
      IF(DET.EQ.0.)RETURN
C SOLVE FOR CROSSING POINT, USING CRAMER'S RULE, AND SET UP DIFFERENCES
C FROM END POINTS
      DO 2 K=1,2
      X(K)=(DELSO(K)*CR21-DEL21(K)*CRSO)/DET
      DO(K)=X(K)-XOBS(K)
      DS(K)=X(K)-XSRC(K)
      D1(K)=X(K)-B1(K)
      D2(K)=X(K)-B2(K)
 2    CONTINUE
C TEST FOR CROSSING POINT BEING BETWEEN END POINTS OF BOTH LINES.  EXACTLY
C TOUCHING THE END POINT COUNTS AS CROSSING.
C  X BETWEEN SOURCE AND RECEIVER:
      DO 3 K=1,2
```

```
      IF(DO(K)*DS(K).LE.0..AND.DELSQ(K).NE.0.)GO TO 4
 3    CONTINUE
      RETURN
 4    CONTINUE
C  X BETWEEN BARRIER END POINTS:
      DO 5 K=1,2
      IF(D1(K)*D2(K).LE.0..AND.DEL21(K).NE.0.)GO TO 6
 5    CONTINUE
      RETURN
 6    ICROSS=1
C NOW OBTAIN Z BY INTERPOLATING ALONG B1-B2.  NOTE THAT INTERPOLATION IS
C ALONG B(K), K BEING THE INDEX EMERGING FROM DO 5 LOOP AND IS A
C DIRECTION ALONG WHICH THE DIFFERENCES ARE NOT ZERO.
      X(3)=(B2(3)*D1(K)-B1(3)*D2(K))/DEL21(K)
      RETURN
      END
```

```
      SUBROUTINE PTBAR(XOBS,XSRC,XBAR,IFRQ,ATTEN)
C SUBROUTINE TO COMPUTE DELTA AND BARRIER ATTENUATION FROM SOURCE,
C OBSERVER, AND BARRIER LOCATIONS.  A SEPARATE FUNCTION SUBROUTINE
C 'DIFRAC' IS USED TO COMPUTE SHIELDING FORM A MAEKAWA-TYPE CURVE.
C
C INPUT VARIABLES:
C    IFRQ = INDEX IDENTIFYING SPECTRUM OF SOURCE
C    XBAR(3) = COORDINATES OF BARRIER TOP
C    XOBS(3) = COORDINATES OF RECEIVER
C    XSRC(3) = COORDINATES OF SOURCE
C OUTPUT VARIABLE:
C    ATTEN = BARRIER ATTENUATION FACTOR
C OTHER VARIABLES:
C    A,B,C = DISTANCES USED IN MAEKAWA'S GEOMETRY
C    DBO = X-DISPL. (Y- IF X-DISPL. IS ZERO) FROM BARRIER TO RECEIVER
C    DBS = X-DISPLACEMENT (OR Y- ) FROM BARRIER TO SOURCE
C    DELT = X-DISPLACEMENT (OR Y- ) FROM SOURCE TO RECEIVER
C    DELTA = PATH LENGTH DISTANCE, A+B-C
C    HEIGHT = HEIGHT OF SOURCE-RECEIVER LINE OF SIGHT AT BARRIER
C    I = DO LOOP INDEX
C
      DIMENSION XOBS(3),XSRC(3),XBAR(3)
C COMPUTE A,B,C
      A=SQRT((XOBS(1)-XBAR(1))**2+(XOBS(2)-XBAR(2))**2+(XOBS(3)-
     1XBAR(3))**2)
      B=SQRT((XSRC(1)-XBAR(1))**2+(XSRC(2)-XBAR(2))**2+(XSRC(3)-
     1XBAR(3))**2)
      C=SQRT((XSRC(1)-XOBS(1))**2+(XSRC(2)-XOBS(2))**2+(XSRC(3)-
     1XOBS(3))**2)
      DELTA = A+B-C
C TEST FOR BREAK IN LINE OF SIGHT
      DO 1 I=1,2
      DELT=XOBS(I)-XSRC(I)
      IF(DELT.NE.0.)GO TO 2
 1    CONTINUE
 2    CONTINUE
      DBS=XBAR(I)-XSRC(I)
      DBO=DELT-DBS
      HEIGHT=(XOBS(3)*DBS+XSRC(3)*DBO)/DELT
C ASSIGN SIGN OF BREAK TO DELTA
      DELTA = SIGN(DELTA,XBAR(3)-HEIGHT)
C GET ATTENUATION FROM FUNCTION ROUTINE
      ATTEN=DIFRAC(IFRQ,DELTA)
C CONVERT FROM DB TO FACTOR
      ATTEN=.1**(ATTEN*.1)
      RETURN
      END
```

```
      FUNCTION DIFRAC(IFREQ,DELTA)
C THIS FUNCTION SUBROUTINE COMPUTES BARRIER SHIELDING FROM MAEKAWA'S
C CURVE FOR A SINGLE FREQUENCY 'IFREQ' AND PATH LENGTH DIFFERENCE 'DELTA'.
C THIS IS A MINIMAL ROUTINE; A MORE GENERAL ONE WOULD HAVE DIFFRACTION
C CURVES BASED ON SHIELDING OF ACTUAL A-WEIGHTED SPECTRA, WITH IFREQ
C BEING AN IDENTIFYING INDEX OF THE SPECTRUM FOR A PARTICULAR SOURCE.
C THE FREQUENCY IS INPUT AS AN INTEGER 'IFREQ' RATHER THAN A REAL NUMBER
C SO AS TO FACILITATE ANY SUCH FUTURE CHANGE.
C
C INPUT VARIABLES:
C     DELTA = PATH LENGTH DIFFERENCE
C     IFREQ = FREQUENCY OF SOUND, HZ
C OUTPUT:
C     DIFRAC = SHIELDING, IN DECIBELS
C OTHER VARIABLES:
C     ATTEN(14) = TABLE OF SHIELDING VALUES
C     EN(14) = TABLE OF FRESNEL NUMBERS, CORRESPONDING TO ATTEN
C     FREN = FRESNEL NUMBER OF INPUT FREQUENCY AND DELTA
C     I = DO LOOP INDEX
C     VOV2 = SOUND SPEED DIVIDED BY TWO
C             (DATA ITEM - VALUE STORED BELOW IS IN FT/SEC, SO DELTA
C             MUST BE GIVEN IN FEET)
      DIMENSION ATTEN(14),EN(14)
      DATA ATTEN/0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,10.5,11.5,12.4,13./
      DATA EN/-.3,-.2,-.1,-.05,-.01,0.,.01,.05,.1,.2,.4,.6,.8,1./
      DATA VOV2/580./
C COMPUTE FRESNEL NUMBER
      FREN=DELTA*IFREQ/VOV2
C SET UP AND TEST FOR ZERO ATTENUATION CASE
      DIFRAC=0.
      IF(FREN.LE.-.3)RETURN
C TABLE LOOK-UP AND INTERPOLATION FOR FRESNEL NUMBER LESS THAN 1.
      DO 1 I=1,14
      IF(FREN.LE.EN(I))GO TO 2
 1    CONTINUE
C LOG FORMULA FOR FRESNEL NUMBER GREATER THAN 1.:
      DIFRAC=13.+ALOG10(FREN)
      RETURN
 2    CONTINUE
C INTERPOLATION FORMULA
      DIFRAC=((FREN-EN(I-1))/(EN(I)-EN(I-1)))*
     1(ATTEN(I)-ATTEN(I-1))+ATTEN(I-1)
      RETURN
      END
```

```
      SUBROUTINE LINSRC(D,EYE,EN,PHI,GNDA,EYEEQ,OBS,RLEN)
C THIS SUBROUTINE COMPUTES THE INTENSITY CONTRIBUTION FROM A LINE SOURCE
C WHOSE GEOMETRY IS DEFINED BY A DISTANCE D AND ANGLES PHI, AND WHOSE
C STRENGTH IS GOVERNED BY EN VEHICLES PER UNIT LENGTH, WITH AVERAGE
C EMISSION INTENSITY EYE.  POWER LAW EXCESS ATTENUATUION IS INCLUDED.
C
C INPUT VARIABLES:
C     D = NORMAL DISTANCE FROM LINE TO RECEIVER
C     EN = SOURCE DENSITY, NUMBER PER UNIT LENGTH
C     EYE = INTENSITY EMISSION LEVEL OF SOURCES
C     GNDA = GROUND ATTENUATION EXPONENT
C     OBS = X COORDINATE OF RECEIVER RE: LINE SEGMENT
C     PHI(2) = ANGLES RE:NORMAL  FROM RECEIVER TO LINE SEGMENT ENDS
C OUTPUT VARIABLE:
C     EYEEQ = INTENSITY CONTRIBUTION OF SEGMENT
C OTHER VARIABLES:
C     G = FINITE LINE LENGTH FACTOR
C     IRR = ERROR FLAG
C     SIGH1,SIGH2 = SIGNS OF ANGLES PHI
C     SI1,SI2 = SQUARES OF SINES OF PHI
C     TN1 = 1.+2*GNDA
C     X1,X2 = DISTANCES FROM RECEIVER TO END POINTS (USED IN COLINEAR CASE)
C
      DIMENSION PHI(2)
      COMMON/CONSTS/PI,TWOPI,PIOV2
      COMMON/UNITS/DO,D02
C TEST FOR OBSERVER IN LINE WITH SOURCE CASE
      IF(ABS(D).LT..1)GO TO 5
C
C TEST FOR VALID VALUES OF PHI AND D.  IF ERRORS ARE DETECTED, A
C MESSAGE IS PRINTED AND EXECUTION STOPS.
C CLEAR ERROR TRAP
      IRR=0
      IF(PHI(2).GT.PHI(1)) GO TO 1
      WRITE(7,10)
 10   FORMAT(' ERROR - PHI2 IS LESS THAN PHI1')
      IRR=1
 1    IF(-PI/2.LE.PHI(1).AND.-PI/2.LE.PHI(2)) GO TO 2
      WRITE(7,20)
 20   FORMAT(' ERROR - PHI1 OR PHI2 IS LESS THAN -PI/2')
      IRR=1
 2    IF(PI/2..GE.PHI(2).AND.PI/2.GE.PHI(1)) GO TO 3
      WRITE(7,30)
 30   FORMAT(' ERROR - PHI1 OR PHI2 IS GREATER THAN PI/2')
      IRR=1
 3    IF(D.GT.0.) GO TO 4
      WRITE(7,40)
 40   FORMAT(' ERROR - D IS LESS THAN OR EQUAL TO ZERO.')
      IRR=1
 4    IF(IRR.EQ.1) STOP
C END OF DATA CHECK SECTION
C
      SI1=SIN(PHI(1))**2.
      SI2=SIN(PHI(2))**2.
```

```
      SIGH1=SIGN(1.,PHI(1))
      SIGH2=SIGN(1.,PHI(2))
      A=GNDA+.5
      G=(BX(SI2,.5,A)*SIGH2-BX(SI1,.5,A)*SIGH1)/2.
C THE ABOVE STATEMENT CALLS/COMPUTES THE BETA FUNCTION.
      EYEEQ=EYE*DO*EN*(DO/D)**(2*A)*G
      RETURN
 5    CONTINUE
C OBSERVER IN LINE WITH SOURCE CASE
C TEST FOR OBSERVER BETWEEN END POINTS
      IF(OBS.LT.O..OR.OBS.GT.RLEN)GO TO 6
      WRITE(7,50)
 50   FORMAT(' ERROR - RECEIVER WILL GET RUN OVER')
 6    CONTINUE
C COMPUTE COLINEAR LINE SOURCE CONTRIBUTION
      X1=ABS(OBS)
      X2=ABS(RLEN-OBS)
      TN1=2.*GNDA+1.
      EYEEQ=ABS(EYE*DO*EN*((DO/X1)**TN1-(DO/X2)**TN1)/TN1)
      RETURN
      END
```

```
      FUNCTION BX(XIN,AIN,BIN)
C THIS FUNCTION SUBROUTINE COMPUTES THE INCOMPLETE BETA FUNCTION.  IT
C USES A POWER SERIES SOLUTION DIRECTLY FOR 0<X<.5, REVERSION RELATION
C TO EXTEND TO RANGE .5<X<1.  X=0 CASE IS TRIVIAL, AND X=1 CASE IS
C EXPRESSED AS GAMMA FUNCTIONS.
C PARAMETERS:
C    XIN = SUBSCRIPT VARIABLE
C    AIN = FIRST VARIABLE
C    BIN = SECOND VARIABLE
      DATA CRITER/.0001/
C BRING XIN WITHIN RANGE 0 TO 1.  THE SIN FUNCTION USED IN LINSRC
C TO PRODUCE XIN SOMETIMES GIVES A VALUE ONE OR TWO BITS OFF.
      IF(XIN.GT.1.)XIN=1.
      IF(XIN.LT.0.)XIN=0.
    5 X=XIN
C     SET FOR X<.5
      ALPHA=1.
      BETA=0.
      A=AIN
      B=BIN
      IF(X.LE..5) GO TO 1
C     X>.5 CASE
      ALPHA=-1.
      BETA=GX(A,IER)*GX(B,IER)/GX(A+B,IER)
      IF(IER.EQ.0) GO TO 6
      WRITE(5,102)IER
  102 FORMAT(' ERROR',I2,' IN GAMMA FUNCTION')
    6 CONTINUE
      X=1.-XIN
      A=BIN
      B=AIN
    1 CONTINUE
C     INITIALIZE SUM
      SUM=0.
C     TEST FOR X=0. CASE
      IF(X.EQ.0.) GO TO 4
C     COMPUTE BX FROM POWER SERIES
      APB=A+B
      AP1=A+1.
      A1=1.
      DO 3 J=0,100
      A1=(APB+J)/(AP1+J)*A1
      A2=A1*X**(J+1)
      SUM=SUM+A2
      IF(A2/SUM.LT.CRITER) GO TO 4
    3 CONTINUE
C     ERROR - DID NOT CONVERGE
      WRITE(5,103)
  103 FORMAT(' ERROR - SERIES DID NOT CONVERGE')
    4 BX=X**A*(1.-X)**B*(1.+SUM)/A
      BX=BETA+ALPHA*BX
      RETURN
      END
```

```
      FUNCTION GX(XX,IER)
C THIS FUNCTION COMPUTES THE GAMMA FUNCTION GX FOR A GIVEN ARGUMENT
C PARAMETERS:
C    XX = THE ARGUMENT FOR THE GAMMA FUNCTION
C    IER = RESULTANT ERROR CODE, WHERE
C          IER=0  NO ERROR
C          IER=1  XX IS WITHIN .000001 OF BEING A NEGATIVE INTEGER
C          IER=2  XX GT 34.5, OVERFLOW, GX SET TO 1.0E38
C    METHOD:
C      THE RECURSION RELATION AND POLYNOMIAL APPROXIMATION
C      BY C.HASTINGS,JR., 'APPROXIMATIONS FOR DIGITAL COMPUTERS',
C      PRINCETON UNIVERSITY PRESS, 1955
C
      IF(XX-34.5)6,6,4
    4 IER=2
      GX=1.E38
      RETURN
    6 X=XX
      ERR=1.0E-6
      IER=0
      GX=1.0
      IF(X-2.0)50,50,15
   10 IF(X-2.0)110,110,15
   15 X=X-1.0
      GX=GX*X
      GO TO 10
   50 IF(X-1.0)60,120,110
C
C         SEE IF X IS NEAR NEGATIVE INTEGER OR ZERO
C
   60 IF(X-ERR)62,62,80
   62 Y=FLOAT(INT(X))-X
      IF(ABS(Y)-ERR)130,130,64
   64 IF(1.0-Y-ERR)130,130,70
C
C         X NOT NEAR A NEGATIVE INTEGER OR ZERO
C
   70 IF(X-1.0)80,80,110
   80 GX=GX/X
      X=X+1.0
      GO TO 70
  110 Y=X-1.0
      GY=1.0+Y*(-0.5771017+Y*(+0.9858540+Y*(-0.8764218+Y*(+0.8328212+
     1Y*(-0.5684729+Y*(+0.2548205+Y*(-0.05149930))))))
      GX=GX*GY
  120 RETURN
  130 IER=1
      RETURN
      END
```

```
      SUBROUTINE LNWALL(OBSROD,XLNSRC,IFRQ,GNDA,EYESRC,EN,ROTCS,PHI,
     1                  RDLNTH,BARPTS,NBAR,NBPTS,EYE)
C PROGRAM TO COMPUTE THE LINE SOURCE CONTRIBUTION WITH BARRIERS.  ALL
C BARRIERS ARE CHECKED FOR THE FIRST ONE THAT BLOCKS LINE OF SIGHT FOR
C THE LINE SOURCE SEGMENT.  BARRIERS ARE CHECKED IN ORDER OF STORAGE,
C AND ONLY THE FIRST BLOCKING ONE IS CONSIDERED.
C
C INPUT VARIABLES:
C     BARPTS(3,5,3) = COORDINATES OF POINTS DEFINING BARRIERS
C     EN = NUMBER OF SOURCES PER UNIT LENGTH ON LINE SOURCE
C     EYESRC = INTENSITY EMISSION LEVEL OF SOURCES ON LINE
C     GNDA = GROUND ATTENUATION EXPONENT
C     IFRQ = IDENTIFIER OF SOURCE SPECTRUM
C     NBAR = NUMBER OF BARRIERS
C     NBPTS(3) = NUMBER OF POINTS DEFINING EACH BARRIER
C     OBSROD(3) = RECEIVER POSITION RE: TRANSFORMED LINE SEGMENT
C     PHI(2) = ANGLES TO END POINTS OF LINE SEGMENT
C     RDLNTH = LENGTH OF LINE SOURCE SEGMENT
C     ROTCS(2) = COSINE AND SINE OF LINE SEGMENT TRANSFORMATION ROTATION ANGLE
C     XLNSRC(3,2) = COORDINATES OF END POINTS OF LINE SEGMENT
C OUTPUT VARIABLE:
C     EYE = INTENSITY CONTRIBUTION OF LINE SEGMENT
C OTHER VARIABLES:
C     ATTEN = SHIELDING FACTOR FOR SHIELDED PART OF LINE
C     BROT(3) = COORDINATES OF BARRIER SEGMENT END POINTS RE: LINE
C     D = NORMAL DISTANCE FROM LINE TO RECEIVER
C     EYETEM(3) = INTENSITY CONTRIBUTIONS FROM THREELINE SECTIONS
C     GNDAB(3) = GNDA FOR UNSHIELDED SECTIONS, 0 FOR SHIELDED
C     IBAR = INDEX OF DO LOOP THROUGH BARRIERS
C     ICROSS = TEST FLAG FOR BLOCKAGE
C     IEND = PARAMETER DEFINING WHETHER BARRIER SECTION IS AN END SECTION;
C            SEE LNBLOK VARIABLE DICTIONARY FOR FULL DEFINITION
C     ILEFT, IRIGHT = USED IN SETTING UP IEND
C     ISHELD =  PARAMETER TO KEEP TRACK OF WHETHER SHIELDING HAS OCCURRED
C     JBAR = INDEX OF DO LOOP THROUGH BARRIERS
C     K = INDEX OF DO LOOP THROUGH THREE SECTIONS OF LINE SEGMENT
C     TEST = TEMPORARY VARIABLE USED IN COMPARING SIZES OF ANGLES
C     PHI3(2,3) = SETS OF ANGLES TO END POINTS OF THREE LINE SECTIONS
C     X(3) = DUMMY ARRAY USED IN CALL TO CROSS
C     XS(3) = POINT ON LINE POTENTIALLY SHIELDED
C     Z(2) = HEIGHTS OF ENDS OF LINE SOURCE SEGMENT
C
      DIMENSION OBSROD(3),XLNSRC(3,2),ROTCS(2),PHI(2),BARPTS(3,5,3)
      DIMENSION GNDAB(3),BROT(3,2),PHI3(2,3),EYETEM(3),NBPTS(3),Z(2)
      DIMENSION X(3),XS(3)
      GNDAB(1)=GNDA
      GNDAB(2)=0.
      GNDAB(3)=GNDA
      ISHELD=0
      EYE=0.
      D=ABS(OBSROD(2))
      DO 3 JBAR=1,NBAR
C JUMP OUT IF SHIELDING HAS OCCURRED - ONLY ONE BARRIER CONSIDERED:
      IF(ISHELD.EQ.1)RETURN
```

```
      DO 1 IBAR=1,NBPTS(JBAR)-1
C SET UP IEND AND PHI3
      ILEFT=0
      IF(IBAR.GT.1)ILEFT=1
      IRIGHT=0
      IF(IBAR.LT.NBPTS(JBAR)-1)IRIGHT=2
      IEND=ILEFT+IRIGHT
      PHI3(1,1)=PHI(1)
      PHI3(2,1)=PHI(2)
C TEST FOR LINE OF SIGHT BLOCKAGE.  'LNBLOK' RETURNS THREE SETS OF
C ANGLES DEFINING BLOCKED AND UNBLOCKED SEGMENTS, PLUS TRANSFORMED
C COORDINATES OF BARRIER SECTION.
      CALL LNBLOK(OBSROD,PHI3,XLNSRC,BARPTS(1,IBAR,JBAR),
     1IEND,ROTCS,BROT)
C IF NO BLOCKAGE IS FOUND (BASED ON ANGLE DIFFERENCE FOR BLOCKED CENTER
C SEGMENT), GO ON TO THE NEXT BARRIER SECTION.  IF THERE IS BLOCKAGE,
C COMPUTE NOISE CONTRIBUTIONS AND SET ISHELD=1.  IF THERE IS NO
C BLOCKAGE AND ISHELD IS SET, THEN THE SHIELDING PART OF THE BARRIER
C IS PAST: JUMP OUT.
      TEST=PHI3(2,2)-PHI3(1,2)
      IF(TEST.EQ.0..AND.ISHELD.EQ.1)RETURN
      IF(TEST.EQ.0.) GO TO 1
C ANGLES ARE O.K.  NOW MAKE SURE BARRIER IS BETWEEN SOURCE AND RECEIVER.
C CHECK ALONG PATH FOLLOWING PHI3(1,2).  IF PATHS DON'T CROSS,GO
C TO THE NEXT BARRIER.  ANY SUBSEQUENT SEGMENTS OF THIS ONE CHECKING
C OUT CAN HAPPEN ONLY BECAUSE OF BAD DATA.
      P=PHI3(1,2)
      XS(1)=OBSROD(1)+ABS(OBSROD(2))*SIN(P)/AMAX1(COS(P),.00001)
      XS(2)=0.
      CALL CROSS(OBSROD,XS,BROT(1,1),BROT(1,2),X,ICROSS)
      IF(ICROSS.EQ.0)GO TO 3
      ISHELD=1
C OBTAIN UNSHIELDED LINE SOURCE CONTRIBUTIONS FOR THE THREE SEGMENTS
      DO 2 K=1,3
      EYETEM(K)=0.
      IF(PHI3(2,K).GT.PHI3(1,K))CALL LINSRC(D,EYESRC,EN,
     1PHI3(1,K),GNDAB(K),EYETEM(K),OBSROD,RDLNTH)
 2    CONTINUE
C OBTAIN BARRIER SHIELDING FOR CENTER SECTION
      Z(1)=XLNSRC(3,1)
      Z(2)=XLNSRC(3,2)
      CALL LNBAR(D,PHI3(1,2),Z,RDLNTH,OBSROD,BROT,IFRQ,ATTEN)
C ADD COMPONENTS TO SOURCE SEGMENT TOTAL
      EYE=EYE+EYETEM(1)+EYETEM(3)+EYETEM(2)*ATTEN
 1    CONTINUE
 3    CONTINUE
C IF NO BLOCKAGE WAS FOUND, RETURN A VERY HIGH VALUE OF EYE.
C THE MAIN PROGRAM WILL REJECT THIS THROUGH THE LOGIC THAT REJECTS
C AMPLIFICATION OF BARRIER RE: NO BARRIER
      IF(ISHELD.EQ.0)EYE=1.E37
      RETURN
      END
```

```
      SUBROUTINE LNBLOK(OBSROD,PHI3,X1,BAR,IEND,ROTCS,BROT)
C SUBROUTINE TO DETERMINE WHETHER A BARRIER SECTION CAN BLOCK A
C LINE SOURCE SEGMENT.  THE ANGLES TO EACH END OF THE BARRIER ARE
C COMPUTED, AND COMPARED TO PHI.  THREE PAIRS OF ANGLES ARE RETURNED
C IN PHI3: LEFT UNSHIELDED END (BEGINNING AT PHI1), THE CENTER SHIELDED
C PART, AND THE RIGHT UNSHIELDED END (ENDING AT PHI2).  IF ANY OF THESE
C SEGMENTS DO NOT EXIST, THE PAIR OF ANGLES ARE SET EQUAL TO EACH
C OTHER.  ALL ANGLES ARE LIMITED BY THE INPUT VALUES OF PHI.
C      ONLY THE ANGLES ARE CREATED IN THIS PROGRAM.  THE FINAL TESTS,
C BOTH ON ANGLE AND INTERPOSITION, ARE MADE IN LNWALL.
C
C INPUT VARIABLES:
C    BAR(3,2) = END POINTS OF BARRIER SEGMENT
C    IEND = FLAG INDICATING WHETHER THIS IS A CENTER OR END SEGMENT
C             0: BOTH ENDS ARE END POINTS (ONE-SEGMENT BARRIER)
C             1: BAR(K,1) IS AN END POINT
C             2: BAR(K,2) IS AN END POINT
C             3: NIETHER END IS AN END POINT:THIS IS A MIDDLE SEGMENT
C    OBSROD(2) = RECEIVER LOCATION RE: LINE SEGMENT
C    PHI3(2,3) = END POINT ANGLES ON INPUT; ALSO AN OUTPUT.  SEE BELOW.
C    ROTCS(2) = COSINE AND SINE OF LINE SEGMENT TRANSFORMATION ANGLE
C    X1(2) = BEGINNING END POINT OF LINE SEGMENT
C OUTPUT VARIABLES:
C    BROT(3,2) = BARRIER SECTION END POINTS TRANSFORMED RE: LINE SOURCE
C    PHI(3,2) = ON OUTPUT, THREE PAIRS OF ANGLES DEFINING PARTS OF LINE
C OTHER VARIABLES:
C    ALPHA(2) = ANGLES FROM RECEIVER TO END POINTS OF BARRIER SEGMENT
C    DELTX,DELTY = X AND Y DIFFERENCES BETWEEN POSITIONS
C    I = DO LOOP INDEX IDENTIFYING END POINTS OF SEGMENTS
C    IFLIP = SIGNAL VARIABLE INDICATING WHETHER BARRIER ORIENTATION
C             IS  SAME OR OPPOSITE THAT OF LINE SOURCE
C
      DIMENSION PHI3(2,3),BAR(3,2),ROTCS(2),X1(2),OBSROD(2)
      DIMENSION ALPHA(2),BROT(3,2)
      COMMON /CONSTS/PI,TWOPI,PIOV2
C TRANSFORM BARRIER END POINTS TO ROTATED COORDINATES
      DO 1 I=1,2
      DELTX=BAR(1,I)-X1(1)
      DELTY=BAR(2,I)-X1(2)
      BROT(1,I)=DELTX*ROTCS(1)+DELTY*ROTCS(2)
      BROT(2,I)=DELTY*ROTCS(1)-DELTX*ROTCS(2)
      BROT(3,I)=BAR(3,I)
 1    CONTINUE
C COMPUTE ANGLES TO BARRIER ENDS
      DO 2 I=1,2
      DELTX=BROT(1,I)-OBSROD(1)
      DELTY=BROT(2,I)-OBSROD(2)
      IF(DELTY.EQ.0.)GO TO 3
      ALPHA(I)=ATAN(ABS(DELTX/DELTY))
      GO TO 4
 3    ALPHA(I)=PIOV2
 4    CONTINUE
C TEST FOR SECOND QUADRANT - BARRIER POINT FURTHER THAN RECEIVER FROM SOURCE,
C AND ON THE SAME SIDE
```

```
      IF((ABS(OBSROD(2)).LT.ABS(BROT(2,I))).AND.(OBSROD(2)*BROT(2,I))
     1.GT.0.)ALPHA(I)=PI-ALPHA(I)
C ASSIGN APPROPRIATE SIGN
      ALPHA(I)=SIGN(ALPHA(I),DELTX)
 2    CONTINUE
C STORE ALPHA IN MIDDLE OF PHI3, IN SIZE PLACE
      PHI3(1,2)=AMIN1(ALPHA(1),ALPHA(2))
      PHI3(2,2)=AMAX1(ALPHA(1),ALPHA(2))
C LIMIT BY ORIGINAL ANGLES
      PHI3(1,2)=AMAX1(PHI3(1,1),PHI3(1,2))
      PHI3(2,2)=AMIN1(PHI3(2,1),PHI3(2,2))
C FILL OUT MATRIX TO MAKE THREE PAIRS
      PHI3(2,3)=PHI3(2,1)
      PHI3(1,3)=PHI3(2,2)
      PHI3(2,1)=PHI3(1,2)
C DELETE UNSHIELDED ENDS IF BARRIER CONTINUES BEYOND THIS SECTION
      IF(IEND.EQ.0)RETURN
      IFLIP=0
      IF(ALPHA(1).GT.ALPHA(2))IFLIP=1
      IF((IEND-IFLIP).EQ.1.OR.IEND.EQ.3)PHI3(1,1)=PHI3(2,1)
      IF((IEND+IFLIP).EQ.2.OR.IEND.EQ.3)PHI3(2,3)=PHI3(1,3)
      RETURN
      END
```

```
      SUBROUTINE LNBAR(D,PHI,Z,RDLNTH,OBS,B,IFRQ,ATTEN)
C PROGRAM TO COMPUTE THE SHIELDING FACTOR ATTEN FOR A LINE SOURCE SHIELDED
C BY A BARRIER.  IT IS BASED ON KURZE AND ANDERSON'S INTEGRAL OF POINT
C SOURCE SHIELDING.  THE INTEGRATION MESH USED HERE CONSISTS INITIALLY
C OF THE END POINTS.  THE MESH IS SUCCESSIVELY DOUBLED (ADDING 1,2,
C 4,8,... POINTS.  THE END POINTS RECEIVE A WEIGHTING HALF THAT OF
C CENTER POINTS) UNTIL THE INTEGRAL CONVERGES TO WITHIN
C 10 PER CENT.  THE MESH POINTS ARE EQUALLY SPACED ON X.  THE INTEGRAND
C IS A*COS(PHI)**2, WHERE A IS THE POINT ATTENUATION.  THIS INTEGRAL
C IS NORMALIZED BY THE INTEGRAL OF COS(PHI)**2 FOR THE FINAL ATTENUATION.
C
C INPUT VARIABLES:
C     B(3,2) = COORDINATES OF BARRIER SEGMENT END POINTS
C     D = NORMAL DISTANCE FROM LINE SOURCE TO RECEIVER
C     IFRQ = SOURCE SPECTRUM IDENTIFIER
C     OBS(3) = RECEIVER COORDINATES RE: LINE SEGMENT
C     PHI(2) = ANGLES FROM RECEIVER TO ENDS OF SHIELDED PART OF LINE SOURCE
C     RDLNTH = LENGTH OF LINE SOURCE SEGMENT
C     Z(2) = HEIGHT OF END POINTS OF LINE SEGMENT
C OUTPUT VARIABLE:
C     ATTEN = LINE BARRIER SHIELDING FACTOR
C OTHER VARIABLES:
C     A = ATTENUATION FOR POINT SOURCE
C     ATT = ATTENUATION INTEGRAL IN CURRENT ITERATION
C     ATTOLD = ATTENUATION INTEGRAL FROM NEXT TO LAST ITERATION
C     A1,A2 = SHIELDING FROM END POINTS OF SHIELDED SECTION
C     CI = INTEGRAL OF COSINE - USED TO NORMALIZE ATT
C     CIOLD = CI FROM PREVIOUS ITERATION
C     COSPH2 = COS(PHI)**2, WHERE PHI IS ANGLE TO CURRENT X
C     DELX = DISTANCE FROM X1 TO POINT ON SOURCE
C     DX = DISTANCE BETWEEN POINTS ON SUBDIVIDED SOURCE
C     FK = FLOAT(K)
C     FM = FLOAT(MPTS)
C     ICROSS = SIGNAL VARIABLE FROM SUBROUTINE CROSS (NOT USED HERE)
C     K = INDEX OF DO LOOP THROUGH POINTS OF SUBDIVIDED SOURCE
C     M = INDEX OF DO LOOP SUCCESSIVELY DOUBLING NUMBER OF POINTS
C     MPTS = NUMBER OF ADDED POINTS M'TH TIME THROUGH M LOOP
C     SLOPZ = DZ/DX ON LINE SEGMENT
C     X(3) = POINT ON LINE
C     XC(3) = CROSSING POINT OF BARRIER AND LINE OF SIGHT
C     XLEN = LENGTH OF SHIELDED PART OF SOURCE
C     X1(3),X2(3) = END POINTS OF SHIELDED SECTION OF LINE SOURCE SEGMENT
      DIMENSION PHI(2),OBS(3),B(3,2),XC(3),X1(3),X2(3),Z(2),X(3)
C SET UP END POINT COORDINATES
      X1(1)=D*SIN(PHI(1))/AMAX1(COS(PHI(1)),.000001)+OBS(1)
      X2(1)=D*SIN(PHI(2))/AMAX1(COS(PHI(2)),.000001)+OBS(1)
      X1(2)=0.
      X2(2)=0.
      SLOPZ=(Z(2)-Z(1))/RDLNTH
      X1(3)=Z(1)+X1(1)*SLOPZ
      X2(3)=Z(1)+X2(1)*SLOPZ
C GET ATTENUATION FROM END POINTS
      CALL CROSS(OBS,X1,B(1,1),B(1,2),XC,ICROSS)
      CALL PTBAR(OBS,X1,XC,IFRQ,A1)
```

```
      CALL CROSS(OBS,X2,B(1,1),B(1,2),XC,ICROSS)
      CALL PTBAR(OBS,X2,XC,IFRQ,A2)
      ATTOLD=(A1*COS(PHI(1))**2+A2*COS(PHI(2))**2)*.5
      CIOLD=(COS(PHI(1))**2+COS(PHI(2))**2)/2.
      CI=CIOLD
      ATTOLD=ATTOLD/CIOLD
C SET UP PARAMETERS FOR INTEGRATION
      XLEN=X2(1)-X1(1)
      X(2)=0.
C SPLIT SOURCE LINE AND GET ATTENUATIONS
      DO 1 M=1,8
      MPTS=2**(M-1)
      DX=XLEN/MPTS
      ATT=0.
      DO 2 K=1,MPTS
      FK=K
      DELX=DX*(FK-.5)
      X(1)=DELX+X1(1)
      X(3)=X1(3)+DELX*SLOPZ
      CALL CROSS(OBS,X,B(1,1),B(1,2),XC,ICROSS)
      CALL PTBAR(OBS,X,XC,IFRQ,A)
      COSPH2=1./(1.+((X(1)-OBS(1))/D)**2)
      ATT=ATT+A*COSPH2
      CI=CI+COSPH2
 2    CONTINUE
C COMBINE WITH LAST
      ATTEN=(ATT+ATTOLD*CIOLD)/CI
C TEST FOR CONVERGENCE
      IF(ABS(1.-ATTEN/ATTOLD).LT..1)RETURN
      ATTOLD=ATTEN
      CIOLD=CI
 1    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE AREA(NCLPTS,CLPTS,WIDTH,OBSPTS,
     1OBSROT,NOBS,EYETOT,EYESTR,STRIPL,STRIPR,NSTR,CLLNTH)
      DIMENSION STRIPL(3,2,5,20),STRIPR(3,2,5,20),EYETOT(20),
     1EYESTR(20)
      COMMON /EDGE/ EDGEL(3,20), EDGER(3,20)
      DIMENSION ROTGER(3,2),ROTGEL(3,2)
      DIMENSION OBSPTS(3,10),DCL(10,20),OBSROT(3,10,20),CLLNTH(20)
      DIMENSION SLOPE(2),JSMALL(20),ISM(20)
      DIMENSION CLPTS(3,20),WIDTH(20),PHI(2),ROTCS(2,20)
      DIMENSION WSTRIP(20),ILG(20),NSTR(20), NSHRT(20)
C     OBTAIN EDGE POINTS
      CALL EDGES(NCLPTS,CLPTS,WIDTH)
      DO 1 I=1,NCLPTS-1
C     OBTAIN SIDELINE DISTANCES AND TRANSFORMATION
C     ANGLES RE: CENTERLINE SEGMENTS FOR ONE OBSERVER.
      CALL GEOM(CLPTS(1,I),CLPTS(1,I+1),OBSPTS,DCL(1,I),
     1PHI,ROTCS(1,I),OBSROT(1,1,I),CLLNTH(I),0)
C     SIDELINE DISTANCE
      DCL(1,I)=ABS(OBSROT(2,1,I))
C     TRANSFORM REMAINING OBSPTS RE: CENTERLINE
      DO 2 J=2,NOBS
      IF(NOBS.EQ.1) GO TO 2
      DX=OBSPTS(1,J)-CLPTS(1,I)
      DY=OBSPTS(2,J)-CLPTS(2,I)
      OBSROT(1,J,I)=DX*ROTCS(1,I)+DY*ROTCS(2,I)
      OBSROT(2,J,I)=DY*ROTCS(1,I)-DX*ROTCS(2,I)
      OBSROT(3,J,I)=OBSPTS(3,J)
C     SIDELINE DISTANCES
      DCL(J,I)=ABS(OBSROT(2,J,I))
    2 CONTINUE
C     FIND CLOSEST OBSERVER TO CL
      CALL SMALL(DCL(1,I),NOBS,JSMALL(I))
C     CALCULATE TRANSFORMED EDGES I AND I+1 ABOUT EACH
C     SEGMENT I
      DO 3 III=1,2
      II=III-1
      DXL=EDGEL(1,I+II)-CLPTS(1,I)
      DXR=EDGER(1,I+II)-CLPTS(1,I)
      DYL=EDGEL(2,I+II)-CLPTS(2,I)
      DYR=EDGER(2,I+II)-CLPTS(2,I)
      ROTGEL(1,III)=DXL*ROTCS(1,I)+DYL*ROTCS(2,I)
      ROTGEL(2,III)=DYL*ROTCS(1,I)-DXL*ROTCS(2,I)
      ROTGER(1,III)=DXR*ROTCS(1,I)+DYR*ROTCS(2,I)
    3 ROTGER(2,III)=DYR*ROTCS(1,I)-DXR*ROTCS(2,I)
C     ROTGEL(2)=-ROTGER(2)=NORMAL WIDTH OF SECTION
      ILG(I)=1
      IF(ROTGEL(2,1).LT.ROTGEL(2,2))ILG(I)=2
      ISM(I)=3-ILG(I)
C DETERMINE NUMBER OF STRIPS: RATIO BETWEEN NEAR AND FAR EDGE OF
C NEAREST STRIP SHOULD BE 1.1 (ROUNDED DOWN TO NEXT SMALLER NUMBER)
      WOV2=ROTGEL(2,ILG(I))
      DIST=DCL(JSMALL(I),I)
      NST=WOV2/(.1*(DIST-WOV2))
C CASE OF OBSERVER BETWEEN EDGES: WOV2>DIST
```

```
      IF(NST.LT.0)NST=5
C NO MORE THAN 5 STRIPS, DUE TO DIMENSION LIMITS
      NSTR(I)=MINO(NST,5)
      WSTRIP(I)=ROTGEL(2,ILG(I))/(NSTR(I)+.5)
C     HOW MANY STRIPS GO TO SHORT END
      NSHRT(I)=ROTGEL(2,ISM(I))/WSTRIP(I)
      NSHRT(I)=MINO(NSHRT(I),5)
C     SET UP TO COMPUTE FULL LENGTH STRIPS
      SLOPE(1)=ROTGEL(1,1)/ROTGEL(2,1)
      SLOPE(2)=(ROTGEL(1,2)-CLLNTH(I))/ROTGEL(2,2)
      DO 5 N=1,NSTR(I)
C     COMPUTE Y VALUES
      DO 6 K=1,2
      STRIPL(2,K,N,I)=N*WSTRIP(I)
    6 STRIPR(2,K,N,I)=-STRIPL(2,K,N,I)
C     X VALUES-FULL LENGTH STRIPS
      STRIPL(1,1,N,I)=STRIPL(2,1,N,I)*SLOPE(1)
      STRIPR(1,1,N,I)=-STRIPL(1,1,N,I)
      DELX=STRIPL(2,2,N,I)*SLOPE(2)
      STRIPL(1,2,N,I)=CLLNTH(I)+DELX
    5 STRIPR(1,2,N,I)=CLLNTH(I)-DELX
C     STRIPS FINISHED IF ALL ARE FULL LENGTH
      IF(NSHRT(I).EQ.NSTR(I)) GO TO 7
C     SET UP TO COMPUTE X FOR SHORT END
C     LIES ON LINE AY+B CONNECTING CORNERS
      BL=(ROTGEL(1,2)-ROTGEL(1,1))
     1/(ROTGEL(2,2)-ROTGEL(2,1))
      AL=ROTGEL(1,1)-BL*ROTGEL(2,1)
      BR=(ROTGER(1,2)-ROTGER(1,1))
     1/(ROTGER(2,2)-ROTGER(2,1))
      AR=ROTGER(1,1)-BR*ROTGER(2,1)
C     COMPUTE X FOR SHORT END
      DO 8 N=NSHRT(I)+1,NSTR(I)
      STRIPL(1,ISM(I),N,I)=AL+BL*N*WSTRIP(I)
      STRIPR(1,ISM(I),N,I)=AR-BR*N*WSTRIP(I)
 8    CONTINUE
    7 CONTINUE
C     COMPUTE TOTAL LENGTH OF STRIPS
      SUM=CLLNTH(I)
      DO 9 N=1,NSTR(I)
    9 SUM=SUM+STRIPL(1,2,N,I)-STRIPL(1,1,N,I)
     1+STRIPR(1,2,N,I)-STRIPR(1,1,N,I)
C     LINE SOURCE DENSITY RE: TOTAL
      EYESTR(I)=EYETOT(I)/SUM
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EDGES(NCLPTS,CLPTS,WIDTH)
C     COMPUTES LEFT AND RIGHT EDGE POINTS FOR AN
C     AREA DEFINED BY NCLPTS CENTERLINE POINTS AND WIDTHS.
      COMMON/EDGE/EDGEL(3,20),EDGER(3,20)
      DIMENSION CLPTS(3,20),THETA(19),WIDTH(20),PHI(20)
      COMMON/CONSTS/PI,TWOPI,PIOV2
C
C     COMPUTE SEGMENT ANGLES THETA.
C
      DO 1 I=1,NCLPTS-1
      DELTX=CLPTS(1,I+1)-CLPTS(1,I)
      DELTY=CLPTS(2,I+1)-CLPTS(2,I)
C     TEST FOR 90 DEGREE CASE.
      IF(DELTX.NE.0.) GO TO 2
      THETA(I)=PIOV2
      GO TO 3
    2 CONTINUE
C     CALCULATE PRINCIPLE VALUE OF ANGLE
      THETA(I)=ATAN(DELTY/DELTX)
 3    CONTINUE
C     THIS IS CORRECT FOR FIRST AND SECOND QUADRANTS.
C     TEST FOR SECOND OR THIRD QUADRANTS.
      IF(DELTX.LT.0.)THETA(I)=THETA(I)+PI
    1 CONTINUE
C
C     COMPUTE LATERAL ANGLES PHI.
C     END POINTS
      PHI(1)=THETA(1)+PIOV2
      PHI(NCLPTS)=THETA(NCLPTS-1)+PIOV2
C DONE IF ONLY ONE SEGMENT
      IF(NCLPTS.EQ.2)GO TO 6
C MIDDLE POINTS
      DO 4 I=2,NCLPTS-1
      PHI(I)=(THETA(I)+THETA(I-1)+PI)/2.
C     TEST FOR PROPER QUADRANT.
      ANGL1=AMOD(ABS(PHI(I)-THETA(I)),TWOPI)
      ANGLE2=AMOD(ABS(PHI(I)-THETA(I-1)),TWOPI)
      IF(ANGL1.LT.PI.AND.ANGL2.LT.PI) GO TO 4
C     ADJUST QUADRANT
      PHI(I)=PHI(I)+PI
 4    CONTINUE
 6    CONTINUE
C     COMPUTE LEFT AND RIGHT EDGES
      DO 5 I=1,NCLPTS
      DEL=WIDTH(I)/2.
      DELTAX=DEL*COS(PHI(I))
      DELTAY=DEL*SIN(PHI(I))
      EDGEL(1,I)=CLPTS(1,I)+DELTAX
      EDGER(1,I)=CLPTS(1,I)-DELTAX
      EDGEL(2,I)=CLPTS(2,I)+DELTAY
      EDGER(2,I)=CLPTS(2,I)-DELTAY
C     Z AT EDGES = Z ON CENTERLINE
      EDGEL(3,I)=CLPTS(3,I)
    5 EDGER(3,I)=CLPTS(3,I)
      RETURN
      END
```

```
      SUBROUTINE SMALL(A,N,J)
C THIS ROUTINE FINDS THE SMALLEST ABSOLUTE VALUE IN ARRAY A(N) AND
C SETS J EQUAL TO ITS INDEX
      DIMENSION A(1)
      J=1
      DO 1 I=2,N
      A1=ABS(A(J))
      IF (ABS(A(I)).LT.A1) J=I
    1 CONTINUE
      RETURN
      END
```