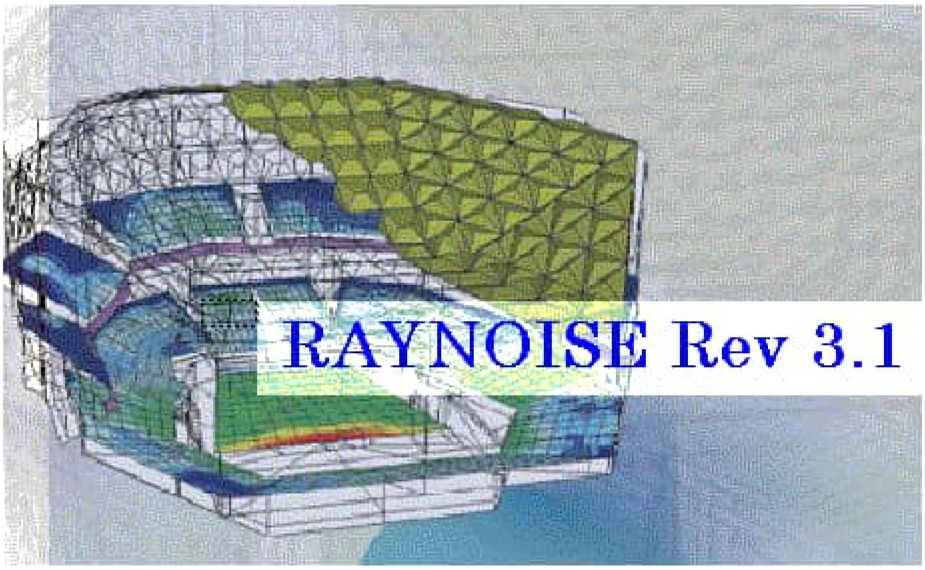


LMS INTERNATIONAL
Empowering Engineering Innovation

RAYNOISE Revision 3.1

Users Manual

RAYNOISE Revision 3.1 documentation

COPYRIGHT NOTICE:

REGISTERED TRADEMARKS:

PRELIMINARY NOTICE

LMS International aims to deliver software of high quality which conforms to the specifications agreed with licensees, but cannot guarantee that all features will perform exactly as described. Those who use the program do so on the understanding that the results from analysis software are dependent on the inputs. In any case, the program should be used in accordance with the instructions contained in this documentation and other information provided with the program or otherwise supplied by LMS International. LMS International does not warrant that any specific feature will produce specific results, and does not accept liability for any losses, direct, consequential or otherwise, resulting from the use of the program.

About LMS

LMS International (and previously Numerical Integration Technologies) is a leading developer of Computer–Aided Engineering software for Computational Vibro–Acoustics and Engineering Design Analysis. The main activities of the CAE Division of LMS International are product research, development and technical support, complemented by specialized engineering services and project consultancy with a wide range of advanced numerical analysis methods.

About RAYNOISE

RAYNOISE is a computer–aided acoustic modeling and analysis system for the architects, building, environmental, industrial and other acousticians and noise specialists. It uses advanced ray–tracing methods, to predict the sound field produced by multiple sources at any locations in closed, open or mixed (partially–open) spaces. RAYNOISE automatically handles complex interactions such as multiple reflections from different surfaces and the effects of coherent and incoherent sources. Special methods enable diffusion, diffraction and through–wall transmission to be modeled. The results can be shown in many different ways: color contour maps of sound pressure levels and other quality and speech–intelligibility metrics, 1/1 and 1/3–octave and narrow–band spectra, echograms, reverberation–time tables, and others. Results can be convolved with sound signals to produce stereo auralizations. The modeling and results display are carried out in an interactive graphical environment. Interfaces enable exchange of geometry and other data with other programs such as CAD systems.

See *What is RAYNOISE ?*, Chapter 1, for more details.

RAYNOISE is one of a range of acoustic, vibro–acoustic and design optimization programs, including LMS SYSNOISE, VIOLINS, SEADS, LMS OPTIMUS and the LMS Virtual.Lab product suite.

Table of Contents

List of Tables

List of Figures

1

What is RAYNOISE and what can it do?

This chapter describes the general features and principles of RAYNOISE, including:

- ○ *How RAYNOISE models acoustic phenomena*
- ○ *Typical applications*
- ○ *An overview of the manuals*

1.1 What is RAYNOISE ?

1.1.1 An outline description of RAYNOISE

RAYNOISE is an advanced computer program, designed to simulate the acoustical behavior of any arbitrary enclosed volume, any open space or a combination of both: half open/half closed.

RAYNOISE precisely models the physics of acoustical propagation, including specular and diffuse reflections against physical boundaries, wall absorption and air absorption, diffraction across screens, and transmission through walls.

Sound source directivity patterns can be entered in complete detail. Sources can be either coherent or incoherent.

The core of the model is a hybrid algorithm combining the Mirror Image Source Method (MISM) and the Ray Tracing Method (RTM). See *Geometrical Acoustics*, Section 3.1 on Page 3–3, for more information.

1.2 What does RAYNOISE do ?

1.2.1 Typical applications

While the application of RAYNOISE is mainly in industrial noise control, room acoustics and environmental acoustics, applications also include reverberation time assessment in churches, acoustical design of sound recording studios, speech–intelligibility assessment of public address systems in railway and subway stations, design of sound reinforcement systems, road and rail noise assessment from elevated track, structures and tunnel exits, noise control on offshore plants, audience noise inside and outside sports stadiums, acoustic layout of office environments, *etc*.

1.2.2 The modeling process

RAYNOISE starts with a geometry model, made in an exterior program (*eg* a CAD program such as AutoCAD, or a CAE program such as MSC/PATRAN). The geometry can be altered if required or entered directly into RAYNOISE. Material properties are assigned to polygon elements and as many sources and receivers are defined as the user requires.

Analysis parameters can be controlled by the user, including a choice of analysis options, level of detail of data storage, and refinement of the calculation (*eg*, addition of diffusion) generally with a trade–off between complexity, calculation time and accuracy.

1.2.3 The results

Results from calculations include statistical reverberation time, enclosed volume and mean free path. Color maps enable the visualization of acoustical quantities such as SPL, SPLA, RT60, STI, LE, and many others.

Detailed echogram/histograms at prescribed receiver points, enable a careful examination of the propagation of sound, in the time and frequency domains.

Finally, listening to the simulated sound and noise by convolving source material with the calculated binaural impulse response, enables auralization and subjective assessment, for a complete acoustical study.

1.3 About the manual

1.3.1 Structure of the manuals

This manual is structured in a manner which is intended to allow the user to follow it through in parallel with a typical modeling and analysis sequence. In general, this also follows the structure of the main menus. Some matters which are especially detailed, such as the formats of interfaces to other programs, are dealt with in Appendices. Some complete worked examples of typical applications are also included with the software installation, with command–language files which allow the examples to be re–created and re–run.

A large number of cross–references are supplied, together with Contents and Index pages.

1.3.2 Presentation of the manuals

The manuals are presented both in printed form and as on–line interactive text. In the printed form, explicit cross–references and page numbers are shown. In the on–line form, the same basic text is accessible, but the cross–references are shown in the form of ‘hot text’ or hyperlinks: by clicking on these with the mouse pointer, the user can go directly to the corresponding section where further, relevant information can be found.

The ’Home’ button to be found at the bottom of every page in the on–line format, enables an immediate return to the welcome and contents pages.

The documentation viewer also provides possibilities for browsing and searching the on–line manuals, and has its own help function.

2

Release notes: RAYNOISE Revisions 3.0 and 3.1

This chapter gives an overview of the new features of RAYNOISE Revisions 3.0 and 3.1. It is mainly intended for existing users, and highlights the changes from previous program versions, including:

- ○ *New graphical interface*
- ○ *New calculation features*
- ○ *New data handling*
- ○ *New results post−processing*
- ○ *Making the transition from Revision 2.1A*

2.1 Welcome to RAYNOISE Revision 3.1

RAYNOISE Rev 3.1provides a complete set of acoustic modeling tools for closed, open and partially–open geometries, using geometrical acoustics methods. The acoustic models can use geometry data from architectural CAD programs and geometric modelers and mesh generators. Results of the RAYNOISE calculations can be scanned, displayed and further post–processed within RAYNOISE and can be output in a range of formats for use in other programs.

In these Release Notes, the extensive changes and improvements implemented in Revision 3.0 are described, followed by a shorter description of the further changes made at Revision 3.1.

In developing RAYNOISE Revision 3.0, we at LMS took into account many requests and comments from users. We have also taken advantage of the extensive base of core technologies which exists within LMS, and we have used many of the features of other software products and in particular the SYSNOISE program. Those who are already familiar with SYSNOISE will find that RAYNOISE has much the same 'look and feel'. We believe that those for whom this is new, will find it a more natural user environment, which is easier to use.

RAYNOISE 3.1 is an update of Revision 3.0, with some features added or extended and the operation of other functions improved. We welcome comments from users, to help us to make further improvements in the next enhancement to the program.

2.2 Our Objectives

Before starting the development of RAYNOISE Rev 3.0, we had four major objectives in mind:

2.2.1 Ease–of–use

We wanted to make RAYNOISE even easier–to–use than before, by implementing many suggestions from users, by redefining the menus and the graphical user interface, and by using an integrated database structure for model data and results.

2.2.2 More analysis capabilities

We have implemented new calculation sequences like phase ray–tracing with coherent and incoherent sources, and have integrated the former Image and Mapping options into a single solution sequence, with flexible choices for parameters like numbers of rays and reflection order, where there is a trade–off between speed and accuracy. Calculation efficiency has been improved, with an enhanced echogram tail correction.

2.2.3 Improved results presentation and auralization

Presentation and interpretation of results is a key to the productive use of modeling software, so we have added many new capabilities for presenting and understanding the results, such as interactive picking of receiver points for the presentation of spectra. The auralization module adds new, binaural virtual–reality capabilities.

2.2.4 To build a platform for the future

The new multi–model architecture of RAYNOISE allows the user to control complex modeling procedures and comparative studies. The modular architecture of the program allows us to add new features in a well–structured manner, to respond to users' requests, and as new algorithms are proved.

2.3 What is new in RAYNOISE Rev 3.0?

2.3.1 Summary

- Graphic User Interface with a permanent display of the model and an extensive display manager
- On−line help
- Database files and multi−model data handling
- Automatic transfer of layers from DXF files into RAYNOISE sets
- Sabine coefficients, Diffusion coefficients and Transmission Loss values in octave bands or tables of frequency dependence
- Source power level in octave bands or tables of frequency dependence
- Ellipsoidal source directivity patterns, from the interpolation of horizontal and vertical polar diagrams
- Diffraction edges can be defined by mouse picking
- Generalized receiver surfaces (the 'field point mesh') and receiver geometry tools: points, lines, circles, planes, spheres, boxes, cylinders...
- Simultaneous calculation of spatial maps and echograms (the previous options 'Mapping' and 'Image' are combined)
- Coherent sources with phase ray−tracing, and the combination of coherent and incoherent sources (*eg* STI calculations of sound plus noise)
- Continuous echogram 'tail' correction allowing faster convergence with fewer rays and/or reflections
- Generalized frequency selection, *eg* 1/1 or 1/3−octave, narrow−band, ...
- Narrow−band analysis using echogram interpolation
- Flexible control of extensive calculation parameters, to balance speed and accuracy
- 'Combine' tools for processing data and response functions, including (I)FFT
- Binaural impulse responses, computed by the convolution of an echo−histogram with measured Head−Related Transfer Functions

- Auralization: convolution with anechoic sound samples, with output in WAV, AU or AIFF format.

2.3.2 New Features

2.3.2.1 Graphical User Interface

We have re–worked the RAYNOISE user interface to provide truly interactive graphical tools, and have implemented many suggestions from our users. A better–organized layout of the menus makes the commands easier to understand. We have designed the menus to focus more on the ‘modeling process’ rather than the underlying calculation methods.

2.3.2.2 Toolbar

The RAYNOISE Toolbar offers short–cuts to frequently–used commands. It contains colored buttons and ’tooltips’ – customizable strings which appear when the mouse pauses for a few seconds over a given button.

A very useful new feature is the RAYNOISE button window. This is a collection of user–programmable buttons. Users can assign to each of the buttons a sequence of commands to guide them through their standard calculation processes.

2.3.2.3 Enhanced graphics

We have incorporated several graphics enhancements. RAYNOISE now supports graphical animation of all results. Model rotation is much faster thanks to ‘feature line’ rotation. All mouse and keyboard operations can be recorded in a RAYNOISE trace file and can be replayed for demonstration purposes – or to help our support engineers repeat your manual operations.

2.3.2.4 Graphical selection of objects

It is now much easier to select items graphically and to apply the correct data to them. Box selection and a single point picking tool are complemented by a free–hand selector. Selected items are automatically pasted into the command dialog which is open at that time. This allows you to bypass the previous procedure of creating sets before you could apply data to the selection. By clicking on a screen item, such as a field point, RAYNOISE will directly display the calculated response function, such as a spectrum, in a separate window.

2.3.2.5 Audio output

RAYNOISE will create audio files (.AIFF, .AU, .WAV) which can be played through your computer speakers or headset, or can be processed in external signal–processing systems or sound quality analysis software such as LMS CADA–X Sound Quality Monitor.

2.3.2.6 Manipulating results

A powerful ‘Combine’ tool is included, which allows post–processing manipulations of frequency–domain results, and transformation with (Inverse) FFT and many other mathematical operators on complex results, tables or other data. This can allow the comparison of alternative models, derivation of insertion loss, *etc etc*.

2.3.3 New Program Structure

2.3.3.1 Integrated database

All data and results for each model are now stored in a model database to reduce file clutter and to give faster and easier access to all results. On Unix systems, the network license server can give users direct access to RAYNOISE from any workstation on their network, offering more flexibility to the user regarding the choice from where to run the program. The on–line help of RAYNOISE has also been improved. RAYNOISE Revision 3.0 now has context–sensitive hypertext–based help which is accessible from within any command dialog.

2.3.3.2 Multi–model architecture

The new RAYNOISE multi–model architecture allows users to operate with more than one model at a time and to present the models and results side–by–side or superimposed. Until now, RAYNOISE could only be used with a single acoustic model. With the new model manager, you can access several models and immediately compare their data and results, for instance to assess the influence of different design features or the sensitivity of the results to different analysis techniques, such as the inclusion of diffusion at surfaces.

However, this new multi–model architecture does not complicate the use of RAYNOISE: most existing command files can be re–used with very few changes.

2.3.3.3 Database implementation

For each model, RAYNOISE keeps the complete data and results information in a binary model database file, which significantly reduces the number of files stored on the user’s disk. Therefore, your disk will no longer be cluttered with directories of

*.MAP, *.RES, or other files. You can access the model database and import or export model information and results, to share with co−workers or support staff. RAYNOISE is also upwardly− and downwardly−compatible with other revisions for most data and results.

A model database structure also has several advantages for results post−processing:

- Selecting results is much easier since you don't have to remember all the filenames or their file extensions.
- Statistical reverberation time results are stored in the database and can be re−used when re−starting RAYNOISE.
- Because all mesh−based results (SPL, RTs, quality parameters, STI, ...) are available in the database, you can easily create any type of spectrum, other response functions, or animate the results.
- Graphic display is faster as results are read from a single database file.

2.3.3.4 Network license server

RAYNOISE Rev 3.0 gives you more flexibility through floating network licenses. Users can access RAYNOISE from any workstation in a UNIX network which runs the network license server program. Single and multi−user configurations are possible. The 120−character Authorization Code has been replaced by an 11−character checksum string and a readable license file. Contact your local distributor, agent or support representative for detailed installation information.

2.3.3.5 On–line help

The RAYNOISE User's Manual is now available on−line.

You can choose whether to copy the contents of the Documentation CD−ROM onto your local disk, or to read the Documentation files directly from the CD−ROM. The Help function is context−sensitive and contains hyperlinks to navigate to other related RAYNOISE topics and background information. All command dialogs have a Help button which jumps to the relevant section of the on−line manual.

Postscript versions of the manuals are included on the CD−ROM. These manuals may be printed by any registered user.

2.3.4 Compatibility with previous versions

The data structures for RAYNOISE Revision 3.0 are very different from those for Revision 2.1A, as regards both internal data−handling during execution and the permanent storage of models and results. As a consequence, there is no direct upwards compatibility of models from Revision 2.1A to Revision 3.0. However, various procedures have been implemented which allow both models and certain results to be transferred, as well as the re–use of command files in order to re–run calculations. The details are given in *Appendix 5: Converting data from Revision 2.1*. (see Chapter 13 on Page 13–1).

2.4 What is new in RAYNOISE Rev 3.1?

2.4.1 Summary

- Before importing a mesh geometry, the user can choose what happens if non–coplanar polygons are found: they can either be forced to become coplanar (as was standard in Rev 3.0) or they can be split into triangles. See *Import and Export of a mesh*, Section 5.3.5 on Page 5–11.

- The graphics rendering has been improved, to eliminate some problem effects with the hidden–line algorithm. The earlier algorithm is still available (and is usually faster, which may be significant for large models). The new algorithm is selected by an option in the Display, Graphical Options dialogs. The improved hidden–line view may be significant when picking objects, especially edges for diffraction. See *Painter algorithm*, Section 12.1.3.6 on Page 12–5.

- Acoustic frequency functions can be weighted using the A, B, C or D weightings, implemented according to the ANSI standards. See *Options for response functions*, Section 10.3.3 on Page 10–30.

- The display of results from Statistical Reverberation Time computations includes a Preferred Reflection Order, which indicates the calculation parameter which should be used in order to get accurate impulse response results. See *Extracting reverberation time results*, Section 9.2.3 on Page 9–4.

- The normalization factor applied to the output of an auralization process (when writing a .WAV or other file) is displayed. A user–controlled normalization can be applied, via the Environment Variable NormIRF. See *Convolution*, Section 9.8 on Page 9–24.

- Databases can be Merged using the File, Database command. See *Merge*, Section 5.2.4.2 on Page 5–7.

- RAYNOISE Rev 3.1 licenses are managed by FlexLM. This means little change for use on a UNIX network. On PC Windows, it means a 'floating' (network) license is possible, without a hardware security key; or the hardware–key–based approach can be retained. In both cases, the format of the license code file has changed from the previous Revision: contact your local distributor, agent or support representative for detailed installation information and to update your license.

2.4.2 Compatibility with previous versions

- The data structures for RAYNOISE Revision 3.1 are identical to those for Revision 3.0 and database files used in the previous version are compatible. If a Revision 3.0 database file is opened but the contents are not read, it may be necessary to change the Environment Variable RECORDLENGTH (see *RECORDLENGTH*, Section 11.8.3.58 on Page 11–32 and if necessary consult your local support representative). This should not affect databases that are made New within Revision 3.1.

- Command files used in Rev 3.0 are fully–compatible with Rev 3.1. However, it can be necessary (or desirable) to add new commands or options to command–files, where certain dialogs have been extended (for example, the addition of new Environment Variables).

- The requirements if converting directly from Revision 2.1A to Revision 3.1 are the same as what is described when moving from Revision 2.1A to Revision 3.0. See *Converting data from Revision 2.1*, Section 13 on Page 13–1.

3

Modeling principles and theoretical basis

This chapter describes the general features and principles of RAYNOISE, theory and calculation parameters, including:

3.1 Geometrical acoustics

3.1.1 General principles

RAYNOISE is based on the principles of geometrical acoustics and is therefore subject to its restrictions. Geometrical acoustics assumes that sound waves behave as sound rays, exactly as in geometrical optics light waves behave like light rays. Acoustic rays are reflected by solid surfaces and lose part of their energy at each rebound. This approach is only valid at medium−to−high frequencies as it partly neglects the wave aspect of sound behavior.

The Mirror Image Source Method (MISM) and Ray Tracing Method (RTM) are two well−known computer algorithms, which have been applied for several decades. They are briefly described below.

3.1.2 The Mirror Image Source Method (MISM)

The MISM uses virtual mirror image sources, in order to trace back sound reflection paths from the receiver to the sound source.

This can be easily illustrated with a simple 2D problem of a rectangular box, containing a spherical source at point *S* and a receiver at point *R* (see Figure 3.1−1). We start by constructing the first−order mirror images of point S with respect to all walls: S_1, S_2, S_3, S_4. By calculating the intersection points of the lines S_iR with the corresponding walls *i*, one can readily project the first−order reflection paths.

In the same way we proceed with the secondary sources, eliminating the wall at which the secondary source was mirrored last.

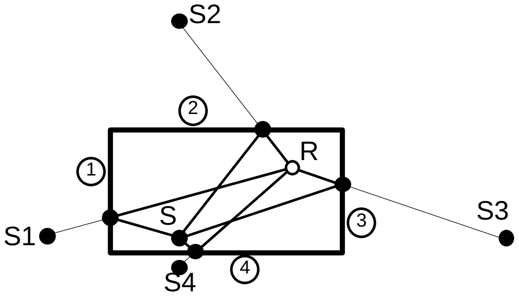


Figure 3.1–1 First–order reflection paths in rectangular box

This process has to be continued up to a prescribed order of image sources. Figure 3.1−2 shows the construction of an image source of third order (S_{124}). This image source thus represents a reflection path that has suffered three wall reflections (wall 1, wall 2 and wall 4).

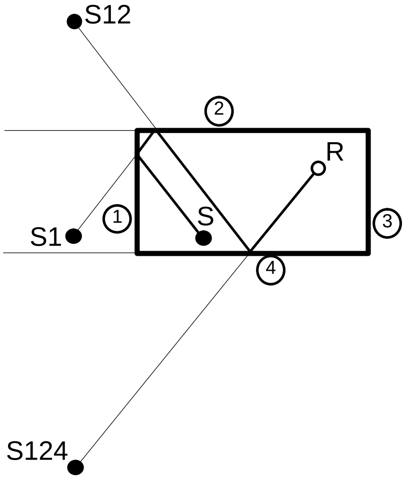


Figure 3.1–2 Third–order mirror image and corresponding reflection path

When applied to a rectangular box, a regular lattice of mirror sources results, of which every element is visible from every position in the room (see Figure 3.1−3).

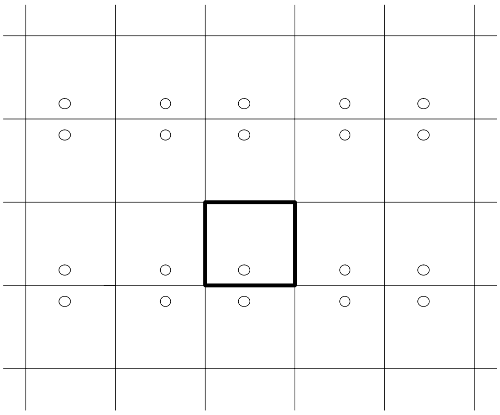


Figure 3.1–3 Constructing Mirror Image Sources

In irregular rooms, however, this is not the case and visibility tests have to be made. The example in Figure 3.1−4 shows that receiver R_1 can be reached by a first−order reflection against wall 1, while receiver R_2 cannot. In other words, R_1 is ’visible’ from S_1, R_2 is not. This means that for all calculated intersection points, one has to check if they are situated within the real physical boundaries.

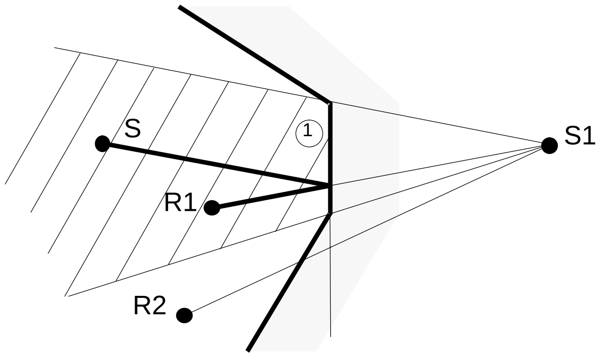


Figure 3.1–4 Visibility test

Due to these visibility tests, the MISM suffers from dramatically long computation times, especially when a large number of surfaces is involved and/or when the mean absorption is small, thus increasing the number of image sources that need to be taken into account.

3.1.3 The Ray Tracing Method (RTM)

In the Ray Tracing Method (RTM), one assumes that the energy emitted by the sound source is distributed into a discrete number of sound rays. Each ray has an initial energy equal to the total energy of the source divided by the number of rays. Each one travels at the speed of sound and collides with the walls, floor and ceiling, *etc*, where it is reflected in accordance with the law of specular reflection. The energy level of each ray decreases at reflections by means of wall absorption and progressively as it travels by means of air absorption. When the energy level falls below a user−defined threshold, the ray is abandoned and the next one is traced. (Figure 3.1−5)

In order to calculate the sound energy at different points in a room, receiver cells with finite volumes are defined. Each ray is checked to see whether it crosses the receiver volume. The number of rays crossing a receiver volume and the energy contributions of those rays give a measure of the sound pressure level. Losses due to spherical divergence are included as a result of the increasing separation between the rays as they spread out from the source with increasing travel time.

When echograms are needed, one proceeds as follows. The energy contributions of the various rays to a certain receiver cell are added within prescribed time intervals, resulting in a histogram. However, because of the time−averaging effect and the strongly random character of the ray arrivals, the histogram will only be an approximation of the true echogram.

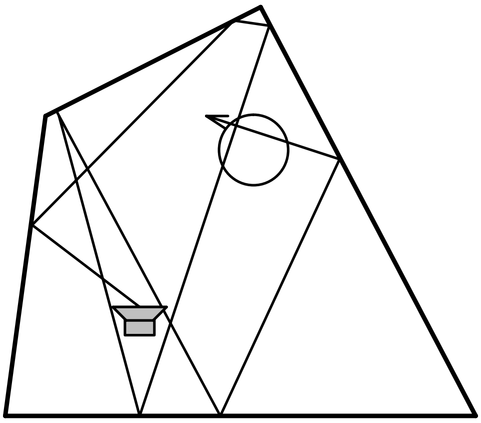


Figure 3.1–5 Sound Rays

3.1.4 Hybrid methods

3.1.4.1 The Conical Beam Method

The Conical Beam Method (CBM) emits a large number of cones with their vertices at the source. The propagation of the cones through the room is handled by applying a ray tracing algorithm to the axes of the cones (see Figure 3.1–6). Receivers are points, not volumes as in the RTM. When a receiver point lies inside a truncated cone, between two successive reflections, a visible image source has been found. Its contribution is easily calculated, using spherical divergence in the cone.

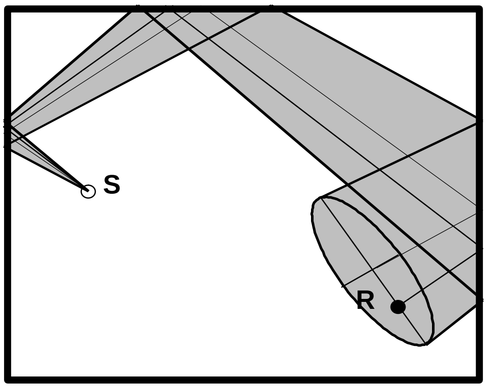


Figure 3.1–6 Cone propagation through physical boundaries

Figure 3.1–7 illustrates how the cone tracing method would find for the example in Figure 3.1–4 the contribution of image source S_1 to R. It is readily seen that, in contrast with the MISM, no visibility tests are needed. Only visible image sources will be detected.

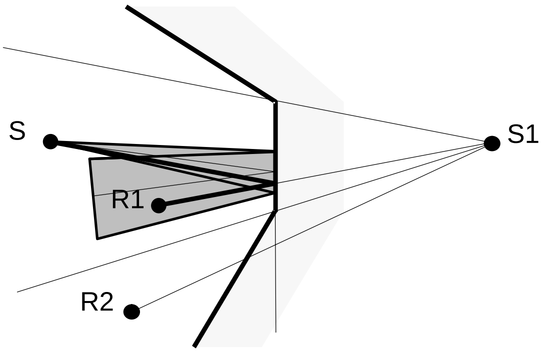


Figure 3.1–7 Equivalence of MISM and cone tracing

Still, two problems need special attention:

PROBLEM 1: The example of Figure 3.1–6 is a 2D case. In 3D things get more complicated. How can cones with circular cross–sections reconstruct the original spherical wave front?

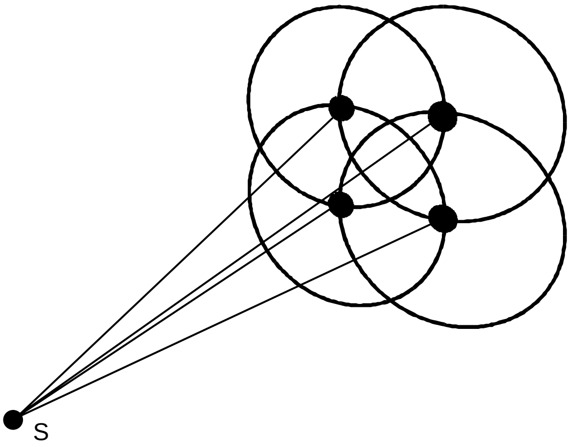


Figure 3.1–8 Overlapping cones

SOLUTION: In order to avoid gaps between the cones, it is necessary to proceed with overlapping cones (Figure 3.1–8). But this means that an image source can be detected two, three or four times. By keeping track of the path histories of each arrival at the receptor point, and reducing identical image sources to only one contribution, this problem can be resolved. A more elegant solution is to apply a weighting function on the cross–section of each cone, in such a way that the superposition of the cones reconstructs the original spherical wave source (see Figure 3.1–9). This has the advantage of not requiring additional data storage.

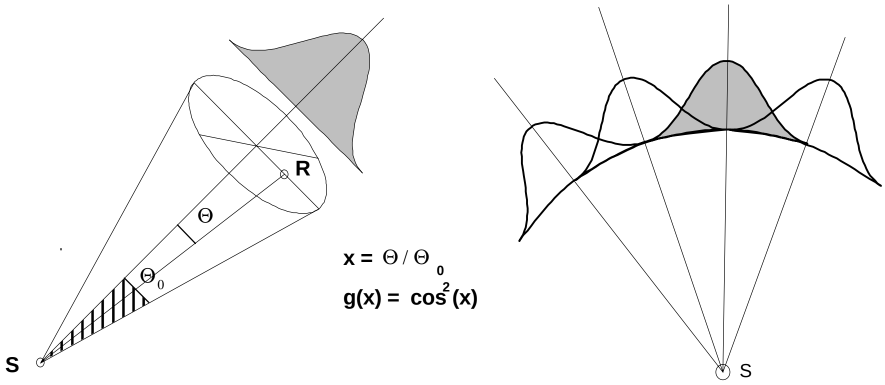


Figure 3.1–9 Using a weighting function

PROBLEM 2: As the cone propagates in space with increasing time, the chance of hitting a corner increases. When this occurs, an effect appears that can be described as ‘cone narrowing’ (Figure 3.1−10): some of the so−called visible image sources will be associated with a wrong reflection path and can therefore not be accounted for. This implies that some of the sound energy will be arithmetically absorbed.

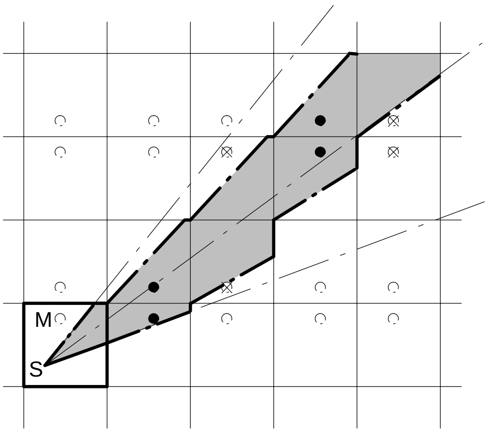


Figure 3.1–10 Cone narrowing

SOLUTION: This anomaly will always occur. It can be reduced by applying a smaller cone solid angle, but this increases the computation time. However, if we relax the rigid idea of using the MISM, we can accept these ’false’ image sources on the basis that in that case no arithmetic absorption of energy takes place. In a way, the ’false’ image sources compensate for ones missed. The cone tracing method thus trades some of its deterministic nature, derived from the MISM, for some statistical character, from the RTM.

3.1.4.2 The Triangular Beam Method (TBM)

The Triangular Beam Method is very similar to the CBM, but in stead of emitting cones, triangular−based pyramids are used to discretize the spherical wave front. This has the advantage that no ‘overlapping’ takes place, and thus no weighting functions have to be applied (see Figure 3.1−11).

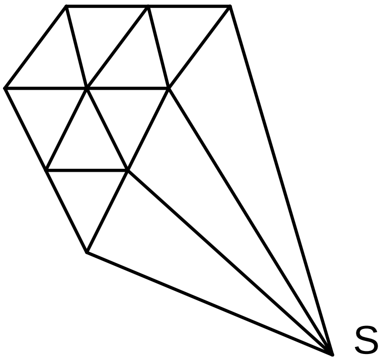


Figure 3.1–11 The Triangular Beam Method

The TBM yields more accurate results than the CBM, but its convergence rate is somewhat lower. It is recommended for exterior applications, where most reflections will have a low order and also the maximum order set for the calculation will usually be given a low value.

3.1.4.3 Conclusion

Due to the inaccurate acoustical behavior of the RTM, the MISM is preferred, but its calculation time increases exponentially with the mirror order, while increasing only linearly for the RTM. The Conical Beam Method (CBM) and the Triangular Beam Method (TBM) are MISM/RTM mixtures, combining the advantages of both methods in one algorithm. RAYNOISE is based on these methods in order to simulate the acoustical behavior of any arbitrary enclosed or open space.

- *Calculation parameters*, Section 9.5 on Page 9–13, details the different control parameters which can be applied, which influence calculation speed and accuracy as well as the particular calculation procedure and saving of results.

3.1.5 Approximate beam–tracing *vs* full beam–tracing

As explained above, in Section 3.1.4, Beam Tracing methods, this method emits a large number of cones (or pyramids) with their vertices at the source. The propagation of the cones through the room is handled by applying a ray–tracing algorithm to the axes of the cones. This means that, in order for a receiver to receive a contribution from the cone, it has to comply with two 2 conditions:

- The receiver must lie inside the geometrical cone;
- The receiver must lie in front of the geometrical plane that contains the surface with which the cone axis has collided.

This is an approximation, since only the axis is traced and not the entire cone face. For most interior simulations this is acceptable. However, for half–open/half–closed situations, or multi–domain cases (transmission from one zone to another) this approximation can give rise to important modeling errors: see Figures 3.1–12 and 3.1–13.

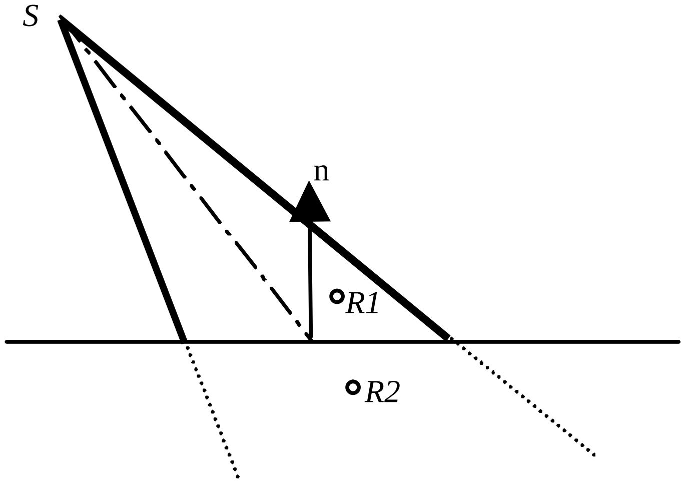


Figure 3.1–12 Receiver R1 gets a contribution, Receiver R2 not

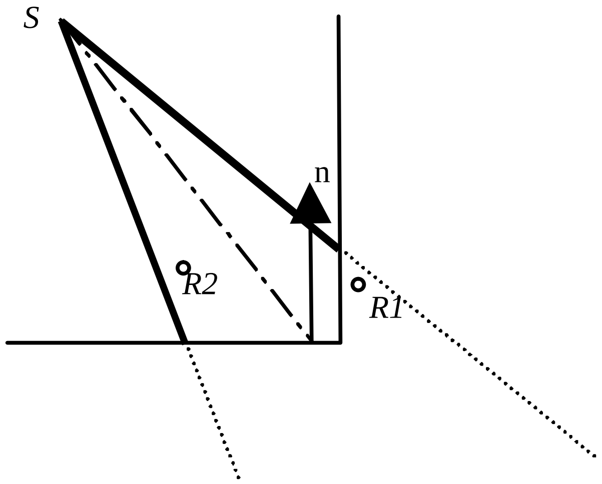


Figure 3.1–13 Receiver R1 gets an erroneous contribution, Receiver R2 gets a correct contribution

In order to avoid errors of this kind, one can optionally tell the program to verify whether or not the line from source to receiver is intersected by other polygons. This, however, gives rise to longer calculation times.

- The Environment Variable 'HIGHACCURACY' (see Section 11.8.3.35 on Page 11–27 for details) will determine whether this additional test is applied.

In the case of multi–domain geometries there is another solution to this problem. In this case, any receiver belongs to just one fully–closed field, surrounded by physical walls. Hence, the user can limit the immission of each receiver to just those sources within the relevant closed field (including the sources representing transmission through the walls from other fields). This allows correct results to be calculated, without having to use the HIGHACCURACY option.

The field points (receivers) are put into sets, according to their position in the geometry, and their source immission is specified.

- The source property '*Immission zone*' (see Section 7.4.3.6 on Page 7–21 for details) sets up the use of this technique.

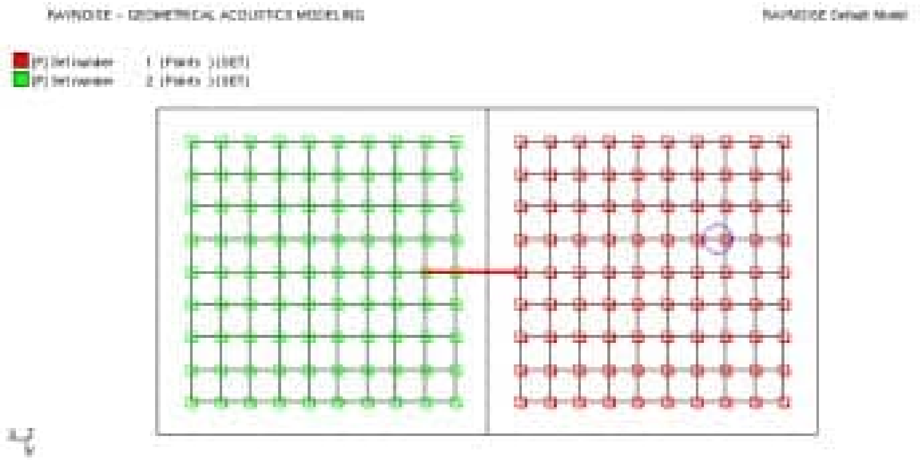

RAYNOISE - GEOMETRICAL ACOUSTICS MODELING
RAYNOISE Default Model
[P] SetNumber 1 (Points)(SET)
[P] SetNumber 2 (Points)(SET)

Figure 3.1–14 Source–receiver immission control: receivers in the left–hand domain only receive immissions from the transmission source, those in the right–hand domain only from the point source.

3.1.6 Tail correction

3.1.6.1 Statistical tail correction

When the cones (or pyramids) reach their abandonment criterion, there still may be some considerable energy left over in these cones (or pyramids). For reasons of energy conservation, one has to make sure that the abandonment criterion is chosen such that this left over energy is negligibly small. In cases where long reverberation times exist, this comes down to very long ray propagation paths, yielding high computation times. In order to reduce these, the user can decide to account for the late part of the echograms in the receivers, by means of statistical acoustics (see Figure 3.1–15).

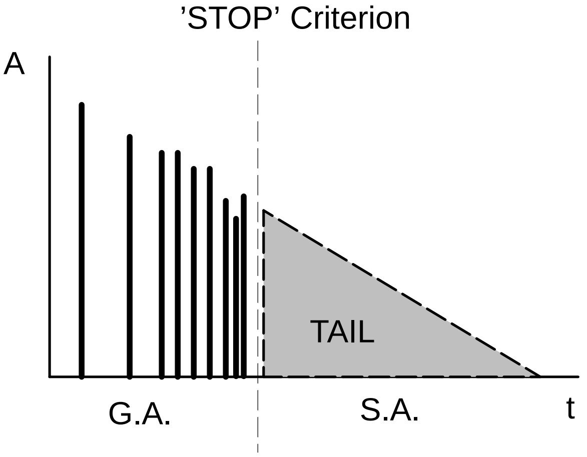


Figure 3.1–15 Transition from Geometrical Acoustics to Statistical Acoustics

The residual energy is distributed across all receivers, using the statistical estimate of the sound pressure level of a diffuse sound field in a closed room:

$$L_p = L_W + 10\log_{10}\left(\frac{4}{R}\right) \qquad \textit{Eqn 3.1–13.1–2}$$

with: L_p = Sound Pressure Level (dB)

L_W = Sound Power Level (dB)

R = Room Constant:

$$R = \frac{S\overline{\alpha}}{(1-\overline{\alpha})} \qquad \textit{Eqn 3.1–3}$$

This marriage between Geometrical Acoustics (GA) and Statistical Acoustics (SA) results in much lower computation times and higher convergence rates (see Figure 3.1–16).

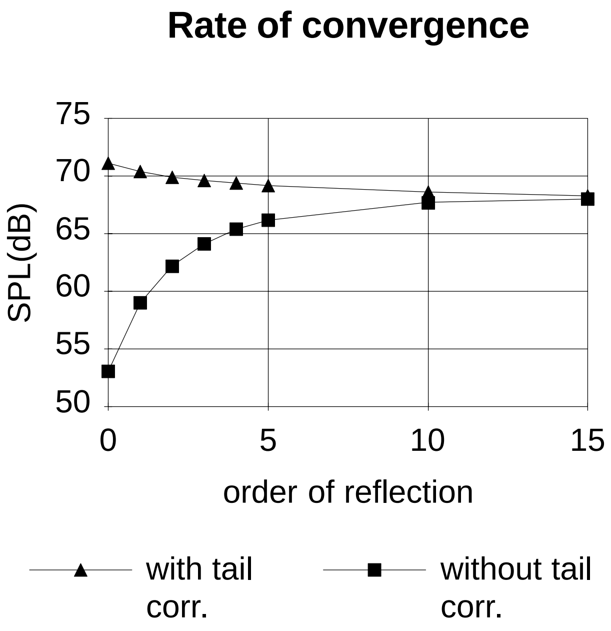


Figure 3.1–16 Faster convergence with tail correction

This means that if the reflection order is set to 0, one obtains SPL results that are calculated according to statistical acoustics.

Note! The use of echogram tail correction is available in all options. Statistical tail correction is set by giving the value 2 to the environment variable TAIL in the *Set–up file* RAYNOISE.STP (see Section 11.8.4 on Page 11–39) or the *Profile file* RAYNOISE.PRF (see Section 11.8.5 on Page 11–40).
Use the Tools, Environment Var. dialog (see *Changing environment variables*, Section 11.8.2 on Page 11–19) or the command ENVIRONMENT, if you want to change it during a RAYNOISE session:

```
RAYNOISE> ENVIRONMENT TAIL 2 RETURN
```

DO NOT use tail correction of any sort in exterior models: it has no physical meaning and will probably give significant errors in the results!

3.1.6.2 Continuous tail correction

The section on *Hybrid methods* (Section 3.1.4 on Page 3–6)proposes that 'false' image sources can compensate for the ones missed due to the cone–narrowing effect. This is only partly true. When the cross–section of the base of the cone or pyramid becomes larger than the size of the room, there is a systematic underestimation of detected image sources.

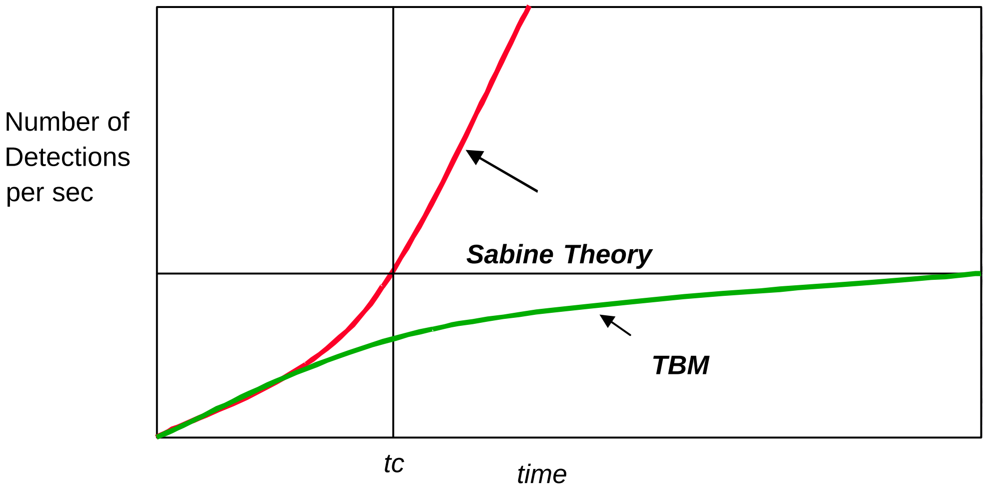


Figure 3.1–17 Comparison of wavefront arrival rates: beam tracing vs theory

According to Sabine acoustics, the number of wavefronts impinging on a receiver in one unit of time increases with the square of the time elapsed from the sound emission:

$$n(\tau) = \frac{4.\pi.c^3.\tau^2}{V}$$

Eqn 3.1–4

While for a beam trace, the number of detections per second tends to a constant value :

$$n(\tau \rightarrow \infty) = \frac{c.N}{l_{mfp}}$$

Eqn 3.1–5

where N is the total number of beams traced and l_{mfp} the mean free path length.

The critical time t_c is the time at which the theoretical parabola of equation 3.1–4 intersects the constant value of equation 3.1–5:

$$t_c = \frac{l_{mfp}}{c} \times \sqrt{\frac{N}{4\beta}}$$

Eqn 3.1–6

where β depends on the width of the beams.

The effective 'smoothed' detection rate for the beam tracing can thus be expressed as:

$$n(\tau) = \frac{4.\pi.c^3.\tau^2}{V}\left[1 - e^{-\frac{t_c^2}{\tau^2}}\right]$$

Eqn 3.1–7

Every detection in the beam tracing is then multiplied by a time–varying correction factor, which is the ratio between equation (1) and equation (4). This compensates in a smooth, continuous, way for the increasing loss of detections with time.

The advantage of this continuous tail correction is that faster convergence can be obtained with less rays and that the transient behavior is modeled correctly as well. However, it is still necessary to choose a reflection order high enough to describe the echogram with reflections that reach the late part of its transient region. Figures 3.1–18 and 3.1–19 show this: the main differences are apparent in the alter part of the histogram.

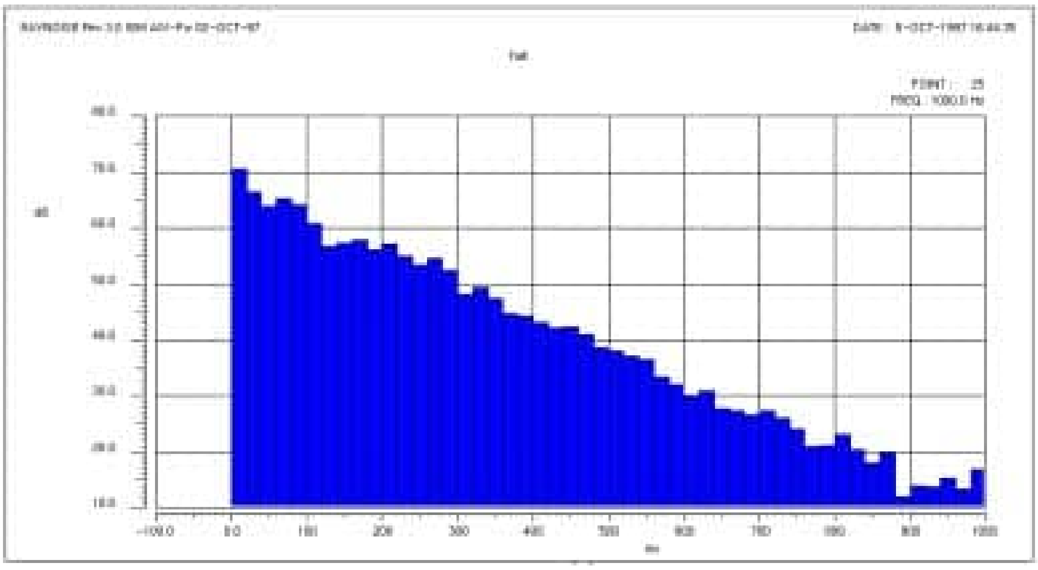


Figure 3.1–18 Calculated histogram without continuous tail correction.

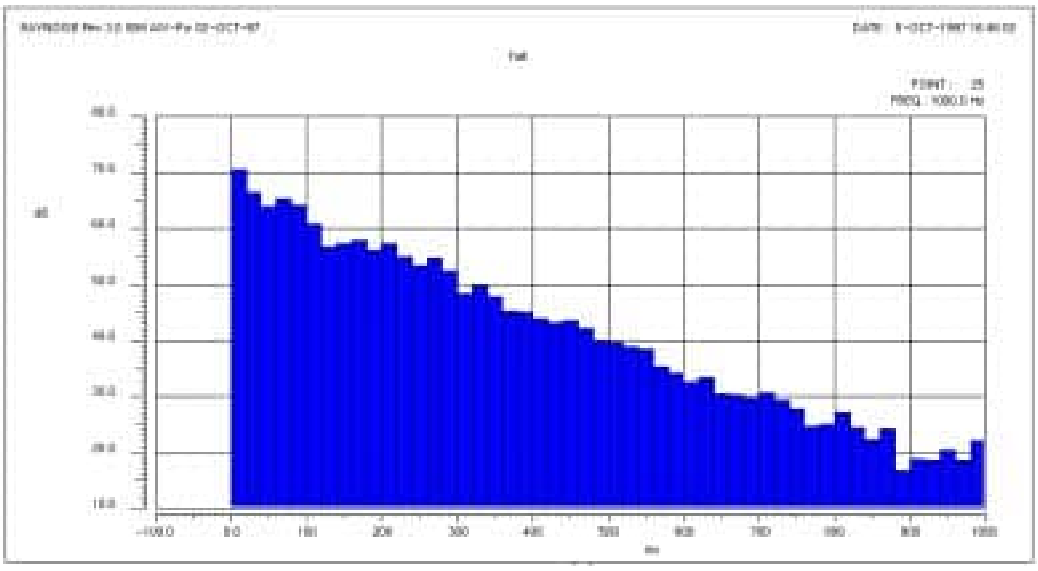


Figure 3.1–19 Calculated histogram with continuous tail correction.

Note! The use of echogram tail correction is available in all options. Continuous tail correction is set by giving the value 1 to the environment variable TAIL in the *Set–up file* RAYNOISE.STP (see Section 11.8.4 on Page 11–39) or the *Profile file* RAYNOISE.PRF (see Section 11.8.5 on Page 11–40).
Use the Tools, Environment Var. dialog (see *Changing environment variables*, Section 11.8.2 on Page 11–19) or the command ENVIRONMENT, if you want to change it during a RAYNOISE session:

```
RAYNOISE> ENVIRONMENT TAIL 1 RETURN
```

DO NOT use tail correction of any sort in exterior models: it has no physical meaning and will probably give significant errors in the results!

3.1.6.3 References on tail correction methods

Farina A., Proceedings Euronoise'95, p 55

Van Maercke D., Martin J., *Applied Acoustics* Vol. 38, p93 (1993)

3.2 Statistical reverberation time

3.2.1 Introduction

Before beginning detailed studies of the spatial distributions of acoustic parameters such as definition, clarity, *etc*, it is often desirable to have a good assessment of the overall (position–independent) reverberation time (RT) of the room.

A position–independent reverberation time can be calculated using formulae from statistical acoustics. These are usually based on the mean free path and its relative variance (standard deviation). Examples of such formulae are given below:

$$RT_{SABINE} = \frac{13.8\bar{l}}{c\bar{\alpha}}$$

Eqn 3.2–1

$$RT_{EYRING} = \frac{13.8\bar{l}}{-\,c\ln(1-\bar{\alpha})}$$

Eqn 3.2–2

$$RT_{KUTTRUFF} = \frac{13.8\bar{l}}{-\,c\ln(1-\bar{\alpha})(1+\frac{\gamma^2}{2}\ln(1-\bar{\alpha}))}$$

Eqn 3.2–3

where:	$\bar{l}$	=	mean free path (m)
	RT	=	Reverberation Time (s)
	c	=	sound velocity (m/s)
	$\bar{\alpha}$	=	mean absorption coefficient (–)
	γ	=	relative variance
	13.8	=	an empirical constant (in MKS units)

3.2.2 Procedure to derive statistical reverberation time

RAYNOISE calculates the mean free path $\bar{l}$ and its variance γ by means of a ray tracing process:

$$\bar{l} = \lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N} l_i$$

Eqn 3.2–4

$$\bar{\alpha} = \sum_{j} f_j.\alpha_j$$

Eqn 3.2–5

$$f_j = \lim_{N\to\infty} \frac{N_j}{N}$$

Eqn 3.2–6

where N = total number of collisions

l_i = distance between two successive collision points

N_j = number of collisions against element j

α_j = absorption coefficient of element j

Along with the reverberation times (according to SABINE, EYRING and KUTTRUFF) additional information is obtained, such as the mean absorption, the area covered by each material, the total area of the inner surface and the volume of the geometry model, from:

$$\bar{l} = \frac{4V}{S}$$

Eqn 3.2–7

S is found by geometrical calculation of all the elements of the mesh, but statistical reverberation times are calculated based on the statistically–weighted percentages of the various material types, not the simple area–weighted percentages.

Wall diffusion is of no effect, because the Sabine theory of reverberation assumes a fully–diffuse field as a prerequisite.

- The procedure is detailed in *Statistical Reverberation Time calculation*, Section 9.2 on Page 9–4.

3.2.3 References on statistical reverberation time

Kuttruff, H, *Room Acoustics*, Elsevier Applied Science, p 118–120

3.3 Diffraction modeling

3.3.1 Diffraction Path

3.3.1.1 Geometry

The classical methods of geometrical analysis of sound fields (MISM, RTM and hybrids) do not take into account diffraction. Because of its importance in some cases, RAYNOISE can account for diffraction in an approximate way. The program computes that part of the diffracted waves that reach the shadow zone behind an object, typically a screen. This is done by searching first–order diffraction paths. This is the shortest broken line between source (or image source) and receiver behind the screen, with the point of break (diffraction point) located on the barrier edge (see Figure 3.3–1).

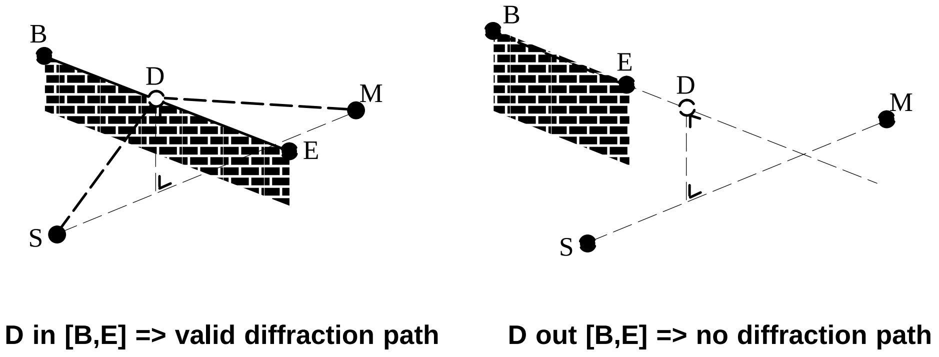

B
D
M
E
S
B
E
D
M
S
D in [B,E] => valid diffraction path
D out [B,E] => no diffraction path

Figure 3.3–1 S(image) source, M receiver, BE diff edge, D diff point

The user has to define the dominant diffraction edges in the geometry model. RAYNOISE then searches for diffraction paths that comply with following conditions:

- The diffraction point must lie between the extremities of the edge;
- The receiver must lie in the shadow zone of the screen;
- The first part of the diffraction path (source–diffraction point) should not be intersected by other polygons.

3.3.1.2 Diffraction equation

The energy contribution is now calculated with the Kurze–Anderson formula:

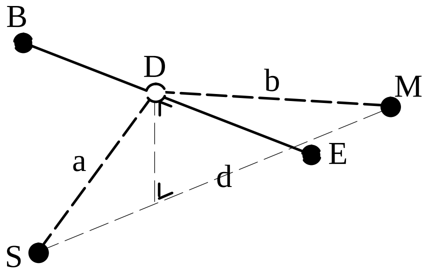


Figure 3.3–2 a=distance(S,D) b=distance(D,M) d=distance(S,M)

$$\Delta L_B = 5 + 20\log\frac{\sqrt{2\pi N}}{\tanh\sqrt{2\pi N}} \qquad (dB)$$

Eqn 3.3–1

with:

ΔL_B = attenuation (dB) due to the screen obstruction

N = Fresnel number:

$$N = 2(\frac{a+b-d}{\lambda})$$

and with:

a,b,d = distances according Figure 3.3–2

λ = wavelength

3.3.1.3 Accumulation of diffraction paths

Naturally, it is possible to have more than one valid diffraction path (Figure 3.3–3). For example, a model can consist of a floor, a screen, a receiver and three diffraction edges (one on top, one on each side of the screen).

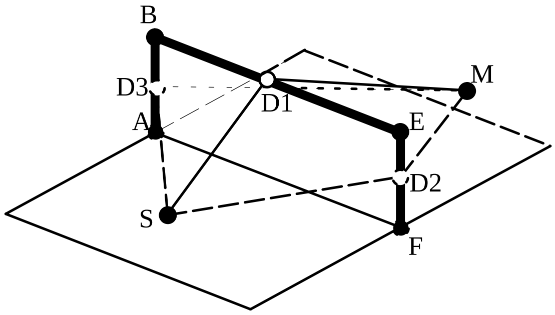


Figure 3.3–3 Three diffraction paths around a screen

In this case, all diffraction contributions will accumulate to give the pressure at the receiver, including interference due to relative phases from path−length differences, if the source is coherent (Figure 3.3−4).

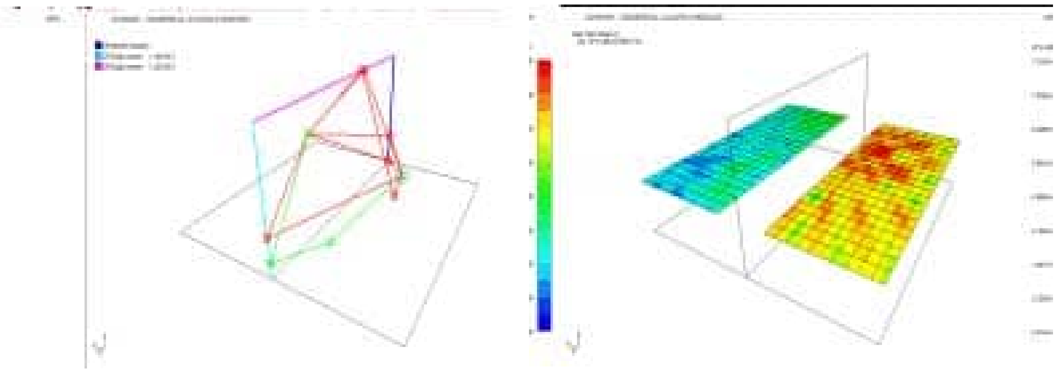


Figure 3.3–4 Multiple diffraction: paths (left) interference (right)

Likewise, it is possible to model the diffracted sound field emerging at a tunnel exit (Figure 3.3−5).

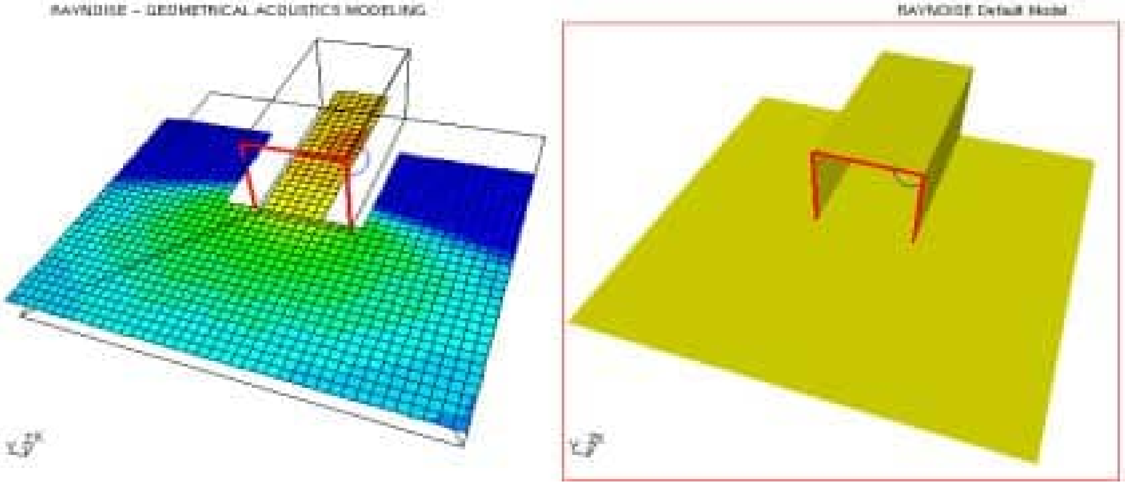


Figure 3.3–5 Diffracted field at tunnel exit

- *Diffraction edges*, Section 7.6 on Page 7−25, gives details on how to define diffraction edges in RAYNOISE.

3.3.2 Equivalent screen approach

So far, we have considered first−order diffraction. If there are multiple−order diffraction edges (*eg* several barriers behind each other) the search for the diffraction paths gives rise to very CPU−intensive calculations. Rather than impose this approach, RAYNOISE offers the 'equivalent screen' approach (Figure 3.3−6).

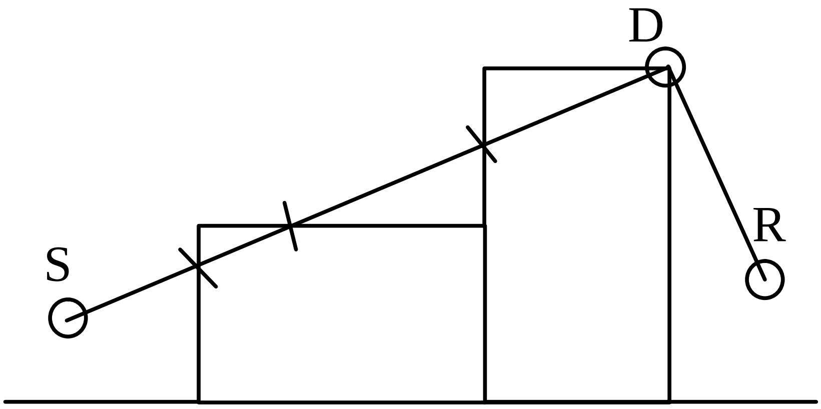


Figure 3.3–6 Multiple edge diffraction, approximated by an equivalent first–order diffraction.

The user chooses the edge that will serve as the 'equivalent screen' diffraction edge. However, in this case we have to allow the diffraction path to be cut by other polygons.

The Environment Variable EQUIVALENTDIFFRACTION determines whether this intersection condition is applied or not. (See *Environment variables*, Section 11.8 on Page 11–19, for more information). All edges must be available to be picked as diffraction edges, as set by the environment variable INCLUDEALLFACES.

3.3.3 Diffraction around a curved object

To model diffraction around a cylindrical or other curved object, we need a 'creative' use of the first–order diffraction method. Basically, we need to model two diffraction waves around the object: one via the top, one via the bottom. To make this happen, it is advisable to define four diffraction edges, equally spaced around the object (see Figure 3.3–7). In this case, whatever its position, the source or image source will always 'see' two diffraction edges that will guide a diffraction path via each side of the object.

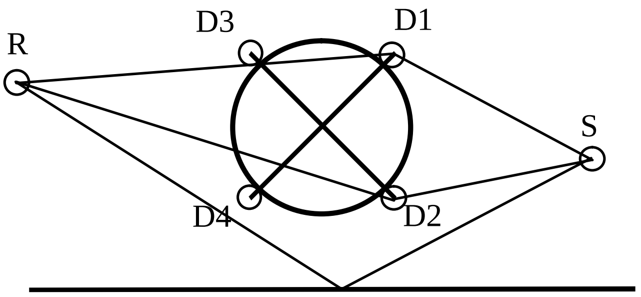


Figure 3.3–7 Diffraction paths around a cylindrical object

To accomplish this, we add two polygons crosswise to the geometry model. These serve as carriers for the four diffraction edges. The edges must emerge slightly out of the cylinder.

Note! Do not use the *Equivalent screen approach* as explained above, in Section 3.3.2 on Page 3–21: in that case, every diffraction edge would give rise to a diffraction contribution, resulting in double the correct diffracted energy. Ensure that the Environment Variable EQUIVALENTDIFFRACTION is *Off*.

Figure 3.3–8 shows a cross–section of the acoustic field for a free–field case, without a floor. One can clearly see the diffracted field being ’twisted’ around the cylinder.

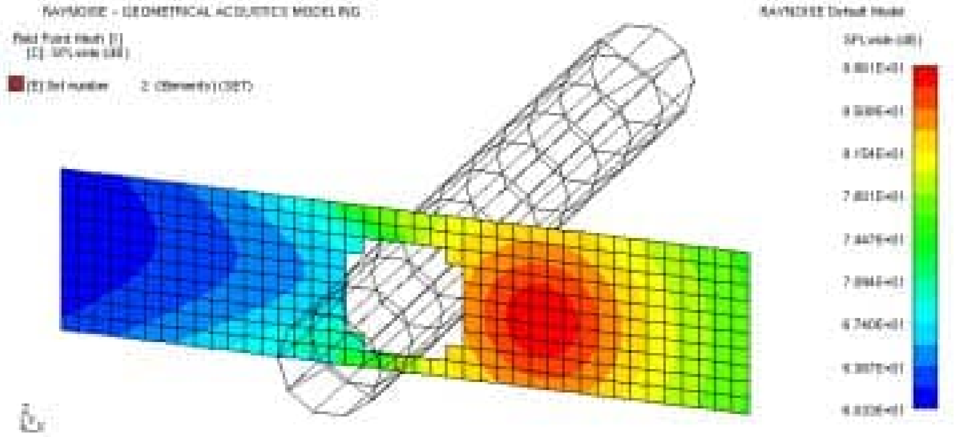


Figure 3.3–8 Sound field in dB–lin around a cylinder, no floor

If the floor is included, an important reflection is added to the acoustic field, which in turn also diffracts around the cylinder (Figures 3.3–9 and 3.3–10).

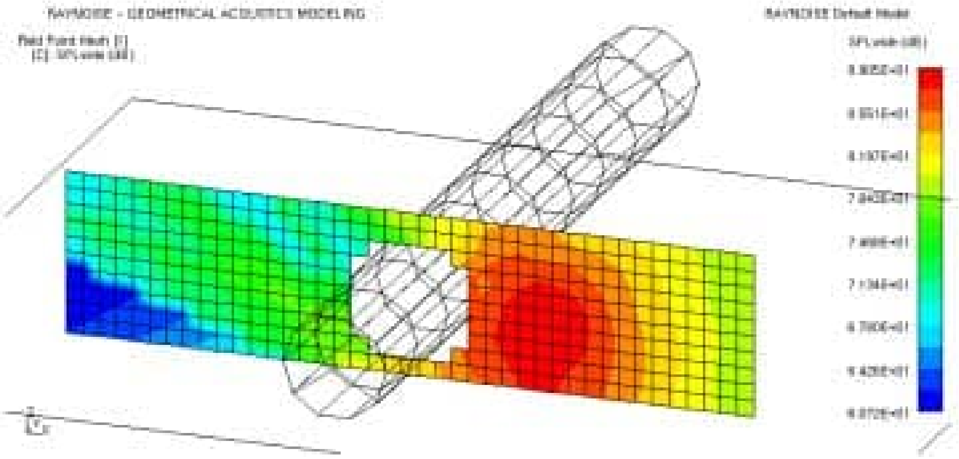


Figure 3.3–9 Sound field in dB–lin around a cylinder, with floor

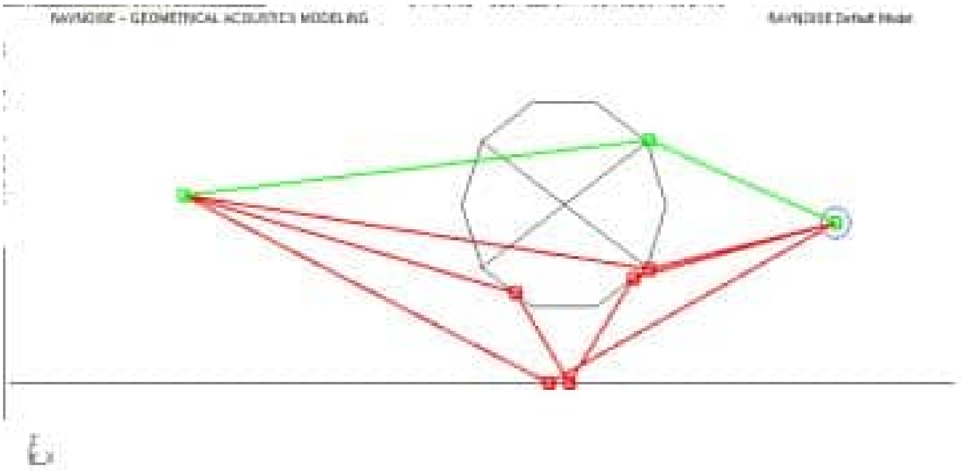


Figure 3.3–10 Ray paths from source to receiver, including reflections and diffractions.

3.4 Phase ray–tracing

3.4.1 Source types

A source in RAYNOISE is defined as coherent or incoherent. The difference is related to the use of phase during summation of the pressure contributions of the reflections.

- *Sound sources*, Section 7.4 on Page 7–10, gives details on how to define sources in RAYNOISE.

3.4.2 Sound pressure with coherent sources

If the source is defined as coherent, the steady–state sound pressure at a receiver in a cavity can be considered as the superposition of numerous components (= reflections) at the same frequency, but with different amplitudes and phase angles. The SPL is derived from

$$p_n = \sum_{i=0}^{n} \sum_{k=0}^{N_i} A_{ik} \exp(j\phi_{ik}) \exp(-jkd_{ik})$$

Eqn 3.4–1

where:

n = the reflection order (=image order)

N_i = the number of image sources of order i

p_n = the pressure

d_{ik} = the distance between receiver and image source ik

ϕ_{ik} = the phase shift at wall reflections
plus the initial phase of the source

k = the acoustic wavenumber

A_{ik} = the amplitude of reflection ik

A_i includes the distance attenuation, the power of the source and the absorption at wall reflections:

$$A_{ik}{}^{2} = \frac{\rho c W}{4\pi d_{ik}^{2}} \prod_{m} (1 - \alpha_m)$$

Eqn 3.4–2

with:

W = the power of the source

ρc = the impedance of the fluid (air)

α_m = the absorption at the mth wall reflection.

3.4.3 Sound pressure with incoherent sources

In the case of incoherent summation, one gets:

$$p_n = \sum_{i=0}^{n} \sum_{k=0}^{N_i} A_{ik}^2 \qquad \textit{Eqn 3.4-3}$$

with the same definitions as above for the coherent case.

3.4.4 Applications

The use of coherent sources is useful in cases where a lot of interference is expected, *eg* because two loudspeakers are positioned close to one another, or because the frequencies are rather low with regard to the room dimensions, modes, *etc*.

The coherent *vs* incoherent label is also necessary when calculating STI maps, where some sources emit the speech signal, other sources produce masking noise.

3.4.5 Echogram interpolation

Echogram interpolation allows the derivation of echogram results at frequencies which were not included in the primary ray–tracing calculation. This is useful, especially, in determining narrow–band transfer functions for selected source–receiver combinations.

The principle is to perform Mapping calculations for a few 'master' frequencies only (*eg* the octave–band center frequencies). Then, results at intermediate ('slave') frequencies are derived by interpolating between the echograms at the master frequencies next above and next below each intermediate frequency. The interpolation is carried out in a linear sense in the magnitude and phase dimensions, for each pair of impulse arrivals (from the higher and lower master frequencies, respectively). The phase data are relevant to the case with coherent sources, which use phase ray–tracing.

3.5 Diffuse reflections

3.5.1 Diffusion coefficient

The diffusion coefficient *d* of a surface is defined as the ratio between reflected sound power in non–specular directions and the total reflected sound power:

$$P_{diffuse} = dP_{reflected} = d(1 - \alpha)P_{incident}$$

Eqn 3.5–1

3.5.2 Calculation procedure with diffusion

The method used in RAYNOISE to handle diffuse reflections is as follows:

First, a beam–tracing method is used to find the specular parts of all the reflections. The specular parts are calculated according to:

$$P_{spec} = (1 - d)(1 - \alpha)P_{incident}$$

Eqn 3.5–2

This is done for all frequencies simultaneously.

In a second step, the missing diffuse reflections are handled by a method that uses rays as transporters of energy. Each time a ray hits a surface, a random number between 0 and 1 is generated. There are two possibilities:

- If the number is lower than the diffusion coefficient of the surface, a secondary source is generated at the collision point, which gives its diffuse energy to all the receivers. A single ray (beam) is radiated in a random direction.
- If the number is higher than the diffusion coefficient of the surface, no energy contribution is given to any receiver (since the specular parts are already handled in the first step). The ray is reflected in a specular direction.

Since the diffusion coefficients are frequency–dependent, and the diffuse wall reflections are handled by random ray reflections, a new ray–tracing process has to be started for each frequency value.

Note that this algorithm combines a deterministic method with a statistical method and that, by doing so, the full matrix of combinations is taken into account:

- Specular–to–specular
- Diffuse–to–diffuse
- Specular–to–diffuse
- Diffuse–to–specular

The addition of diffuse reflections to an echo/histogram is illustrated in Figure 3.5–1. (See Echogram displays, Section 10.4 on Page 10–41, for more information on the plotting procedure).

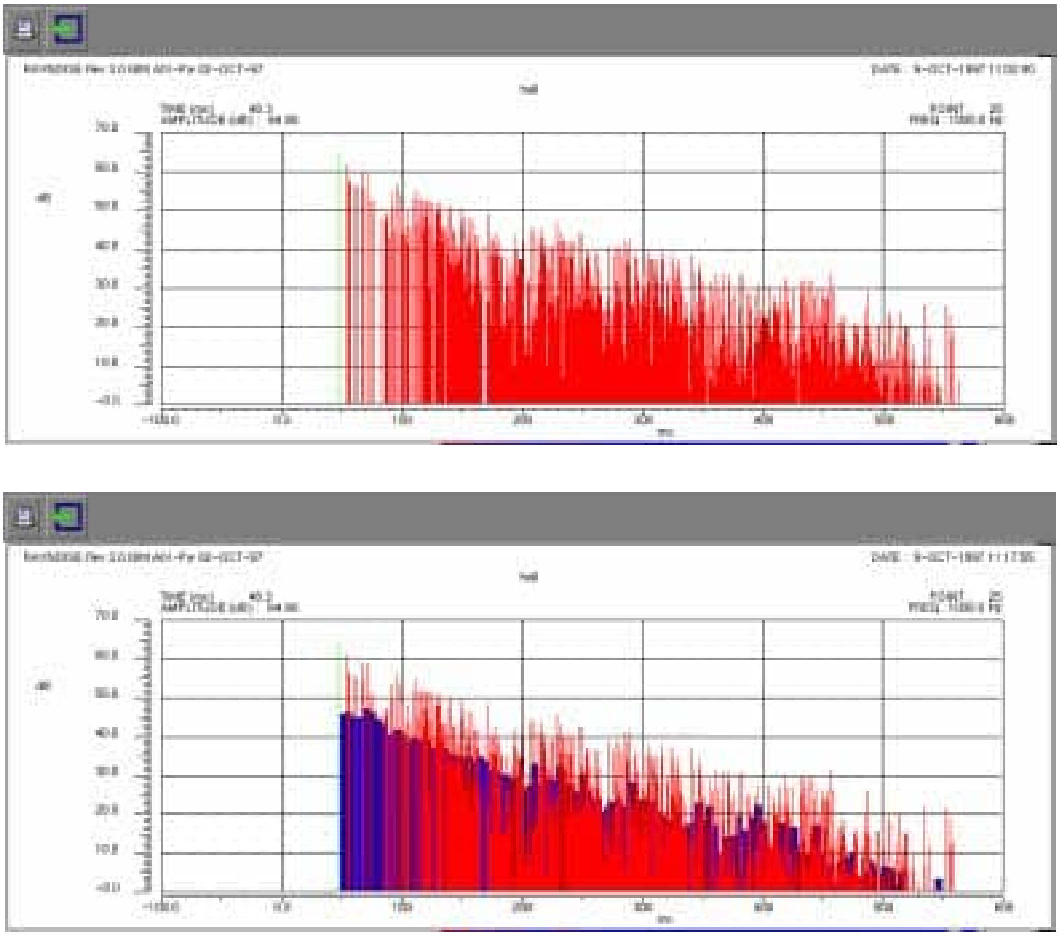


Figure 3.5–1 Echogram (a) without (b) with diffuse reflections

3.5.3 References on diffusion

Heinz, R., *Applied Acoustics* Vol.38, p145 (1993)

Dalenbäck, B.–I. et al, J. Audio Eng. Soc. Vol42, No. 10, (1994)

3.6 Sound transmission modeling

3.6.1 General principles

Sound transmission between rooms and between inside and outside is modeled in RAYNOISE, using the concept of transmission sources. These are sources that are positioned at the centroid of a polygonal surface and radiate hemispherically with the main axis according to the positive or negative normal of the surface. The power of such a transmission source depends on the excitation level at the other side and the Transmission Loss value of the wall. The excitation level is derived from the calculated SPL(dB) and Direct Energy(dB) at a user−specified field point, which should be placed close to the wall.

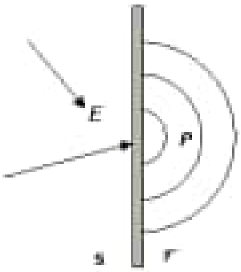


Figure 3.6–1 Features of a sound transmission source

3.6.2 Transmission source power

The sound power of the transmission source is given by:

$$P = I.S.t$$

Eqn 3.6−1

where: P = the sound power

I = sound intensity at excitation side

S = area of the transmitting panel

t = transmission coefficient

The intensity at the excitation side relates to the sound pressure according to:

$$I = \frac{p_{rms}^2}{\rho c}$$

Eqn 3.6–2

for a plane wave incidence (free field conditions) or:

$$I = \frac{p_{rms}^2}{4\rho c}$$

Eqn 3.6–3

for a diffuse field (reverberant room).

3.6.3 Calculation procedure

RAYNOISE calculates the sound pressure at the field point and assesses the degree of diffusion. A transmission from a reverberant room will use the diffuse excitation, while a transmission from an exterior field will use the plane wave incidence condition. Intermediate cases (partly diffuse field) will use a factor in the denumerator between 1 and 4, determined from the total to direct energy ratio.

First, all non–transmitting sources are calculated. The degree of diffusion is assessed by considering the ratio of direct rays to total energy. Then, the powers of the transmitting sources are assessed from the existing noise field, their propagation is traced, and their contributions are added to the noise field.

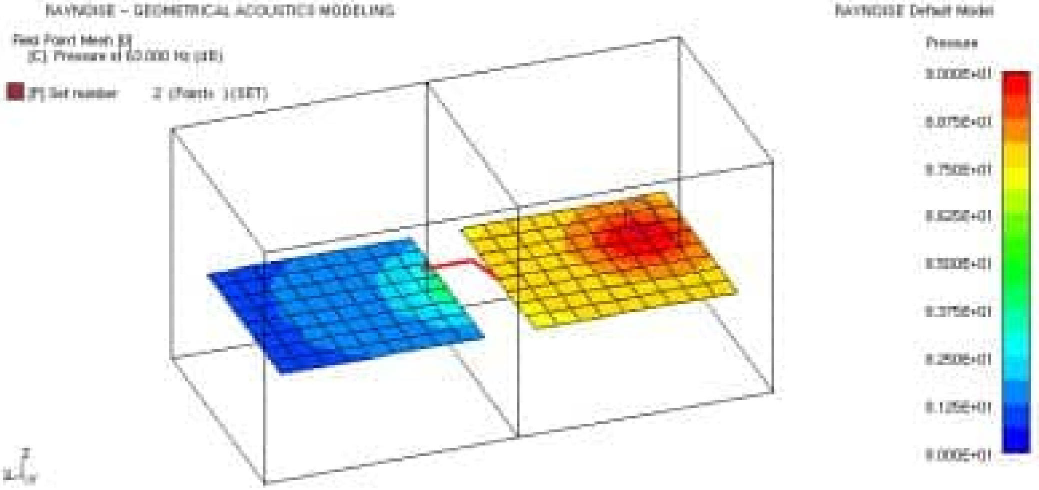


Figure 3.6–2 Example of sound transmission between rooms

Note! RAYNOISE models one–way transmission only.
If two–way transmission is of interest, it can be modeled by two transmission sources, one on each side of the partition, linked to corresponding field–points near the opposite sides. However, the SPL at a field–point, used to find the power of a transmission source, will only include the energy arriving from other transmission sources if they are calculated first: the results will therefore depend on the order in which transmission sources are defined. A fully–recursive, two–way transmission cannot be accurately calculated, and at best only an approximation can be made.

3.7 Auralization

3.7.1 Definition

“Auralization is the process of rendering audible, by physical or mathematical modeling, the sound field of a source in a space, in such a way as to simulate the binaural listening experience at a given position in the modeled space.” (Mendel Kleiner). Auralization in RAYNOISE enables the recreation of the aural impression of the acoustic characteristics of a space, whether it is outdoors or indoors.

3.7.2 Auralization procedure

The auralization procedure in RAYNOISE is based on the following three steps (see Figure 3.7–1):

- Impulse response (IR) calculation using computer modeling
- Derivation of Binaural Impulse Response (BIR) using Head–Related Transfer Functions (HRTF)
- Filtering of source material (music, speech, noise,...) by the BIR, using Digital Signal Processing (DSP) *ie* Convolution.

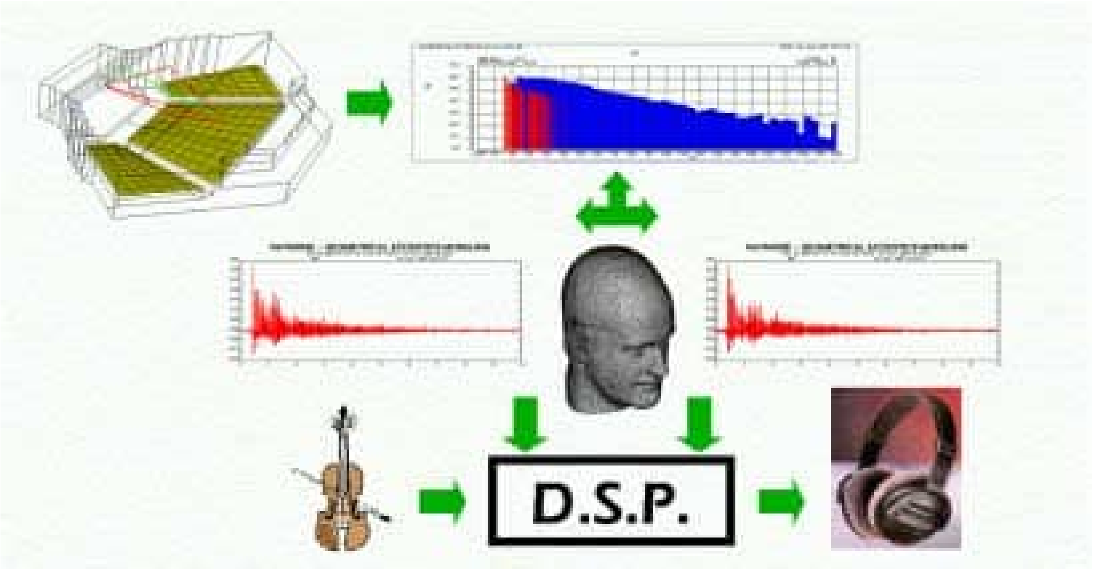


Figure 3.7–1 Flow diagram of the auralization process

The core of the auralization procedure is the generation of the Finite Impulse Response (FIR) based on the echogram–histogram information. To do so, a causal filter is generated for each reflection of the echogram, using its transfer function, accounting for source directivity, wall absorption, air absorption and path length distance. After multiplying by the appropriate HRTF corresponding to the incidence angles, an impulse response is derived using Inverse FFT. All impulse responses are then summed, with time delays corresponding to the path−length distances.

For the histogram, containing the high−order reflections and the diffuse reflections, a similar operation is done, except that no angle−of−incidence information is available and thus no HRTFs are used. Each histogram is randomly sampled to generate the filters. The average density of the samples is 5000 per second.

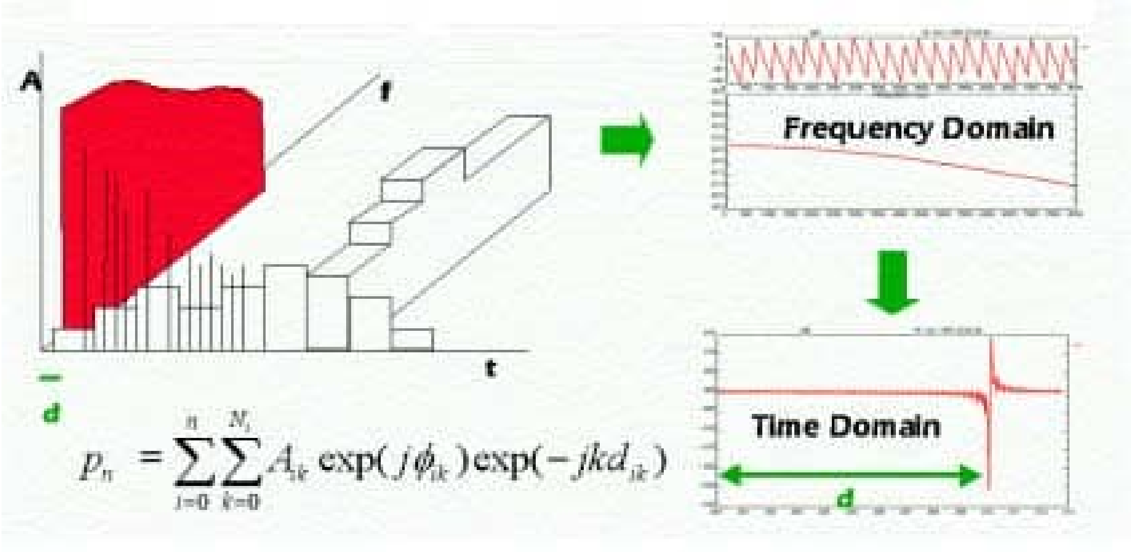

A
f
t
d
$p_n = \sum_{i=0}^{n}\sum_{k=0}^{N_i} A_{ik}\exp(j\phi_{ik})\exp(-jkd_{ik})$
Frequency Domain
Time Domain
d

Figure 3.7–2 Filter synthesis from reflections

The convolution of the BFIR with source material is done using FFT, thereby treating end effects by zero padding. The normalization of the files containing the BFIR and the convolution, by default uses the maximum available dynamic range (for a 16−bit file) but alternatively it can be fixed by the user: in the latter case, multiple files can be given the same scaling, to enable proper comparison or combination.

3.7.3 References on auralization

Martin J. *et al*, J. Acoust. Soc. Am. 94(6), 1993

Kuttruff, H., J. Audio Eng. Soc. Vol. 41, No. 11, 1993

Gardner, W.G., *et al.*, J. Acoust. Soc. Am. 97(6), 1995

Numerical Recipes, 2nd edition, Cambridge University Press

4

The graphical user interface

This chapter describes the general appearance of the RAYNOISE windows and how the user controls the execution of the program, including:

○ *Screen layout and content of each window*

○ *Menus and Toolbar*

○ *Button files*

○ *Command language and command files*

○ *Extracting information*

○ *Exit from RAYNOISE*

4.1 Overview of the GUI

4.1.1 Summary

RAYNOISE is driven by a Graphical User Interface (GUI) based on Motif (or 'Windows' on PC). The screen layout comprises three principal windows: the Main Window, the Echo Window and the RAYNOISE Command History List Window.

The Main Window contains:

- a Menu Bar for activating the commands and dialogs;
- a Tool Bar, with a row of graphical buttons for applying the most−frequent graphical commands;
- a graphical window for displaying the model and performing post−processing;
- a keyboard input area for typing the commands using the RAYNOISE Command Language syntax.

The Echo Window lists all messages produced by RAYNOISE.

The RCL History Window contains the trace of the commands.

The mesh and results display (*what* is displayed and *how*) can be changed using the **View** and **Display** menus.

A fourth window contains a series of buttons, which are user−programmable. These can be used to speed−up the access to commonly−used commands or specific calculation sequences and to customize the user's interaction with RAYNOISE.

4.1.2 General arrangement of windows

4.1.2.1 Screen layout

The screen layout provides several windows which are open in parallel. The initial sizes and positions of the windows are determined by Environment Variables in the Set−up File RAYNOISE.STP or the user's modifications to these in the Profile file RAYNOISE.PRF (see Section 11.8.4 on page Page 11−39 and also the Installation Manual for more details). The sizes and positions of the windows can be altered dynamically by the user. The altered positions can be saved to the permanent file RAYNOISE.PRF so that they are used again when another RAYNOISE session is started, or else the modified values are lost on Exit from RAYNOISE.

4.1.2.2 Details

The detailed content of each window is described in more detail in the following sections:

- Main Window (Section 4.2 on Page 4–5)
- RCL History Window (Section 4.3 on Page 4–9)
- Echo Window (Section 4.4 on Page 4–10)
- Button Window, (Section 4.5 on Page 4–11)

4.1.3 Screen resolution

Some dialogs of the RAYNOISE user interface can occupy a large part of the screen, if the resolution is limited. The user cannot always alter dialog sizes, although the sizes of the GUI windows can be controlled dynamically. (See also *Screen layout*, Section 4.1.2.1 on Page 4–3).

Although it is possible to move over–size dialogs up, down or sideways, it is recommended to have at least 800x600 pixels screen resolution and preferably 1024x768 or more, for ease of use.

4.2 Main graphical window

4.2.1 Layout

The Main Window is laid out with four regions:

- Menu Bar: contains key words from which pull–down menus appear
- Tool Bar: contains buttons with icons, mainly for graphical actions.
- Graphics area: contains the viewport(s) in which the model is plotted
- Command line: is used for entering commands in the RAYNOISE Command Language. These will often be pulled–down from the RCL History Window.

The size and position of the Main Window can be altered dynamically by the user. (See also *Screen layout*, Section 4.1.2.1 on Page 4–3). The menu bar and tool bar are described further in the following sections.

4.2.2 Menu bar

The Menu Bar contains key words which cover the main actions in making and calculating with a RAYNOISE model, generally in sequence from left to right. Clicking on each keyword results in a cascading menu or series of menus appearing below, from which the necessary action can be picked with the mouse, or a dialog opened for the entry of specific data or choices.

More details on the purpose and use of each of the menus is given in the sections referenced below:

- **File** (Chapter 5)
- **Geometry** (Chapter 6)
- **Model** (Chapter 7)
- **Analysis** (Chapter 9)
- **Inquire** (Section 4.8 on Page 4–17)
- **Tools** (Chapter 11)
- **View** (Chapter 8)

- **Postprocess** (Chapter 10)
- **Display** (Chapter 12)
- **Help** (Section 4.7 on Page 4–16)

4.2.3 Tool bar

4.2.3.1 General description

The toolbar contains a series of icon–buttons, nearly all of which activate graphical actions, grouped as listed below, from left to right:

4.2.3.2 Graphical utilities

Undo: reverses the last (graphical) action (*not* the last modeling action).

Plot: sends the contents of the Main Window to the printer (or a plot file).

Rescale: fits the currently–visible objects to the active viewport.

Enquire: identifies the element (nearest to the viewpoint) and its nearest node. If results are visible (*eg* contours) the corresponding result value is printed.

4.2.3.3 Graphic motion tools

The graphic motion and picking tools require the left mouse button to be held down whilst the relevant box, drag, zoom, *etc* is performed. On releasing the mouse button, the view locks to the new position.

Zoom: click with the left mouse button and drag a box to zoom in.

Interactive zoom: toggle *On*, then enlarge/reduce the view dynamically.

Rotate: toggle *On*, then rotate the model dynamically

Drag (translate): toggle *On*, then drag the viewpoint dynamically

4.2.3.4 Graphic picking tools

The graphic picking tools initiate the creation of a Set if activated alone, outside another dialog. They can be used from within a dialog, when a selection of objects is required: in this case, the dialog will have a Node (Element, ...) Selector button labeled green. Do not press this button, just press the button for the preferred graphic picking tool and make the selection graphically. The RAYNOISE Command Language History gets an equivalent command, based on the internal item numbers.

Single select: used when appropriate to pick a single node or point.

Box select: toggle *On*, then click, hold and drag a box to select several objects

Free select: toggle *On*, then click, hold and draw a free–form shape to select several objects. The boundary line must finish near the starting–point and is closed automatically. Avoid 'cross–overs', although the program will try to resolve them.

- See *Sets* (Section 6.5 on Page 6–27) for more information.

4.2.3.5 Numbering tools

Turn numbering of entities on/off as toggles:

Element numbers: puts external element numbers on visible model(s)

Node numbers: puts external node numbers on visible model(s)

Point numbers: puts point numbers on visible field point mesh(es)

4.2.3.6 Rendering and visualization tools

Show normals: draws the element normal vectors (toggle on/off)

Wireframe/face–painting: rendering methods for all objects

Hide All: removes the display of results and groups (sets, boundary conditions, ...) and returns to a plain display of the mesh.

Stop process: stops the calculation process (*ie* ray trace in Mapping) at the next checkpoint. Note that any the results in the database at that time may be incomplete and inconsistent: the stopped calculation step should be re–started when appropriate (*eg* after making a new frequency selection).

4.2.3.7 Tool tips

Each Toolbar icon has an associated tip, which is displayed if the mouse pointer dwells over the icon for a short time. These tips are user–programmable: to alter the standard installation, edit the file RAYNOISE.TIP, usually found in the ’bin‘ directory of the installation. (Consult your installation manual and use the **Tools, Environment Var.** dialog to check the Environment Variable TIPFILE, see Section 11.8 on Page 11–19, which gives the path of the file).

4.3 RCL history window

The RAYNOISE Command Language History Window contains the record of all commands passed to the program during the present session, whether created via GUI actions (menu selections...) or entered on the command line.

A command can be captured from the RCL History Window and passed to the command line, by highlighting it with a single mouse click. It can then be edited before being used as a command–line entry. If the command is double–clicked in the RCL History Window, it is entered immediately, without editing.

On Exit from RAYNOISE, the contents of the RCL History Window can be saved to a Journal file (see Section 11.9 on Page 11–42) which can then be re–used as a command file with the **File, Read** dialog (see Section 4.6.4 on Page 4–14) after any necessary editing and re–naming.

4.4 Echo window

The Echo Window contains all the text messages output from the program. The contents of the window are also saved in the file RAYNOISE.LOG which is available after Exit from the program, usually in the directory from which the RAYNOISE session was launched. (This may be installation–dependent).

4.5 Button window

4.5.1 General principles

The Button Window is a dynamic, user–programmable dialog. On starting RAYNOISE, an initial set of buttons is displayed: a default set of buttons for this, is supplied with the installation. The Button Window is associated with a Button File, in which each button corresponds to one of the following types of operation:

- Execute a RAYNOISE command
- Show a RAYNOISE dialog (for the user to complete)
- Call another button file, to change the contents of the Button Window.

A tip is also associated with each button, which is displayed if the mouse pointer dwells over the button for a short time.

There is no limit to the numbers of button files which can be set up, but only one is active (and has its associated buttons visible) at any one time. It is thus possible to set up a cascading sequence of buttons, which offers a way to carry out repetitive tasks more easily.

The visible buttons are controlled by the active button file, which is set by the Environment Variable BTNFILE, found in the **Tools, Environment Var.** dialog (see Section 11.8 on Page 11–19).

4.5.2 Contents of a button file

A button file is a text file, with a derivative of the RAYNOISE Command Language:

- STR gives the label to be placed in the button.
- TIP gives the text of the associated tip
- The command follows, enclosed in parentheses ( ).

The following examples show the three types of operations:

- Execute a command

```
STR New Model
TIP Create a Model
(New Name `m1´ Model 1 File m1.rdb Return)
```

- Show a dialog

```
STR Get Geometry
TIP Import the Mesh
(Show Dialog File/Import Return)
```

- Call another button file

```
STR Next Step
TIP Branch to the next Button file
(Environment BTNFILE `OTHER.BTN´ Return)
```

4.6 RAYNOISE command language

4.6.1 Features

The GUI of RAYNOISE provides an effective user interface, but the program is driven finally by the RAYNOISE Command Language (RCL). All actions in the dialogs of the GUI produce command sentences, which are usually executed when the 'Apply' or 'OK' button (or, in the case of possibly cumulative definitions like sources, when 'Add') is pressed.

4.6.2 Command language history

The commands produced by the GUI dialogs or entered directly by the user are recorded in the RCL History Window, with the exception of graphical commands (interactive rotation of model, etc) which are only recorded if the necessary option is set (Environment Variable GRAPHICALCOMMANDS = TRUE, see Section 11.8 on Page 11–19). The RCL History Window is scrollable. The commands can be repeated by selecting and double–clicking with the mouse, or transferred to the command input line by a single mouse click, and altered before being entered.

4.6.3 Command syntax

The details of the syntax of each individual command are explained when necessary in the sections describing the related menus and the corresponding commands.

In general, if commands are entered directly, the keywords and their complements and parameters can be shortened to the first four characters. Separators can be blanks, commas or equals signs (=). Real numbers can be entered in integer or real formats, including E– or D–notations. Parameters or data values must be on the same line and following the keyword or complement to which they relate. Most commands require the word RETURN to indicate the end of the related entries and return to the top–level of commands.

Comments are indicated by a preceding semi–colon (;) or left brace ({) character. Comments can be added at the end of a command line, or a complete line may be a comment. Blank lines are ignored when reading the command file.

Hint: It can be useful to make an outline command file using the GUI and saving the Journal file, to get the correct syntax for the required commands, even if it is then edited and extended greatly and used in a semi–batch mode (see Reading a command file, Section 4.6.4 on Page 4–14).

4.6.4 Reading a command file

RAYNOISE can be used in semi–interactive mode, using a command file containing RAYNOISE commands. Such a procedure is executed by the **File, Read** dialog:

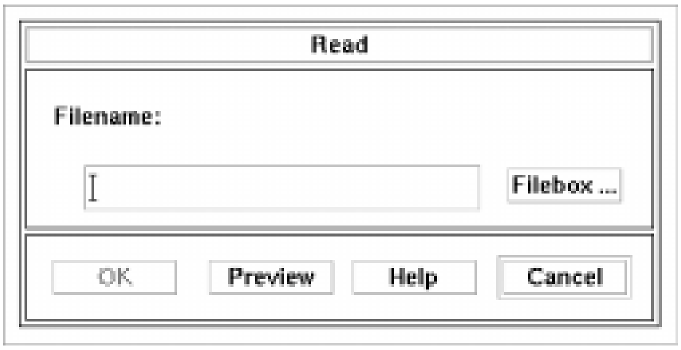


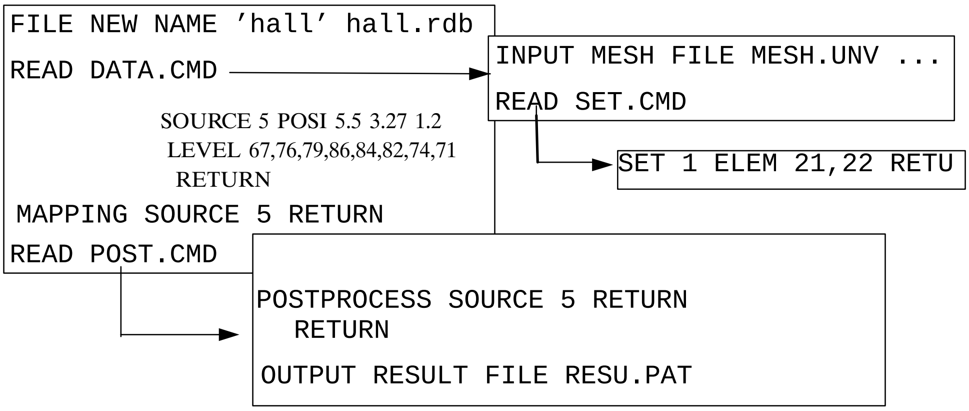
Figure 4.6–1 File, Read dialog

RAYNOISE will read all commands until the end of the file is reached or until the command STOP is encountered. Control is returned to the user and the command file is closed.

The command file can be created using a normal text editor in accordance with the syntax of the RAYNOISE command language, or it may be the result of saving a journal file from a previous RAYNOISE session, possibly edited. (Saving a journal file can be an effective way to ensure the correct command syntax is used). For more information on manipulating journal files within the RAYNOISE session, see *Journal*, Section 11.9 on Page 11–42. See *Exit*, Section 4.11 on Page 4–25, regarding the saving of the journal file when ending a RAYNOISE session.

Command files can execute other command files. A maximum recurrence level of 5 is allowed. When RAYNOISE stops reading the command file of a given level, it goes back to the previous command file which called it.

Hint: It may be useful to insert the keyword **Test** (see Section 11.2.4 on Page 11–5) into a command file before reading it, if it is desired to avoid the execution of a long calculation: only modeling and other data–handling and display tasks will then be executed.



```
FILE NEW NAME 'hall' hall.rdb
READ DATA.CMD
              SOURCE 5 POSI 5.5 3.27 1.2
               LEVEL 67,76,79,86,84,82,74,71
                RETURN
 MAPPING SOURCE 5 RETURN
READ POST.CMD
```

```
INPUT MESH FILE MESH.UNV ...
READ SET.CMD
```

```
SET 1 ELEM 21,22 RETU
```

```
POSTPROCESS SOURCE 5 RETURN
   RETURN
OUTPUT RESULT FILE RESU.PAT
```

Figure 4.6–2   Command file recursivity

4.7 Getting help

4.7.1 Context–sensitive help

Each dialog of RAYNOISE normally contains a Help button as well as the Cancel, Apply or similar. The Help button gives access to the on–line documentation at an entry–point which is context–sensitive. After the on–line help is opened in this way, the links to other parts of the complete documentation, table of contents, a browser, search engine, and so on, may be accessed.

4.7.2 Primary Help button

At the level of the Main Window menu bar, there is a Help button, giving access to the on–line documentation at the Contents page.

Note! The on–line documentation files must either be installed on the the computer or the network, or accessible via a CD drive with the CD loaded, to enable access to the documentation. This is installation–dependent.

4.8 Extracting information

4.8.1 Inquire menu

The menu **Inquire** is used to extract information from the RAYNOISE database. Information on geometry, sources, materials, available results and frequencies, models and similar, can be obtained.

The **Inquire** menu contains the following entries:

- **Extract (select)**: gives information on a selection of Elements, Nodes, Faces, Points, Sets, Tables, Layers, Materials, Sources, Diffraction Edges.
- **Extract (full)**: gives information on all Elements, Nodes, Faces, Points, Tables, Layers, Diffraction Edges.
- **Database**
- **Models**
- **Mapping**
- **Sets**
- **Materials**
- **Sources**
- **Reverberation**
- **Results...**
- **Echograms...**
- **Summary**

If the **Extract (full)** menu is chosen, information is directly listed in the Echo Window and all available information on the selected type of data are listed. Selection sub–dialogs mentioned below do not appear.

If the **Extract (select)** item is chosen, or another item is chosen which may produce a large volume of data to be presented, an item selection dialog is opened and, depending on the type of information required, a node, element, point selection, or frequency or echogram selection must be made. The required information is then listed in the Echo Window.

4.8.1.1 Database

The **Inquire, Database** menu entry can be used to list information about the records stored in the active database file. This is usually for diagnostic purposes or for advanced users.

4.8.1.2 Diffraction edges

The **Inquire, Extract, Diffraction Edges** menu entry can be used to list information about the diffraction edges in the active model. A sub−dialog opens, which lists the diffraction edges by name. After making a selection by highlighting with the mouse, the **OK** button causes the details to be listed: internal and external (user) edge numbers, with the corresponding element and the edge of it used to define the diffraction edge.

4.8.1.3 Echograms

The **Inquire, Echograms** menu entry can be used to list information from the echograms stored in the active database file. A sub−dialog opens, for the selection of the field point and the corresponding frequency, for which results are required: the field−point number is listed first, followed by the frequency in brackets [ ]. After making the selection and applying **OK**, the detailed list has two sections:

- **Echogram** data, giving each discrete echo, with its source, arrival time, amplitude, phase and number of reflections along the path.
- **Histogram** data, listing the histogram bin contents.

Echogram listings can often be very extensive.

4.8.1.4 Elements

The **Inquire, Extract, Element** menu is used to list information about a selection of elements. The listing contains:

- the element internal and external number
- the element type
- the element material identification number
- the element topology, *ie* the list of nodes defining the element

4.8.1.5 Faces

The **Inquire, Extract, Face** menu is used to list information about a selection of faces, *ie* free edges. The listing contains:

- the face internal number
- the internal number of the element to which the face belongs

- the face type (= an element type, which will be a ‘line element’)
- the face number within the element
- the face topology, *ie* the list of nodes defining the face

4.8.1.6 Layers

Layers and the elements which they contain can be listed in the same manner as Sets of elements, via **Inquire, Extract**...

4.8.1.7 Mapping

The **Inquire, Mapping** menu entry can be used to list information about the Mapping calculations which have been performed and for which the results are stored in the active database file. It lists the MAPPING frequencies, with the corresponding number of field points, Storelevel, HistogramLength and HistogramInterval used.

The detailed Echogram (IMAGE) storing follows, with the corresponding field points, the numbers of reflections stored for each, the RayPathStore, HistogramLength and HistogramInterval used, and the number of (master) frequencies for which echograms are stored.

4.8.1.8 Material

The **Inquire, Material** or **Inquire, Extract(select), Material** menu is used to list information about one or all materials defined in the current model. The listing contains the list of properties and depends on the type of material:

- Acoustic absorption properties
- Acoustic diffusion properties
- Acoustic transmission properties

4.8.1.9 Models

The **Inquire, Models** menu is used to list information about currently–open models.

4.8.1.10 Nodes

The **Inquire, Extract, Node** menu is used to list information about a selection of nodes. The listing contains:

- the node internal number
- the node external number
- the three node coordinates

4.8.1.11 Points

The **Inquire, Extract, Points** menu is used to list information about a selection of field points. The listing contains:

- the point internal number
- the point external number
- the three point coordinates

4.8.1.12 Reverberation

The **Inquire, Reverberation** menu entry lists the statistical reverberation results. The values in the standard octave bands are tabulated. See *Extracting reverberation time results*, Section 9.2.3 on Page 9–4, for more details.

4.8.1.13 Results

The **Inquire, Results** menu is used to list information about sound pressure levels and any other results from the database. A sub–dialog opens, for the selection of the field points for which results are required and the selection of the type of results to be listed. The results are grouped into five categories, related to different applications:

- **Pressure** gives the results for pressures as physical quantities (*eg* in MKS units) and SPL in dB (Linear).
- **SPL** gives the results for stationary sound: Sound Pressure Level (SPL), Noise Criterion (NC), Noise Rating (NR), Direct energy and Total–to–Direct energy ratio.
- **Speech** gives the results for Speech Transmission Index (STI), Definition, Echo Criterion (EC) for Speech and Early Reflection Ratio (ERR).
- **Music** gives the results for Clarity, Lateral Efficiency (LE), Central Gravity Time (TCG) and Echo Criterion (EC) for Music.
- **Reverberation** gives the results for Reverberation Times based on different decay ratios (RT60, RT30, RT20) and Early Decay Time (EDT).

See the chapter *Acoustic quantities*, on Page 13–1, for further details and formal definitions. Note that not all results types may be available.

Select the field points by direct specification, or by opening the field–point selection sub–dialog, or by using the graphical picking tools. The corresponding results are tabulated in the Echo Window, for all available frequencies.

4.8.1.14 Sets

The **Inquire, Sets** menu is used to list information about all the sets. The Inquire, **Extract (select), Sets** menu lists the sets in a selection dialog and then lists, for each selected set:

- the set number
- the set name
- the type of items in the set
- the full list of items in the set (and, for elements or faces, their connectivities).

4.8.1.15 Sources

The **Inquire, Sources** menu is used to list information about a selection of sources defined in the currently–active model. The **Extract (full)** menu lists, for each source:

- the source number
- the source name

The **Extract (select)** menu lists, for the selected source:

- the source number
- the source name
- the position (co–ordinates) of the source
- the division data for the source, in the case of lines or areas
- the sound power of the source, versus frequency
- the directivity and emission angle data for the source

4.8.1.16 Summary

The **Inquire, Summary** menu lists summary information about the currently–active model.

4.8.1.17 Tables

The **Inquire, Extract, Tables** menu is used to list information about a selection of tables. The **Extract (full)** menu lists all tables, giving:

- the table number
- the table name
- the number of values in that table

The **Extract (select)** menu lists, for the selected table:

- the table number
- the table name
- the values contained in the table

4.9 Logfile output

When RAYNOISE is used in interactive mode, all the messages issued by the program are displayed on the user's screen (command line mode) or in the Echo Window (GUI mode). These messages are also saved in a file, RAYNOISE.LOG.

RAYNOISE output can be diverted to another logfile using the **File, Logfile** dialog:

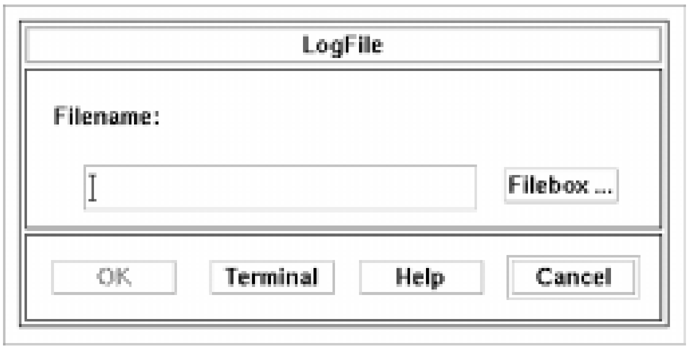


Figure 4.9–1 File, Logfile dialog

Output can be directed to the screen again by selecting TERMINAL as logfile.

See also *Printing displayed information*, Section 4.10 on Page 4–24.

Hint: It can be useful to direct some data extracted from the model, such as lists of node coordinates, to a logfile for subsequent printing or as an archive for tracking purposes. Logfile output can be re–directed and returned to the Echo Window any number of times.

4.10 Printing displayed information

4.10.1 Text information

Text information (*ie* what is normally displayed in the Echo Window) can be printed by first arranging to direct the output to a file, using the **Logfile** command (see Section 4.9 on Page 4–23) and then printing that file, using normal system printing facilities. The latter will be system–dependent. It may also be possible to capture such text files via word–processing software.

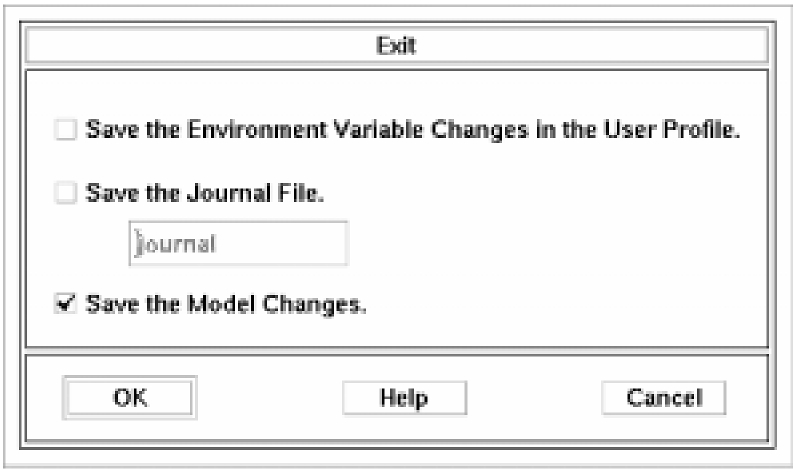
If the output is not re–directed to other files during the RAYNOISE session, using the Logfile command, the complete output to the Echo Window for the current session is saved in the file RAYNOISE.LOG upon Exit from RAYNOISE.

4.10.2 Graphical plots

The currently–displayed view in the main window is plotted by clicking on the printer button on the toolbar. The contents of the window are either directed immediately to an on–line plotter/printer, or directed to a file which can be plotted later. The set–up may be system–dependent, and is also controlled by the current choice of *Plotting Driver* (see Section 12.1.9 on Page 12–13) which also controls frame dimensions, numbers of colors and other parameters. If the plot is directed to a file, and the standard file–name already exists (for instance, if a series of plots is made in the same session of RAYNOISE, or using the same directory path) the name is given an additional numerical extension to show the incrementation.

4.11 Exit

The **File, Exit** dialog closes the RAYNOISE session and offers several saving options:



□ Save the Environment Variable Changes in the User Profile.

□ Save the Journal File.

journal

☑ Save the Model Changes.

Figure 4.11–1 File, Exit dialog

Save changes to models in databases (*On* by default) saves all open databases in their current state.

It is also possible to save the content of the RCL History Window to a journal file (*Off* by default) and to save the environment to the Profile file (*Off* by default): this creates a *Profile file* RAYNOISE.PRF for the user, containing the current values of all Environment Variables (see Section 11.8.5 on Page 11–40).

5

Data handling

This chapter describes the RAYNOISE database file, how it is used and how to communicate with external files:

5.1 General description of data handling

5.1.1 The model database file

All information related to a model prepared and analyzed in RAYNOISE is stored in a single file called the **model database** or model file. A model file generally uses the extension ***.rdb*** (RAYNOISE data base). When RAYNOISE is started, a default database called *_default.rdb* is created and used. The **File** menu contains all commands related to database operations, which are discussed later in this chapter:

- **New**
- **Open**
- **Activate**
- **Database**
- **Close**
- **Save and Save As**

It also contains commands used to read and write information coming from external programs like CAD or FEA programs or test data analysis:

- **Import**
- **Export**

Two more commands manage the re–direction of RAYNOISE input or output from the screen or keyboard to a file:

- **Read** (see Section 4.6.4 on Page 4–14)
- **Logfile** (see Section 4.9 on Page 4–23)

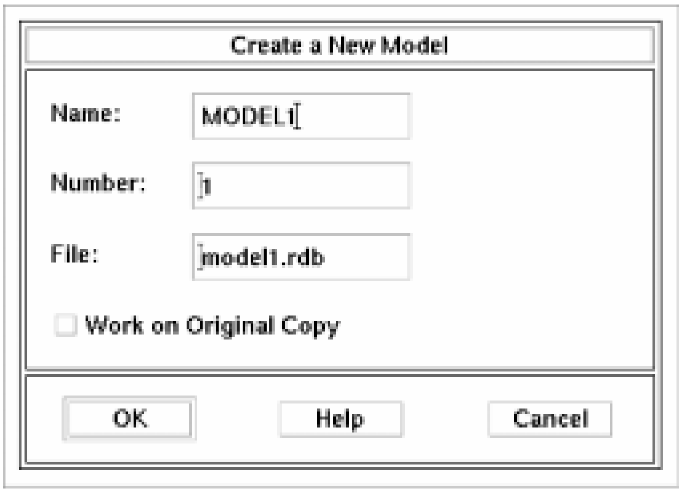
Finally, as is customary in GUI–driven programs, the **Exit** command (see Section 4.11 on Page 4–25) is listed at the end of the File menu.

The most important commands of the File menu are also available as buttons (with icon labels) in a Button Window (see Section 4.5 on Page 4–11) which appears when RAYNOISE is launched.

5.2 Working with database files

5.2.1 New database file

A new model and the corresponding database can be created from the **File, New** dialog:



Create a New Model

Name: MODEL1

Number: 1

File: model1.rdb

□ Work on Original Copy

Figure 5.2–1 File, New dialog

A model is defined by the following elements:

- a model name;
- a model number;
- a file name (generally with the extension .rdb);

When the command is executed, the file _*name* is created in the working directory (where *name* is the full name entered in the dialog, including the extension). The underscore sign (_) indicates that the database file is currently open. If the file _*name* already exists, RAYNOISE will add another underscore (_) until an available file name is found (____*name* for instance).

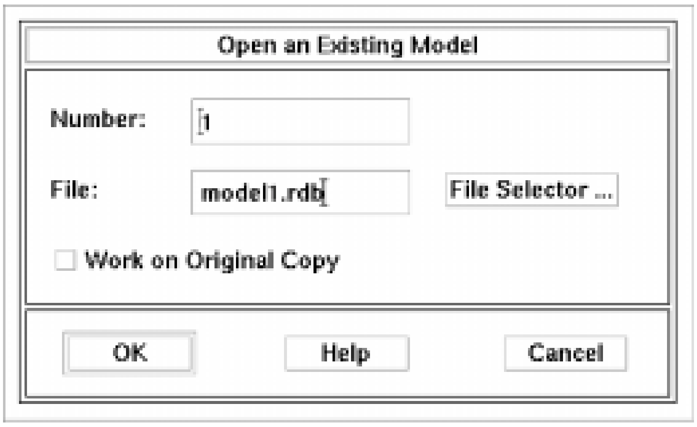
Several new models can be created successively with different names and numbers, and can remain open in one session of RAYNOISE. One can switch from one model to the other with the **File, Activate** command. See also the following commands:

- **Close**
- **Save**

- Save As

5.2.2 Open an existing database file

An existing model can be opened with the **File, Open** dialog:


Open an Existing Model

Number: 1

File: model1.rdb

□ Work on Original Copy

Figure 5.2–2 File, Open dialog

In standard mode, the model database (*model.rdb* for instance) is then copied to the temporary file *_model.rdb*.

If the switch **Work on original** is activated, this physical copy does not occur and work is done directly on the original copy.

Note! The *Work on original* switch must be used with special care, as loss of data may occur. In this case, it is not possible to Close the model without saving changes and restore its original state. An unexpected termination of the program, *eg* due to power loss or computer failure, could cause irrecoverable corruption or loss of data. The main reason for using 'Work on original copy' is if the database file is very large, so that the file system of the computer has insufficient space to hold the original file and the temporary file *_model.rdb*.

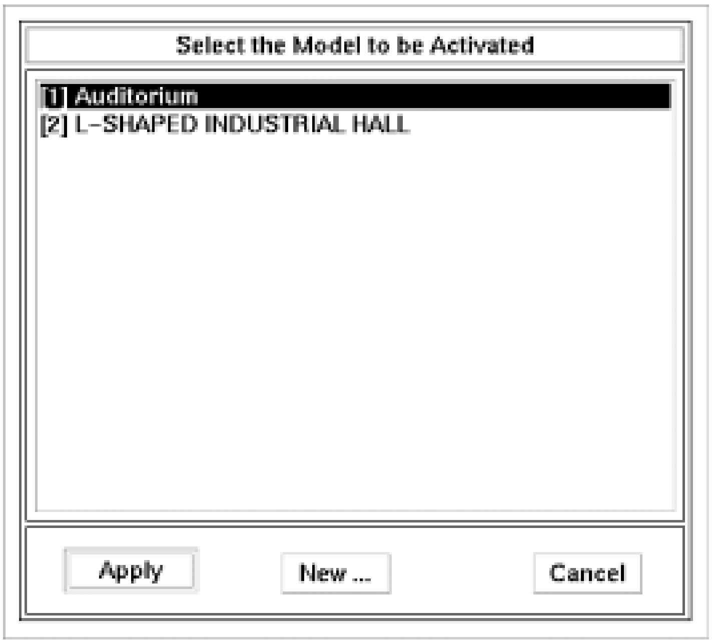
Several models can be opened successively. One can switch from one model to the other with the Activate command.

See also the following commands:

- Close
- Save
- Save As

5.2.3 Activate an open model

Several models can be simultaneously opened, but at any time only one model is active. By default, the active model is the last model created or opened. One can activate any open model using the **File, Activate** dialog which lists all currently−open models:



Select the Model to be Activated

[1] Auditorium
[2] L−SHAPED INDUSTRIAL HALL

Figure 5.2–3 File, Activate dialog

Modifications can only be made to the active model. All RAYNOISE commands are applied to the active model and only to that model, with the exception of display−related commands (**Display**, **View**, **Postprocess** menus). By default all open models are visible and are displayed in all viewports.

The results from any open model can be selected and displayed by the post−processor regardless of which model is currently active. (But note that if the model, or more precisely the object in the model on which results are displayed, like the field−point mesh, is not visible in the active viewport, then a request to display results will not make the object visible and there will be no apparent action).

5.2.4 Database operations

5.2.4.1 Dialog

Several operations can be performed on the <u>active</u> database with the **File, Database** dialog:

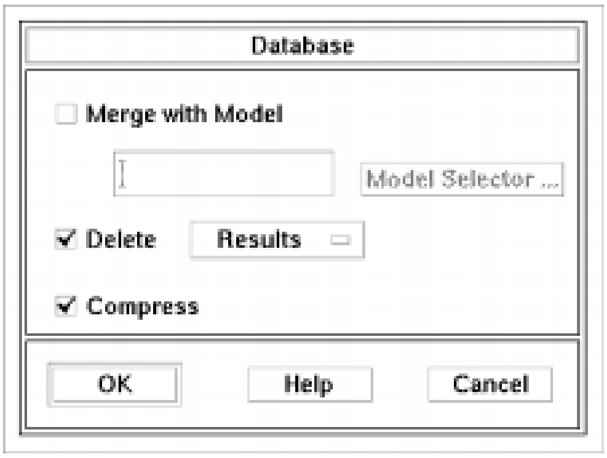


Figure 5.2–4 File, Database dialog

The main purpose of these operations is database management, often for cleaning or space–saving reasons.

5.2.4.2 Merge

Merge enables the import of results from another open model. This is typically of use if two copies of the same model have been processed separately (for instance, to reduce the elapsed time to calculate several frequencies with frequency–dependent calculation processes such as diffusion) and the results are to be combined.

For example, if Model 1 has results at frequencies 63 and 125 Hz and Model 2 has results at frequencies 250 and 500 Hz, the command: **File, Activate**, *Model 2*, followed by: **File, Database, Merge** *Model 1* (which are addressed though the **File** menu) will merge the results of Model 1 into Model 2, so that Model 2 contains results at 63, 125, 250 and 500 Hz.

Model 1 and Model 2 must be compatible, *ie* they must have identical meshes and field point meshes.

5.2.4.3 Delete

Delete enables the removal of certain data types from the database. For instance, **File, Database, Delete, Result** will delete all results from the (temporary) database. The option menu offers a choice between Points, Echograms, Results or Tables. The user should take care to delete database entries in a logical manner (*eg*, deleting Results before deleting Points: in this case, it may be preferable to use the Geometry, Field Point, Reset command).

5.2.4.4 Compress

Deleting data from the database puts the selected data values to zero, without deallocating the unused space. **File, Database, Compress** will deallocate the space and reduce the size of the database file. During the compression process, records are copied to a temporary database using a buffer. The size of this buffer may need to be increased using the Environment Variable BUFFERSIZE (see Section 11.8 on Page 11–19) if errors occur.

5.2.5 Close a database file

The database file for the active model can be closed using the **File, Close** menu. If this model was not previously saved after making some changes, RAYNOISE will ask whether the model changes should be saved or not.

Hint: Closing without saving, followed by re–Opening, allows a general 'abort' of a series of changes. Thus, it can be useful to Save the model periodically, after each set of correct changes.

When Closing a model, it is recommended to close any display–related dialogs (such as results display) and re–open them when required, to ensure consistent references to the remaining open model(s).

5.2.6 Save a model and 'Save As...'

The active model can be saved at any time. The **File, Save** operation consists of copying the temporary database (_*name*.rdb) to its original location (*name.rdb*) therefore overwriting the previous file.

With the **File, Save As...** dialog, a new name can be given to the database (_*name.rdb* is copied to *newname.rdb*).

A model must be made active before being saved.

Saving the model does not close the model, which remains open and active.

Hint: It is recommended to save the model regularly while working, to avoid any loss of data due to hardware or software problems. This is a manual operation to be done by the user.

5.3 Import and Export data

5.3.1 General import/export principles

RAYNOISE is an open program able to communicate with other components of a CAE environment: CAD, structural FEA pre− and post−processors and the like:

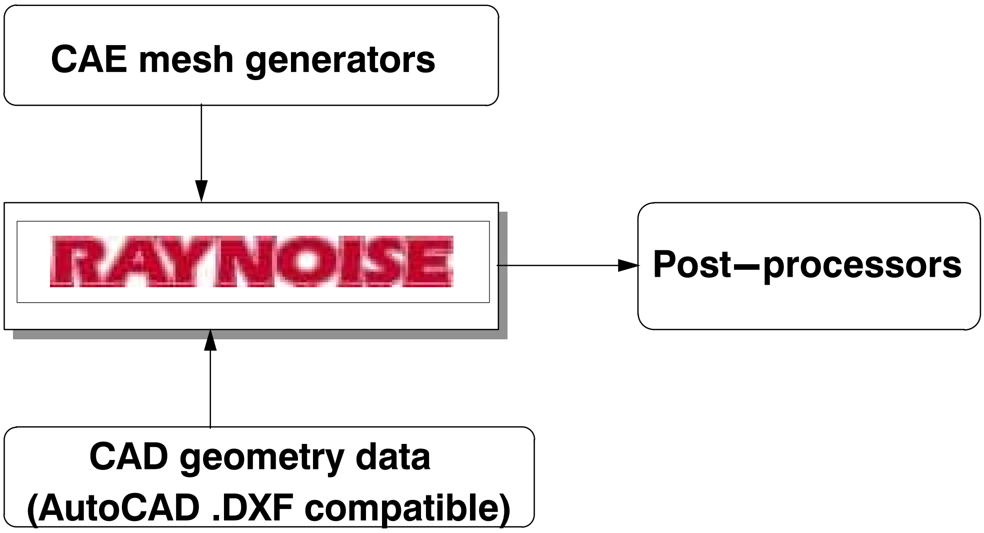
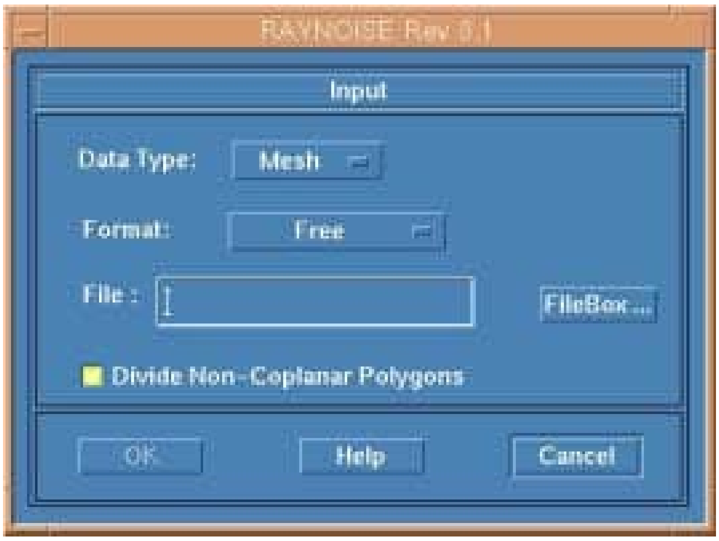


Figure 5.3–1   Data exchange with pre–/post–processors and CAD

The **File, Import** and **File, Export** menu are used to read and write data using the formats of other programs:



Data Type: Mesh

Format: Free

File :

■ Divide Non-Coplanar Polygons

Figure 5.3–2   File, Import dialog

The dialog requires three items of information:

- what do you want to read ?
- in which format is it written ?
- in which file is it available ?

5.3.2 Supported data types for Import and Export

The following data types can be Imported and Exported from RAYNOISE, as explained in the following sections:

- Model database
- Data File
- Mesh
- Points
- Sets
- Results
- Tables

5.3.3 Model database

The complete contents of the database can be Output to an ASCII file, for transfer to another computer and similar purposes. This is done using **File, Database To...** giving a filename (including arbitrary extension) in the dialog.

File, Database From... causes the entire contents of the given filename to be decoded and translated into the internal RAYNOISE database in the reverse of this process.

Thus, it is possible to convert the (binary) contents of the RAYNOISE database into the format of another computer which may not be binary compatible. This is especially useful for conversions from PCs to Unix machines and *vice versa*.

Note that any data existing in the active database will be over–written by the data coming–in from the Database From... file, so normally a New database should be opened before starting the database import.

5.3.4 Data file

It is possible to dump the complete data for the model into a RAYNOISE Free format ASCII file. The definition of this file is very extensive: please contact LMS (via your local support representative, if applicable) if you want to request this information. For data transfers between computers (*eg*, PC to Unix...) it is normally recommended to use the **File, DatabaseTo/From...** commands. (See Section 5.3.3 on Page 5–10).

5.3.5 Import and Export of a mesh

5.3.5.1 Procedure

A mesh must be three–dimensional (*ie* three coordinates per node) although it is acceptable to have a two–dimensional geometry (*ie* all elements may be co–planar).

The data type **Mesh** is selected for reading or writing a geometry model (acoustic mesh).

A mesh is a discretized representation of a physical structure. RAYNOISE considers this structure as a closed or open set of physical boundaries. The mesh information needed by RAYNOISE consists of node coordinates and element connectivities. Elements are two–dimensional polygons, representing geometrical surfaces.

The mesh file contains a list of nodes with their coordinates and a list of elements with their topologies (lists of nodes defining the elements). On **Import**, all other data contained in the file (material properties, layer information, etc) are ignored. If any such data are required, they must be defined subsequently in RAYNOISE (Note that specific capabilities exist for the transfer of Layer information from AutoCAD compatible files, see **Import, Sets** in Section 5.3.7 on Page 5–14). Upon **Export**, only the node coordinates and the element topologies are written and all other RAYNOISE information is ignored.

5.3.5.2 Mesh verification

The import of a mesh from an external file is immediately and automatically followed by a mesh verification, including a check of:

- Superimposed nodes
- Isolated nodes
- Coplanarity of the nodes of the elements
- Colinearity of three subsequent nodes of an element
- Degenerated elements
- Superimposed elements

- Calculation of the normal vector on each element.

There is no pre–requisite regarding the element normals. The local node numbering (which controls the normal, using a 'right–handed screw rule') may be oriented clockwise or counter–clockwise, but must be sequential around the perimeter of the element. The normal of an element is only important when defining a panel source, which must be attached to one side or the other side of an element.

Messages warn the user of detected anomalies, but do not suppress further analysis capabilities. Mesh geometry may be deleted, added or modified, as described in *Mesh editing*, Section 6.2 on Page 6–5.

The particular case of non–coplanarity of the nodes of a polygon with four or more nodes is handled as follows:

- If the toggle button *Divide Non Coplanar Polygons* on the **File, Import, Mesh** dialog is *Off*, and a polygon is found with the fourth or later node not lying on the plane defined by its first three nodes, that node is forced onto the plane by a projection normal to the plane; this is repeated for each element according to their internal numbering (that is, sequence in the mesh file) – which may result in erratic results in case of successive coplanarity corrections.
- If the toggle button *Divide Non Coplanar Polygons* on the **File, Import, Mesh** dialog is *On*, such on–coplanar polygon is divided into triangles, which are implicitly co–planar, and there are no changes to any node coordinates.

Note! Reading in a new mesh will initialize the model and clear the temporary database before loading the new geometry.

For related information on mesh modifications, see:

- *Changing node coordinates* (Section 6.2.2 on Page 6–5)
- *Adding nodes to the mesh* (Section 6.2.3 on Page 6–5)
- *Adding elements to the mesh* (Section 6.2.4 on Page 6–6)
- *Deleting elements from the mesh* (Section 6.2.5 on Page 6–6)
- *Reversing element normals* (Section 6.2.6 on Page 6–7)
- *Geometrical tolerances* (Section 6.2.8 on Page 6–9)

5.3.6 Import and Export of a field point mesh

5.3.6.1 Description of a field point mesh

Field points do not belong to the geometry model mesh, but are points at which acoustic results are evaluated. (They may be referred to as 'microphone positions' or 'receivers').

The field point mesh is a mesh of elements linking the field points together. The elements can be quadrilaterals, triangles or lines (bars). It is also possible to have isolated field points, but they are not recommended because of the limited possibilities for visualizing the results at such points. The field point mesh information needed by RAYNOISE consists of point coordinates and element connectivities.

5.3.6.2 Procedure

The data type **Point** is selected for reading or writing a field point mesh, in the **File, Import** dialog. The **Geometry, Field Point** dialog can also be used to Import points, with the complement **Mesh...** This has the same effect as File, Import with data type Point, except for the possibility to suppress merging of coincident field points (see remarks on field point numbers, below).

The field point mesh file contains data which is the same as a list of nodes (points) with their coordinates and a list of elements with their topologies (lists of nodes defining the elements). On **Import**, all other data contained in the file (material properties, layer information, etc) are ignored. Upon **Export**, only the point coordinates and the field−point element topologies are written.

Hint: It can be useful to Export a field point mesh created by using RAYNOISE tools, to avoid the need to have to repeat a sequence of several FPM–generation commands. Later, the FPM can be Imported in one step.

5.3.6.3 Field point numbers

Field points generated using the RAYNOISE tools receive field point numbers which are identical internal and external numbers, in ascending order depending on the order in which they are defined. Field points imported from mesh files receive internal numbers in the same manner, but but retain their external numbers from the interface file. If a field point already exists in the model with the same external number as one read from the interface file, the new point is assigned the next−highest unused number.

If more than one interface file is read, more field points will be created, with incrementing numbers. Coincident field points will be suppressed in such a situation (and the corresponding field−point element topologies updated) unless the **NoMerge** button is toggled *On* in the **Geometry, Field Points, Mesh** dialog. This toggle does not exist in the **File, Import** dialog. Usually there is no value in retaining coincident field points, since their results will always be identical.

- See *Field point mesh*, Section 6.3 on Page 6−11, for more general information on creating and using field point meshes in RAYNOISE.

- See *Importing a field point mesh from another program*, Section 6.3.9 on Page 6–20, for details of the Geometry, Field Points, Mesh dialog.

5.3.7 Import and Export of sets

5.3.7.1 General procedure

It is only possible to Export sets, with the exception of sets of elements, which can also be Imported from a DXF–compatible file if groups of elements (surfaces) are defined there using layers (AutoCAD term).

Generally, four types of sets may be Exported:

- sets of nodes
- sets of elements (including 'layers' in RAYNOISE terminology)
- sets of faces
- sets of points (field points)

All geometrical items (nodes, elements, faces, points) are written out in accordance with the chosen file Format (typically, RAYNOISE Free format, producing an ASCII readable file). Files with element–like items (*ie* elements or faces or field–point meshes) may be re–Imported as if they contained a mesh.

Open the **File, Export** dialog and select the type and the entity to be exported, together with the Format and the filename of the resulting file: an existing file can be selected (in which case it will be overwritten) or a new file can be created (in which case it will be created in the current working directory or using the specified path).

5.3.7.2 Importing sets of elements from AutoCAD layers

It is possible to import sets automatically if they exist as layer definitions (using the AutoCAD terminology) in a DXF–format file. In this case, use the **File, Import** dialog and select **Sets**, giving the relevant filename and the selecting DXF format.

The layer definitions will be read from the DXF file and sets of elements will be created, using set Names automatically–generated from the layer names in the DXF file and set Numbers increasing from the number used last before the import of the layers.

Hint: Automatic generation of sets of elements from layers in the DXF file can be especially useful if the geometry–modeling work which produces the DXF file can be co–ordinated with the requirements of RAYNOISE – for instance, as regards sets of elements for the assignment of material properties.

- See *Importing DXF entities into RAYNOISE*, *Material assignment using layers*, Section 14.2.5.4 on Page 14–10, for more information.

5.3.7.3 Exporting sets

All nodes and elements contained within the specified set are written by this command in a formatted file (typically, using the RAYNOISE Free format) as a complete mesh. It is only possible to export sets containing geometrical entities (elements, nodes, faces, edges). Sets of sources cannot be exported.

- See *Sets*, Section 6.5 on Page 6–27, for more information on creating and using sets in RAYNOISE.

5.3.8 Export of results

5.3.8.1 Types of results

Results correspond to acoustic quantities at the points of the field–point mesh. There are two types of acoustic quantities: those that are dependent on frequency and those that are independent of frequency.

5.3.8.2 Quantities dependent on frequency

- Complex Pressure (Pa)
- SPL (dB)
- SPL (dB/A)
- Direct Energy (dB)
- Total–to–Direct Energy–Ratio (dB)
- Definition (%)
- STI (–)
- EC (Speech) (–)
- ERR (–)
- Clarity (dB)
- Lateral Efficiency (%)

- Time Central Gravity (ms)
- EC (Music) (–)
- RT60 (s)
- RT30 (s)
- RT20 (s)
- EDT (s)

5.3.8.3 Quantities independent of frequency

- SPLWide (dB)
- SPLWide (dB/A)
- NC (dB)
- NR (dB)
- RASTI (–)

5.3.8.4 Exporting results

Results files contain acoustic results at the nodes of a field point mesh. These acoustic results may be:

- pressure values
- acoustic quality parameters
- speech interference parameters
- auralization files

Results are written to an external file by selecting the Data Type *Results* in the **File, Export** dialog. The external file will contain all available frequency–independent result values, as well as frequency–dependent results values at a certain frequency.

Before the output of results, the frequency must be chosen and the results for that frequency loaded from the database into memory, using the command:

```
RAYNOISE> SEARCH RESULTS FREQUENCY  fff  RETURN
```

where *fff* is the required frequency (real number). The command must be entered at the RAYNOISE> prompt. (There is no GUI equivalent).

Acoustic results are written to interface files using appropriate formats (such as the RAYNOISE Free format, ASCII file). See *Supported interface formats*, Section 5.3.10 on Page 5–17, for more information.

Results can also be read from an external file into a model database using the Data Type *Results* in the **Import** dialog. See *To transfer results from RAYNOISE Revision 2.1A*, Section 17.3 on Page 17–9.

5.3.9 Import and Export of Tables

Tables can be Imported and Exported using RAYNOISE Free format, for any table Type. Tables can be Imported using CADA–X and I–DEAS Universal files, for tables of General type only. When Importing, the file can contain multiple tables (of the same Type) separated by identification fields. Tables are then created in RAYNOISE, with ascending numerical order and table names derived from these identification data.

Tables of any Type can also be created using the Table command, with the **Model, Table, From file...** menu entry. In this case, the file must contain the data for one table only, in the appropriate 'free' format. See *Reading a table from a file*, Section 7.7.3 on Page 7–32, for more information.

Tables are typically frequency–dependent data: see *Combine*, Section 11.4 on Page 11–10, for further information on tools to manipulate these data and change formats.

Not all data types are supported by all interfaces. *Supported interface formats*, Section 5.3.10 on Page 5–17, defines the features implemented in each interface.

5.3.10 Supported interface formats

RAYNOISE Rev 3.1 supports the following formats:

ABAQUS	formatted data files
ANSYS	*file12* and *file28* of ANSYS Rev 4.4A and *file.rst* and *file.cdb* of ANSYS Rev 5.2 and later.
AUTOCAD	DXF files: geometry data; layers (to make RAYNOISE Sets)
CADA–X	universal file
FEMGV	*file.anl*
HYPERMESH	Version 1.30f
I–DEAS	universal file

MEF–MOSAIC	COR, ELE and RES files
MSC/NASTRAN	bulk data deck, punched files and OUTPUT2 files
MSC/PATRAN	neutral file and nodal data results files
RASNA	h–mesh file, displacements and rotations files
SYSTUS	interface file

When several revisions are supported (NASTRAN, ANSYS, I–DEAS), the interface library is selected by setting the appropriate environment variable (IDEASREVISION, ANSYSREVISION, NASTRANREVISION, NASTRANFILE).

The available interfaces are installation–dependent. The corresponding menu items are eliminated if the interface is not installed.

Hint: Besides the standard interfaces, information can also be exchanged through user–defined routines in order to fit any specific user environment. Please contact LMS or your local support representative for information on these user–defined interface formats.

5.4 Space and memory requirements

5.4.1 Memory allocation

The initial allocation of (maximum) memory depends on the installation: it is also set by the complement –*mX* of the command which launches RAYNOISE, where *X* is the allocated memory in Mwords (*ie*, allocated memory = 4*X* Mbytes). In a Windows installation, this parameter has to be set in the file **raynoise.bat** which launches RAYNOISE, usually found in the ...\bin directory.

Command example (Unix):

```
> /path/RAYNOISE/3.1/bin/raynoise -m6
```

which allocates 24 Mb of memory. *path* is installation–dependent.

It is recommended not to allocate more memory than the physical memory available on the computer, allowing some margin for system functions. However, if more memory is allocated, RAYNOISE will use system swap space (virtual memory).

5.4.2 Memory requirements

RAYNOISE allocates memory dynamically during execution, depending on the model data and the procedure being executed.

In general, the more computationally–intensive tasks such as beam–tracing (*Mapping calculation*, see Section 9.3 on Page 9–7) and in particular *Results animation* (see Section 10.2.6 on Page 10–25) will demand more memory.

The number of receivers (field points) combined with the requested refinement of the results (histogram parameters, *etc*) determines the total volume of data arrays and hence the memory required. If insufficient memory is available, the program will give a warning message.

5.4.3 Disk space

Disk space is required for the temporary database file (_*name*.rdb) and the permanent copy, together with log and other run–time files. If disk space is in short supply, the user should beware of the danger of creating a large Logfile if large volumes of data extraction or a high *Debug level* are requested (see Section 11.2.2 on Page 11–4). The disk space required for the temporary copy of the model database can be saved by using the **Work On Original Copy** option of the **File,**

Open dialog, but the user must beware of the serious danger of irrecoverable corruption of the complete model database in the event of an error or computer failure. (See *Open an existing database file*, Section 5.2.2 on Page 5−5, and Database, Section on , for more information).

6

Geometry modeling

This chapter describes the geometric modeling capabilities of RAYNOISE. Complex geometries are imported from external programs (CAD or mesh generators) but RAYNOISE offers further capabilities for manipulating the geometry, including:

6.1 Numbering schemes

6.1.1 General principles

RAYNOISE uses a dual numbering scheme for most items, namely an external number (defined by the user, generated by the CAD program or mesh generator used for geometry modeling, etc) and an internal number, used by the data handling of RAYNOISE, generated in a continuous ascending order. Usually, it is not necessary for the user to address the internal numbers or have any particular concern for them. If it is desired to address an item by its internal number, this can be done by adding the label *Internal* after the relevant item (node, element, source,...) number.

Where a program input requests a number to be inserted, but allows a blank entry, the program defaults to the last–used number plus one.

The maximum size of a number is ten digits.

6.1.2 Node and element numbers

Node and element numbers are generally inherited from the data imported from an external file. The external numbers of nodes and elements are completely arbitrary: the user may therefore employ sequences which assist in identifying the locations of the items. Gaps are are permitted in the sequences.

6.1.3 Numbering of other data items

6.1.3.1 Materials

Materials get default external numbers which are the same as their internal numbers, or they can be given arbitrary external numbers when the material is defined.

Note! The internal or external number of a material relates to its use in the active model and does not relate to the number defining its position (if any) in the materials database.

6.1.3.2 Sources

Sources get default external numbers which are the same as their internal numbers, or they can be given arbitrary external numbers when the source is defined.

Note! The internal or external number of a source relates to its use in the active model and does not relate to the number defining its position (if any) in the sources database.

6.1.3.3 Field points

Field points always get identical external and internal numbers when they are created with the tools in RAYNOISE. Their numbers cannot be altered inside RAYNOISE. The numbering of field points may change if the field point mesh is deleted and re−created in a different way: this may affect the selection of a point for results presentation. To avoid such possible problems, field points can be selected for results presentation by using graphical tools or geometrical references (Point Near, Point Between...).

6.2 Mesh editing

6.2.1 General principles

The geometry model used in RAYNOISE is usually imported from an external file, created by another program (AutoCAD, mesh generator, ..., see *Import and Export of a mesh*, Section 5.3.5 on Page 5–11). However, it may be necessary to edit this mesh, for several possible reasons:

- to adjust node coordinates
- to add nodes and/or elements
- to delete elements
- to add a Field Point Mesh

RAYNOISE allows all these possibilities to be done interactively. It is also possible to create the complete geometry model within RAYNOISE.

Hint: If the geometry is to be created within RAYNOISE, it can be useful to set up the data in a RAYNOISE command file which can be edited. This can be read into RAYNOISE much faster than typing commands or accessing dialogs. See Reading a command file, Section 4.6.4 on Page 4–14, for further information.

If isolated nodes exist in the model, these are ignored during all calculation steps.

Hint: To suppress isolated nodes from the data, all elements of the mesh can be placed in a Set, that set Exported to a file and then Imported as the mesh. The Export step will only include the nodes attached to the elements.

6.2.2 Changing node coordinates

If the **Geometry, Mesh, Nodes...** dialog is used and an external node number is entered which already exists, that existing node will be moved to the newly–entered coordinates. If the complete mesh needs to be re–sized, use the **Geometry, Scale** dialog (see Section 6.2.7on Page 6–8).

6.2.3 Adding nodes to the mesh

Nodes can be added to the mesh using the **Geometry, Mesh, Nodes...** dialog.

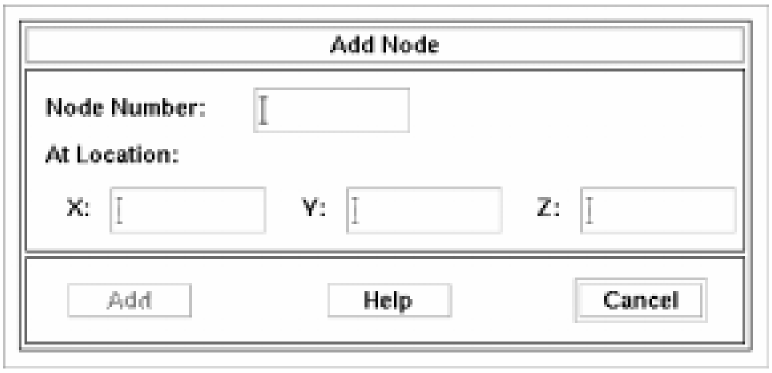


Figure 6.2–1 Geometry, Mesh, Nodes dialog

Each node is added in turn and is created by entering its external (user–defined) number and the coordinates. The *Apply* button causes the node to be created and added to the mesh. New nodes will be isolated until referenced in the definition of a new element.

6.2.4 Adding elements to the mesh

Elements can be added to the mesh using the **Geometry, Mesh, Elements...** dialog.

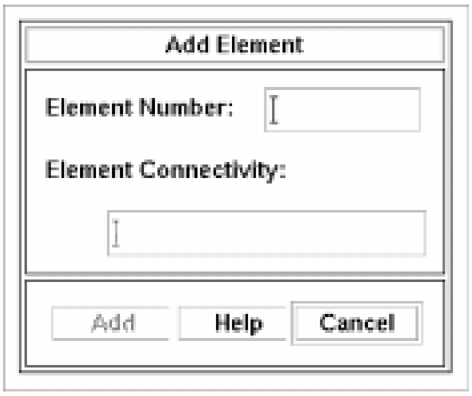


Figure 6.2–2 Geometry, Mesh, Elements dialog

The external (user–defined) element number is entered, together with a list of node numbers to which it is connected. These must follow around the vertices of the element (polygon) in sequence. (Note also that the element normal direction is determined by the first three nodes in this sequence). Commas or blanks are valid separators. The*Apply* button causes the element to be created and added to the mesh.

6.2.5 Deleting elements from the mesh

To delete an element from the mesh, use the **Geometry, Mesh, Elements...** dialog. Enter the number of the existing element to be deleted, and leave the node connectivity field blank. The element will then be deleted, when *Apply* is pressed.

6.2.6 Reversing element orientations

6.2.6.1 Displaying element normals

The element normal vectors can be displayed by using the button on the toolbar, which is equivalent to selecting the Show Element Normal button in the **Display, Graphical Options, Object** dialog. (See Section 12.1.4.10 on Page 12–8). Figure 6.2–3 shows the element normal vectors on a cross–sectional (side) view of a three–dimensional mesh.

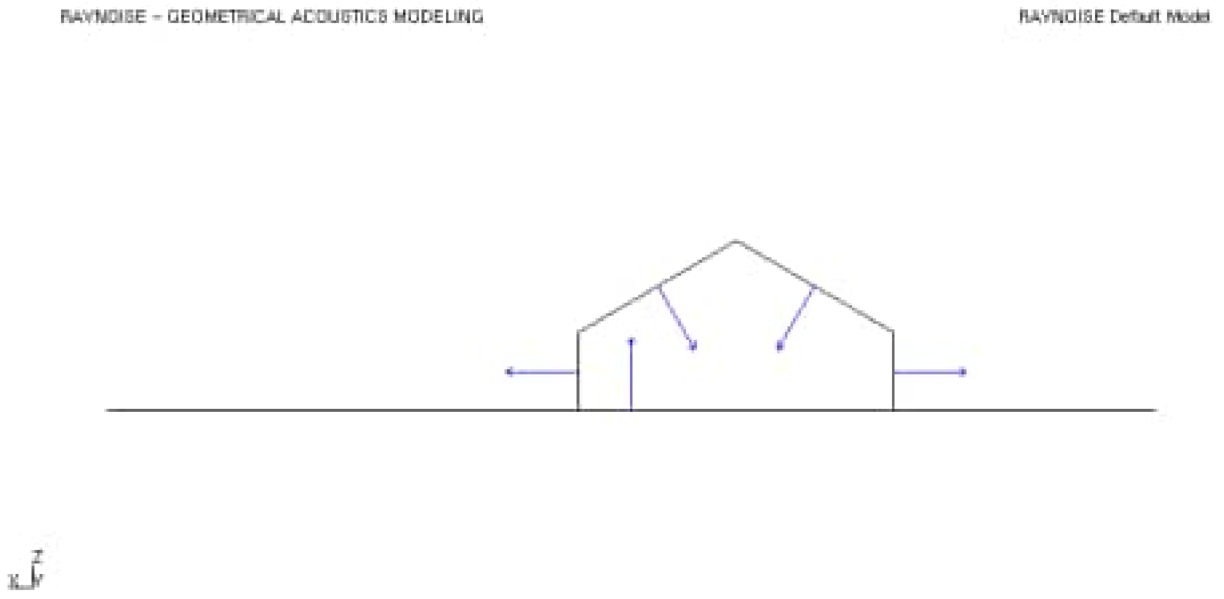


Figure 6.2–3 Model with element normal vectors shown.

6.2.6.2 Element normal definition

The direction of the normal to a 3D surface element is defined as the normalized cross product between the vector joining the first corner node ('A' in Figure 6.2–4) to the second corner node (B) and the vector joining the first corner node to the third corner node (C).

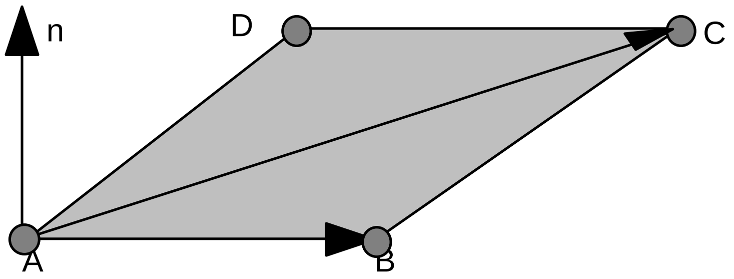


Figure 6.2–4 Normal vector to a 3D element

6.2.6.3 Consistency requirements

It is generally not necessary to have element normals oriented consistently, since material properties are attached to elements without regard to any concept of positive or negative sides, or orientation. However, in the case of a ‘panel source’ associated with an element, this *is* attached to one specific side, positive or negative, determined in the model in accordance with the element normal. It may therefore be helpful to be able to reverse certain elements in order to ensure (for example) that all sources are defined on the positive sides of elements. This is a matter of convenience or the user’s own convention, however.

Similarly, it is desirable to have all the normals of field−point elements oriented in the same direction, so that certain sorts of result presentations (deformation of the field−point mesh, vectors, ...) give consistent results. If the FPM is made with RAYNOISE geometry tools (plane, box, cylinder, ...) it should be inherently consistent, if the geometric definition is carried out in a consistent, logical manner.

Hint: Elements in a field–point mesh can be reversed by Importing the FPM as if it was a mesh and reversing elements, then Exporting it again to a new file, before using it as a FPM. Take care to Save the ‘real’ model before starting such a process, since Importing a mesh clears the temporary database.

The command REVERSE allows the user to reverse the orientation of some elements, for example:

```
RAYNOISE>  REVERSE ELEMENTS BETWEEN Z=2.9 3.1 RETURN
```

or to reverse the orientation of all elements, for example:

```
RAYNOISE>  REVERSE ELEMENTS ALL RETURN
```

The REVERSE command is activated from the **Geometry, Mesh Modification, Reverse Elements...** dialog.

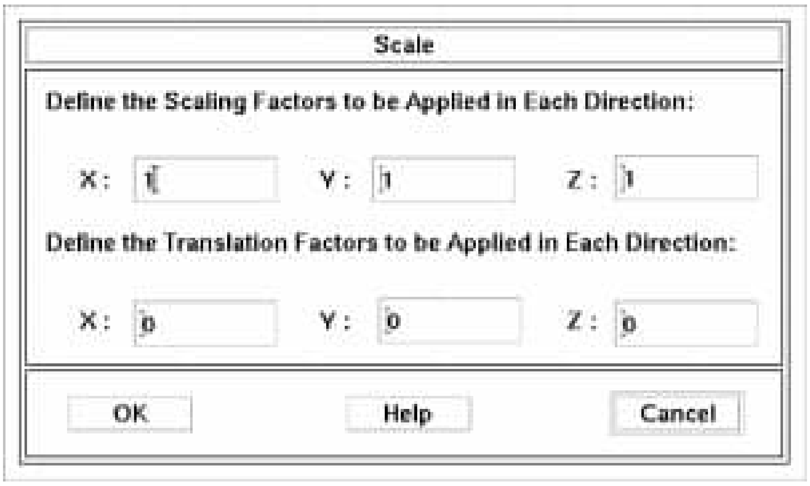
Advanced users or those with experience of the SYSNOISE program can use the command CHECK MESH, to reverse any elements which ought to be reversed and to achieve maximum consistency, under automatic control from the program. The command is only available in command−language mode, not via GUI menus:

```
RAYNOISE>  CHECK MESH NOSET RETURN
```

6.2.7 Scaling the mesh

6.2.7.1 Function

The mesh and field point mesh currently in the RAYNOISE database may be scaled and translated using the **Geometry, Scale** dialog:



Scale

Define the Scaling Factors to be Applied in Each Direction:

X: 1 Y: 1 Z: 1

Define the Translation Factors to be Applied in Each Direction:

X: 0 Y: 0 Z: 0

Figure 6.2–5 Geometry, Scale dialog

The scaling is defined by three factors (α_x, α_y, α_z) and the translation by a vector (t_x, t_y, t_z). The coordinates (x, y, z) of a node or a field point are modified by the following transformation:

$$x_{new} = \alpha_x \cdot x_{old} + t_x$$
$$y_{new} = \alpha_y \cdot y_{old} + t_y$$
$$z_{new} = \alpha_z \cdot z_{old} + t_z$$

This command is useful for instance to change the units of a model: for related information, see *Units in RAYNOISE*, Section 7.8 on Page 7–35.

6.2.7.2 RCL syntax of the SCALE command

The SCALE command is:

```
RAYNOISE>  SCALE ax ay az TRANSLATE tx ty tz RETURN
```

If only one factor is given, it fixes the value of the two others. If two factors are given, the third is equal to the second. For instance, the two following commands are identical:

```
RAYNOISE> SCALE 1000 1000 1000 RETURN
RAYNOISE> SCALE 1000 RETURN
```

6.2.8 Geometrical tolerances

6.2.8.1 General

The coincidence of nodes, points and other geometrical checks make use of tolerances which are factors of the model dimensions. The factors are user–controlled, by the Environment Variable TOLERANCE (see Section 11.8 on Page 11–19).

The default value of TOLERANCE is $1.0E^{-3}$.

6.2.8.2 Tolerance for node coincidence

The coincidence of nodes and other points (*eg* field points) is checked using:

TOLER = TOLERANCE * SIZMX

Two nodes are considered to be coincident if their separation is less than TOLER.

6.2.8.3 Tolerance for coplanarity

The nodes of an element are checked for coplanarity by calculating the normals for each of the planes defined by successive sets of three nodes around the element. If any three nodes are co−linear, the middle node is dropped and the next node added to the set. A node is considered to be non−coplanar if:

$\boldsymbol{n}_1 \,.\, \boldsymbol{n}_i <=$ TOLE2

where $\boldsymbol{n}_i$ is the normal to the set of nodes i, i+1, i+2, and:

TOLE2 = $(1 - \text{TOLER})^3$

During the verification after the Input of a mesh (see *Import and Export of a Mesh*, Section on Page 5−11) non−coplanar elements are identified, with a warning message indicating which node(s) are outside the tolerance. The user may be asked to accept the mesh, if large errors are found: the nodes are forced into coplanarity.

6.2.8.4 Effect of keeping non–coplanar elements

If an element is non−coplanar, the first three nodes in accordance with the connectivity definition of the element are used to fix the normal to the element. Any other node, which does not lie on the plane formed by the first three nodes, will be mapped perpendicularly onto that plane. The boundaries of the element will then be fixed using the mapped node coordinates. This may affect whether a ray hits the element or not, during the Mapping calculation. In turn, this will also influence the number of 'lost' rays. If a Mapping calculation indicates a large number of lost rays, without any physical explanation such as an opening or exterior radiation, this is likely to indicate a geometrical error in the model, possibly due to the non−coplanarity described here. It will give inaccuracies in the results.

It is recommended to check the mesh geometry and if necessary correct the mesh and re−load it into RAYNOISE, or alter node coordinates inside RAYNOISE, if significant non−coplanarities are indicated. Alternatively, the non−coplanar elements can be split into triangles during the Import process, by using the toggle button on the Import dialog. In this case, *all* non−coplanar elements are split.

6.3 Field point mesh

6.3.1 Definition

Field points are the simulation equivalent of microphones and indicate the points where the sound field must be calculated.

Typically, many field points will be defined. For display purposes, it is convenient to join the field points by field elements. The set of field points and field elements is called the *field point mesh*. The field point mesh provides a set of points at which results will be calculated and thus a set of surfaces on which results can be displayed. It is sometimes referred to as a grid of receivers or microphones, but can be any arbitrary shape, with multiple surfaces, not just a simple grid.

The field point mesh (FPM) is quite transparent as regards the ray–tracing process and has no influence on the rays as they pass through it.

A field point mesh can be imported from any mesh generator or created in RAYNOISE using a variety of tools. Each tool can be used any number of times, in any order. Newly–created field points and elements are added to the existing field–point mesh, with their numbering incremented automatically, starting from the previous highest field–point and field–element numbers. The field point mesh is then the assembly of all field points, with their associated elements. If a newly–created and an existing field point have the same coordinates, they are merged unless the **No Merge** option is chosen (toggle *On*) in the dialog creating the new field points.

The complete field point mesh can be deleted using the **Geometry, Field Points, Reset** button.

Note! Deleting the field point mesh always removes the whole field point mesh. See *Deleting field points*, Section 6.3.12.2 on Page 6–24.

Hint: To avoid the need to re–construct a complex mesh created within RAYNOISE, it can be useful to save it in an external file (see Import and Export of a field point mesh, Section 5.3.6 on Page 5–12).

The following paragraphs give details regarding:

- Creating individual field points
- Creating a line of field points

- Creating a circular field point mesh
- Creating a planar field point mesh
- Creating a spherical field point mesh
- Creating a box–like field point mesh
- Creating a cylindrical field point mesh
- Importing a field point mesh from another program
- Reading individual field points from a file

Related information is given in:

- Displaying a field point mesh (see Section 8.1.2 on Page 8–3)
- Processing field points (see Section 9.4 on Page 9–10)
- Exporting a field point mesh to another program (see Section 5.3.6 on Page 5–12)

6.3.2 Creating individual field points

Individual field points are created within the **Geometry, Field Points, Points** dialog:

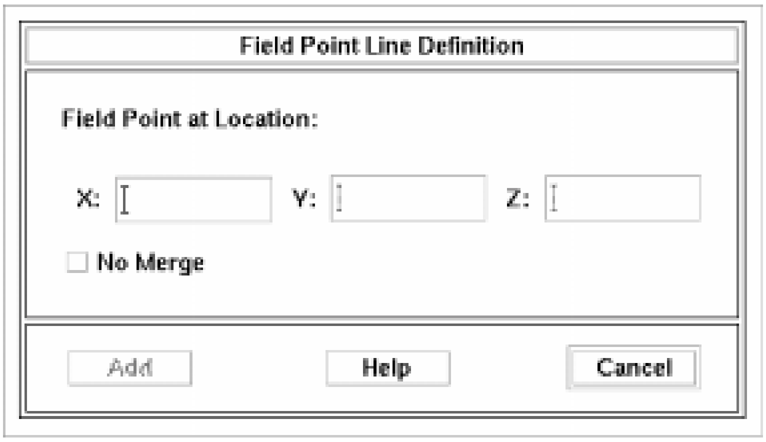


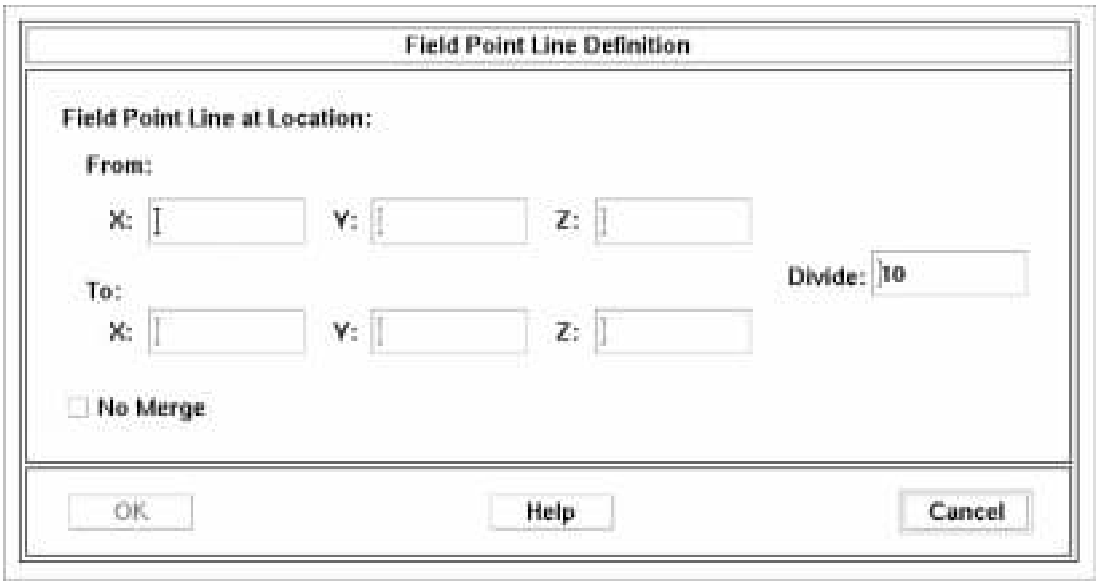
Figure 6.3–1 Geometry, Field Points, Point dialog

Single points are defined by their three coordinates and added to the current field point mesh using the **Add** button. Several points can be created before leaving the dialog with the **Cancel** button.

Hint: Field points created using this dialogue will be isolated, without connecting field elements. It is recommended to avoid isolated field points, since the contour map displays cannot show the results at such points. To view the locations of such points, select the 'highlight node' option in **Display, Graphical Options...** for the field–point mesh object, or turn on point numbering. (See Highlight Node, Section 12.1.4.8 on Page 12–8).

6.3.3 Creating a line of field points

Several field points along a straight line can be created in the **Geometry, Field Points, Line** dialog:



Field Point Line Definition

Field Point Line at Location:

From:

X: Y: Z:

Divide: 10

To:

X: Y: Z:

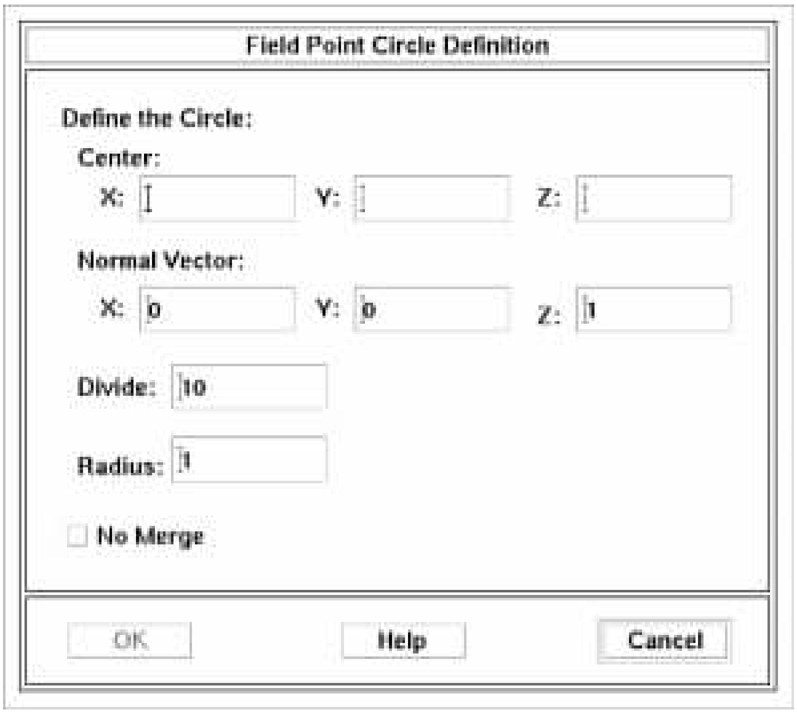
□ No Merge

Figure 6.3–2 Geometry, Field Points, Line dialog

The line is defined by its starting and ending points and by the number of divisions, *ie* the number of field elements, along its length. If n is the number of divisions specified in the *Divide* entry field, $n+1$ field points and n field elements (LINE2 type) are created.

6.3.4 Creating a circular field point mesh

A circle of field points can be defined by the **Geometry, Field Points, Circle** dialog:



Field Point Circle Definition

Define the Circle:

Center:

X: Y: Z:

Normal Vector:

X: 0 Y: 0 Z: 1

Divide: 10

Radius: 1

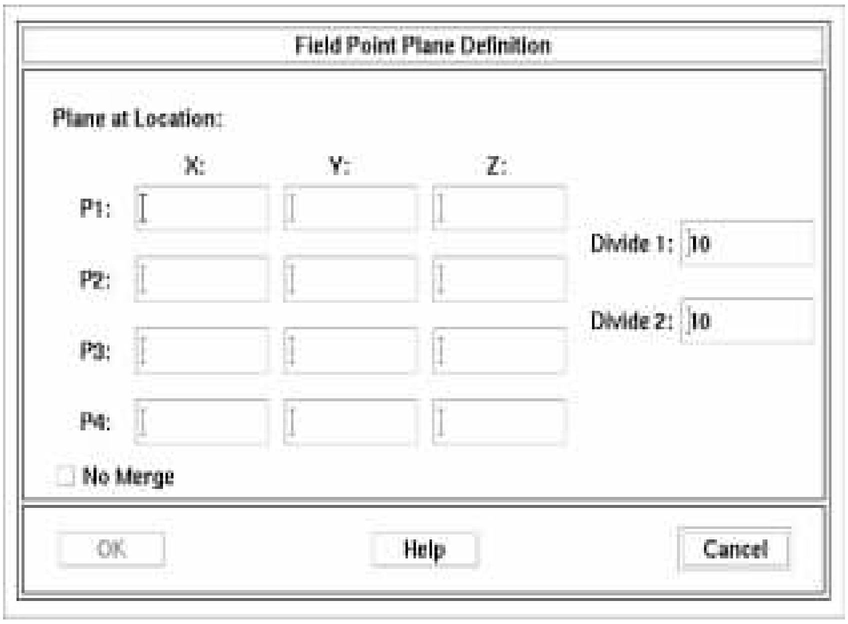
□ No Merge

Figure 6.3–3 Geometry, Field Points, Circle dialog

The circle has its center at the specified point. The plane in which the circle is located is defined by the components of its normal, in the *Normal Vector* entry fields, and the radius of the circle is defined by the *Radius* entry field. The field points are evenly spaced on the circle and separated by an angle of 360/*n*, where *n* is the value in the *Divide* entry field. *n* field elements (LINE2 type) and *n* field points are created.

6.3.5 Creating a plane field point mesh

A regular distribution of field points in a plane can be defined in the **Geometry, Field Points, Plane** dialog:



Field Point Plane Definition

Plane at Location:

	X:	Y:	Z:
P1:			
P2:			
P3:			
P4:			

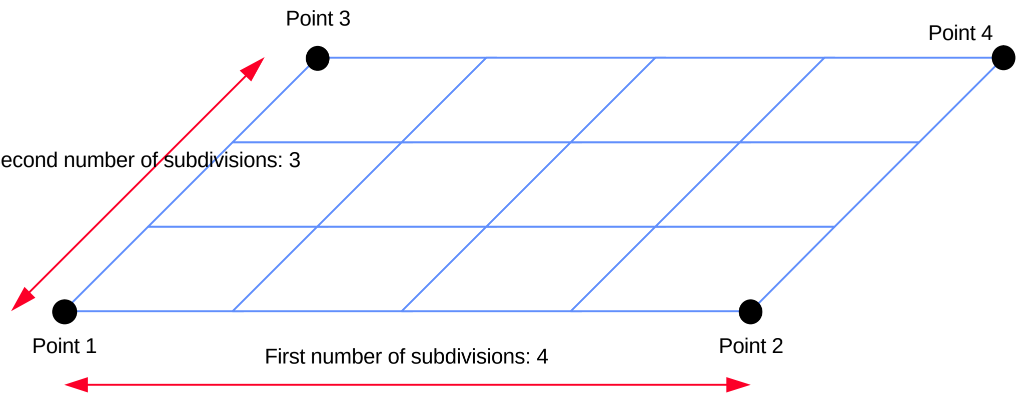
Divide 1: 10

Divide 2: 10

□ No Merge

Figure 6.3–4 Geometry, Field Points, Plane dialog

Four points are required to define a quadrilateral:



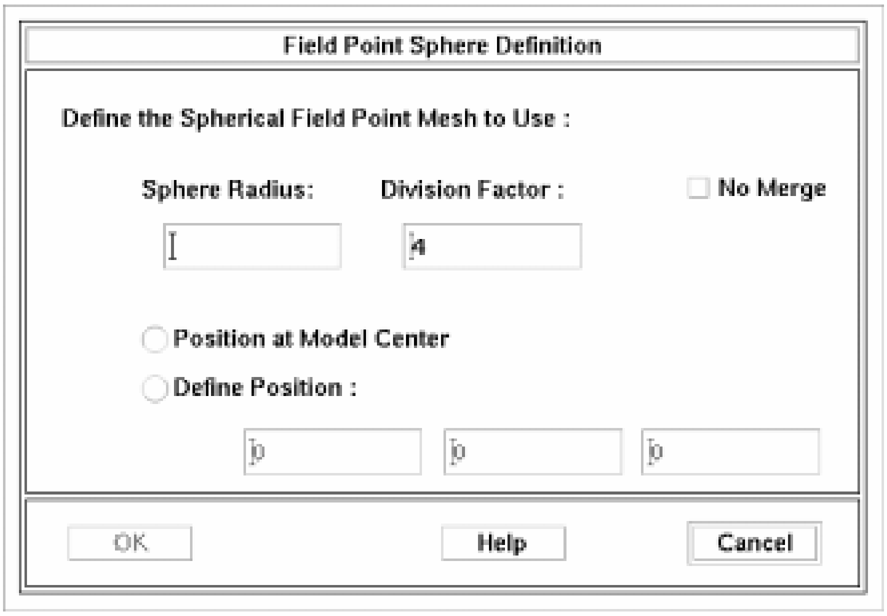
Point 3
Point 4
Second number of subdivisions: 3
Point 1
First number of subdivisions: 4
Point 2

Figure 6.3–5 Parameters for a field point plane

The positions of the four corner points are defined by their coordinates in the P1, P2, P3 and P4 entry fields. The *Divide1* and *Divide2* entry fields are used to specify equally−spaced divisions in the directions from P1 to P2 and from P1 to P3 respectively. If n and m are the numbers of divisions, $n.m$ QUAD4 field elements and $(n+1).(m+1)$ field points are created. Generally, the four corner points should be co−planar: if not, the field−point mesh surface becomes a ‘ruled surface’.

6.3.6 Creating a spherical field point mesh

A spherical field point mesh can be defined in the **Geometry, Field Points, Sphere** dialog:



Field Point Sphere Definition

Define the Spherical Field Point Mesh to Use :

Sphere Radius:

Division Factor : 4

□ No Merge

○ Position at Model Center

○ Define Position : 0 0 0

Figure 6.3–6 Geometry, Field Points, Sphere dialog

The sphere is either centered on the model (by using the *Position at Model Center* toggle button) or has its center at a specified location. The sphere is created by deforming a square box: each face of the box is divided into *n* elements. There are therefore $6.n^2$ QUAD4 field elements and $6.n^2+2$ field points created.

Hint: It is sometimes convenient to create a half–sphere or another sub–set of a sphere. This can be done as follows:

1 Create a spherical field point mesh using **Geometry, Field Points, Sphere**

2 Export the field point mesh using any format (*eg* Free) with the **File, Export** dialog, choosing the **Field Points** data type

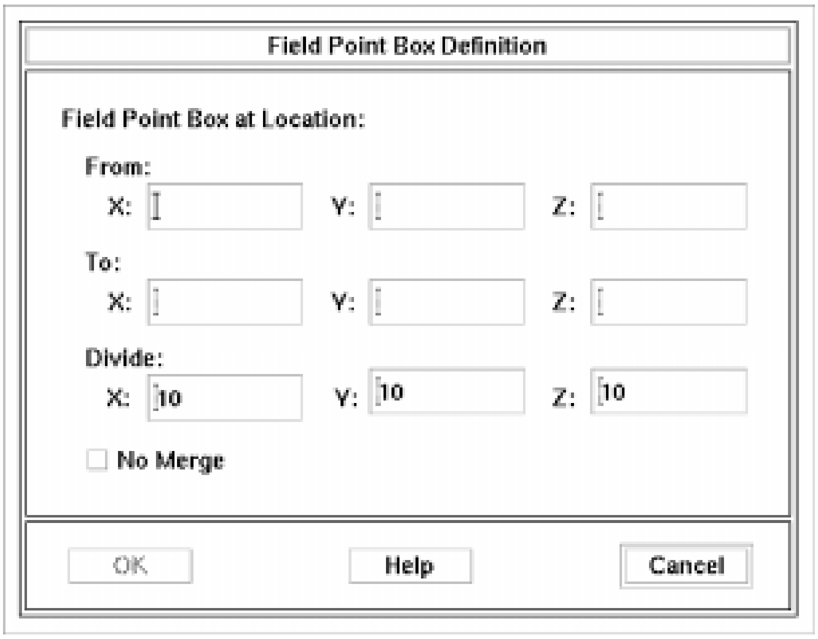
3 Import this file *as a mesh* using the **File, Import** dialog, choosing the **Mesh** data type

4 Create a Set containing the desired sub−set of elements (*eg* hemisphere) using the tools of the **Geometry, Sets** menu or graphically

5 Export the Set using any format with the **File, Export** dialog, choosing the **Set** data type. This file contains the required field point mesh which can be used whenever needed using the **File, Import** dialog and choosing the **Field Points** data type.

6.3.7 Creating a box–like field point mesh

A box−like field point mesh can be defined in the **Geometry, Field Points, Box** dialog:



Field Point Box Definition

Field Point Box at Location:

From:

X: Y: Z:

To:

X: Y: Z:

Divide:

X: 10 Y: 10 Z: 10

□ No Merge

Figure 6.3–7 Geometry, Field Points, Box dialog

The box has its edges parallel to the global model XYZ coordinate axes. The size of the box is defined by two opposite points as shown below, defined in the *From* and *To* entry fields:

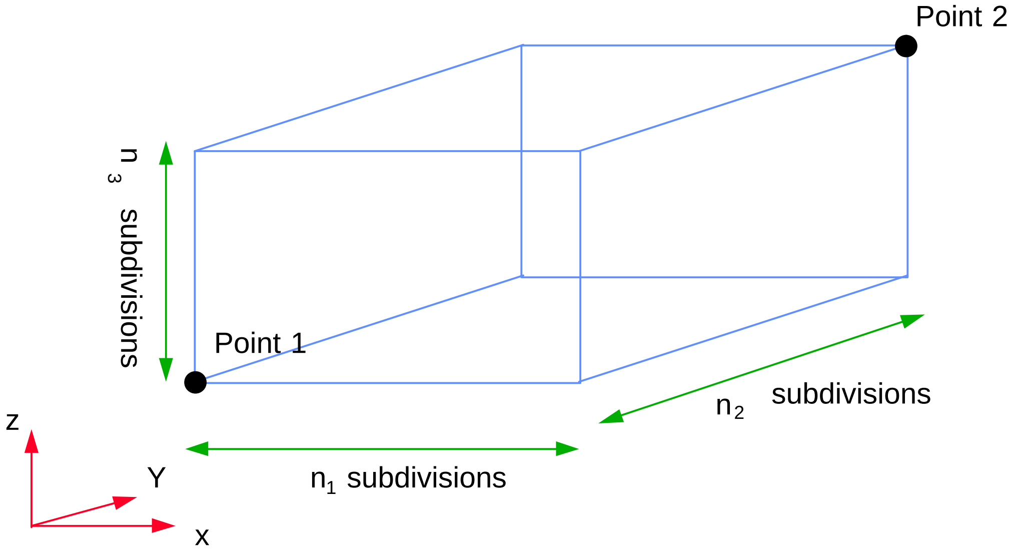
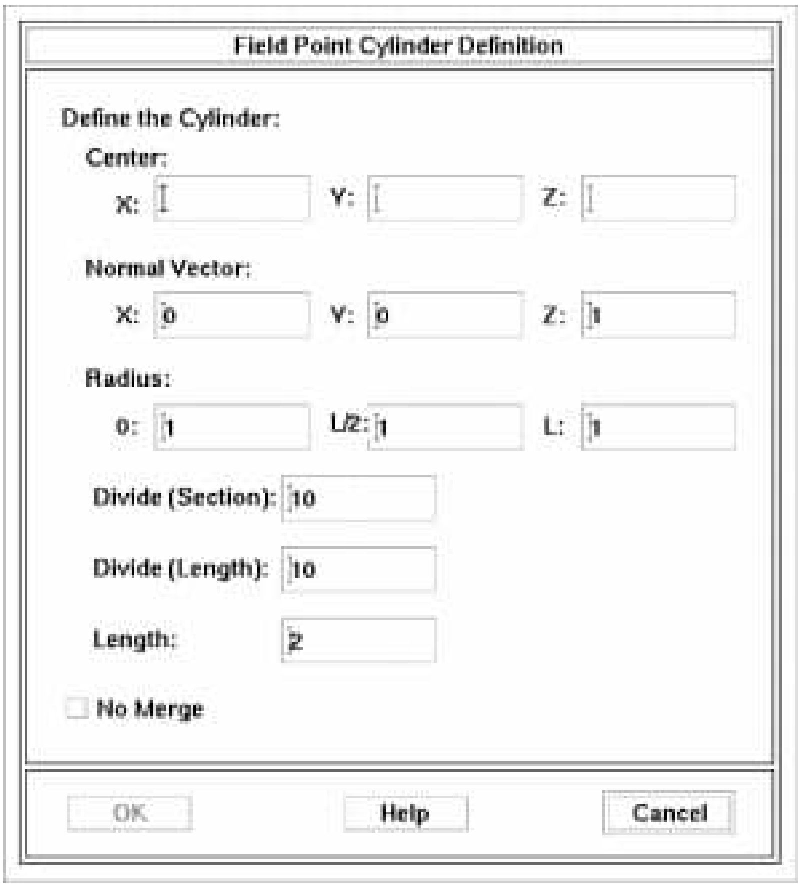


Figure 6.3–8 Parameters for a field point box

The numbers of equally−spaced sub−divisions in the x−, y− and z−directions is determined by the respective values in the *Divide* entry fields. The dialog creates $2(n_1.n_2+n_2.n_3+n_1.n_3)$ QUAD4 field elements and $2(n_1.n_2+n_2.n_3+n_1.n_3+1)$ field points.

6.3.8 Creating a cylindrical field point mesh

A ‘cylindrical’ field point mesh (with varying radius) can be created in the **Geometry, Field Points, Cylinder** dialog:



Field Point Cylinder Definition

Define the Cylinder:

Center:

X: Y: Z:

Normal Vector:

X: 0 Y: 0 Z: 1

Radius:

0: 1 L/2: 1 L: 1

Divide (Section): 10

Divide (Length): 10

Length: 2

□ No Merge

Figure 6.3–9 Geometry, Field Points, Cylinder dialog

The different parameters defining the cylinder are shown in Figure 6.3–10:

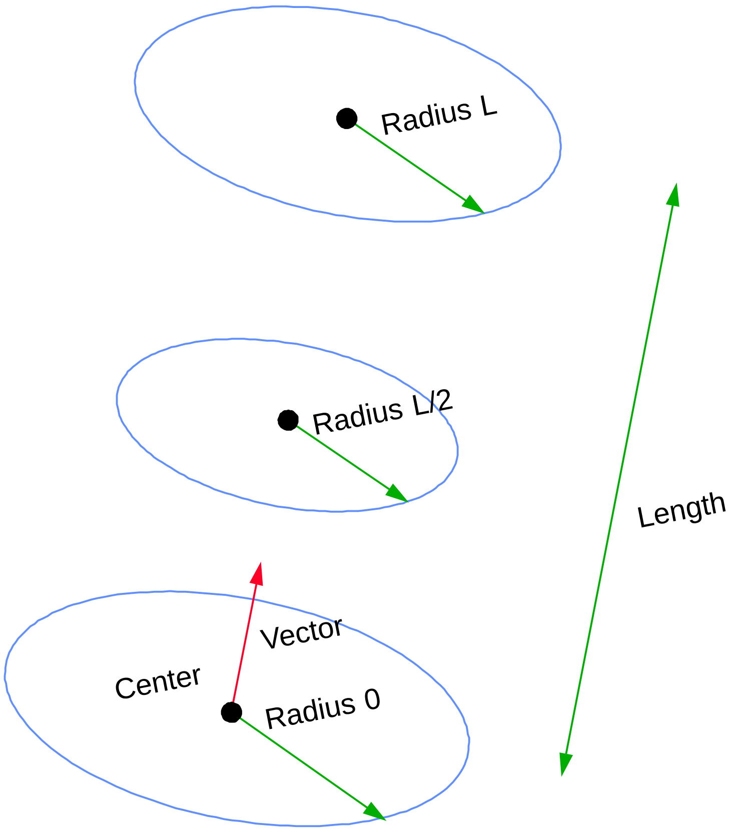


Figure 6.3–10 Parameters for a cylindrical field point mesh

Center defines the coordinates of the center of the first section. The cylinder axis is defined by its *Length* and the *Normal Vector* defining its orientation in space. The cylinder can have a varying radius defined by the value for the first (0), middle (L/2) and last (L) section. A parabola is defined to fit the radii at the first, middle and last sections, and is used as the generator of a surface of revolution.

There are no field points created radially across the sections. The *Divide (Section)* entry defines the number of divisions circumferentially and the *Divide (Length)* entry defines the number of divisions along the length. If n_L divisions are chosen along the length and n_S around the circumference, $(n_L+1).n_S$ field points and $n_L.n_S$ QUAD4 field elements are created.

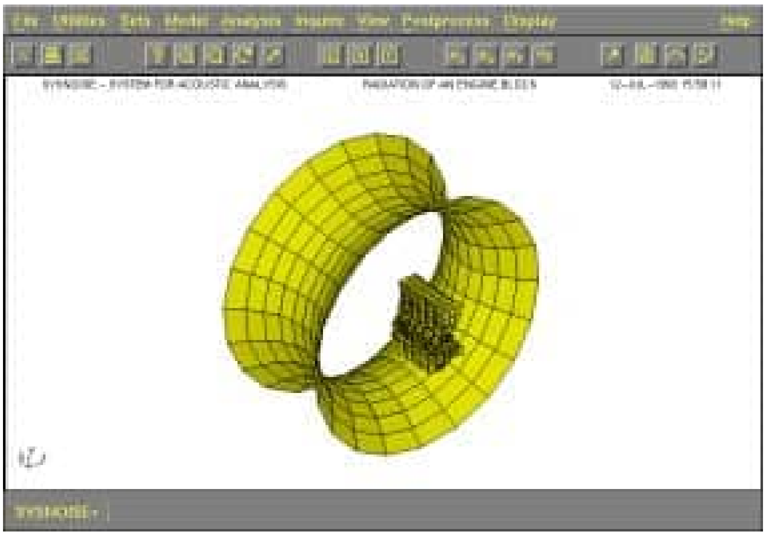
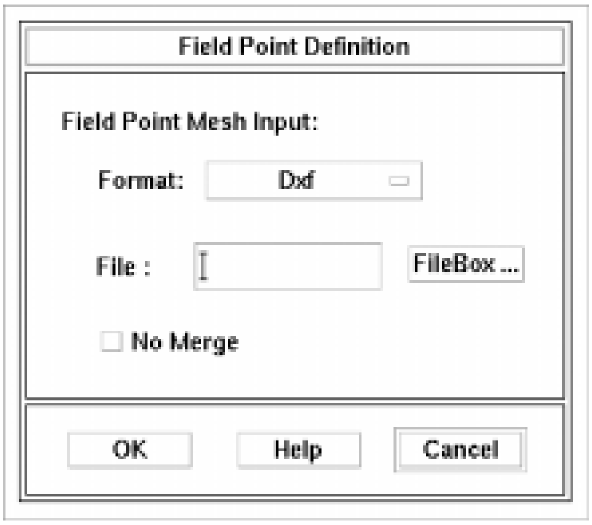

File Utilities Sets Model Analysis Inputs View Postprocess Display Help
SYSNOISE - SYSTEM FOR ACOUSTIC ANALYSIS
RADIATION OF AN ENGINE BLOCK
SYSNOISE>

Figure 6.3–11 Cylindrical field point mesh created with the values shown in the dialog depicted above.

6.3.9 Importing a field point mesh from another program

Any arbitrary mesh can be imported into RAYNOISE using the **Geometry, Field Points, Mesh** dialog. This may be used when a specific field point mesh is required which cannot be created by the tools provided by RAYNOISE.



Field Point Definition

Field Point Mesh Input:

Format: Dxf

File :

□ No Merge

Figure 6.3–12 Geometry, Field Points, Mesh dialog

The **Format** pull–down menu allows the selection of the format for the external interface file and the name of the file is specified in the *File* entry field.

This has the same effect as using the **File, Import** menu and choosing the **Field Points** data type, except that the Field Point, Mesh dialog contains the **NoMerge** toggle button. This button can be used to disable the automatic merging of coincident field points, which is not possible using the File, Import dialog.

The field point mesh can also be exported to another program using the **File, Export** menu and choosing the **Field Points** data type.

6.3.10 Reading individual field points from a file

A group of isolated field points, not connected by field elements, can be read from a free−formatted file. Each line of the file must contain the three coordinates of a point, with the values separated by at least one blank space.

The field points defined in this file are added to the current field point mesh using the **Geometry, Field Points, File** dialog, which also contains the **NoMerge** toggle button:

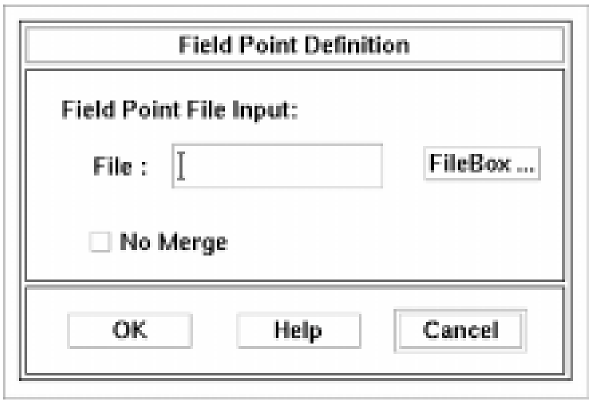


Figure 6.3–13 Geometry, Field Points, File dialog

Note that these points will be isolated (not connected by a field−point mesh) so the same remarks apply as in *Creating individual field points*, Section 6.3.2 on Page 6−12.

6.3.11 Syntax of the POINT command

The command **POINT** is used to define field points and field elements. The syntax corresponding to the different cases discussed above is given below.

- **Single Points**

Single points are defined by the command POINT followed by the 3 coordinates of the point. Several points can be defined within the same command but, to avoid ambiguities, it is recommended to define one point per line.

For example, the command:

```
RAYNOISE> POINT  0 0 0
POINT>            1 1 1
POINT>            2 2 2
POINT>            RETURN
```

creates three isolated points at coordinates (0., 0., 0.), (1., 1., 1.) and (2., 2., 2.).

- **STRAIGHT LINE**

Several points along a straight line can be defined by the complement **LINE**. For example, the command:

```
RAYNOISE> POINT  LINE 0 0 0 TO 1 1 1 DIVIDE 10  RETURN
```

creates eleven points on a straight line between points (0., 0., 0.) and (1., 1., 1.).

- **PLANE**

A regular rectangular distribution of field points in a plane can be defined by the complement **PLANE**. For example, the command:

```
RAYNOISE> POINT  PLANE 0 0 0 TO 1 0 0 DIVIDE 10 TO 0 1 0 DIVIDE 20
POINT>      RETURN
```

creates 231 (21 * 11) points in the plane defined by the three points (0., 0., 0.), (1., 0., 0.) and (0., 1., 0.).

- **SPHERE**

A spherical field point mesh can be defined by the complement **SPHERE**. For example, the command:

```
RAYNOISE> POINT  SPHERE 0 0 0  RADIUS 1  DIVIDE 8  RETURN
```

creates 386 field points and 384 field elements on a sphere centered on point (0., 0., 0.).

- **CIRCLE**

A circular field point mesh can be defined by the complement **CIRCLE**. For example, the command:

```
RAYNOISE> POINT CIRCLE 0 0 0 RADIUS 1 DIVIDE 10 VECTOR 0 0 1
POINT>       RETURN
```

creates 10 points in the XY plane (perpendicular to vector 0 0 1) on a circle of radius 1, centered on (0.,0.,0.).

- **CYLINDER**

A cylindrical field point mesh (with varying radius) can be defined by the complement **CYLINDER**. For example, the command:

```
RAYNOISE> POINT CYLINDER 0 0 0 RADIUS 1 2 3  LENGTH 4  DIVIDE 10 20
POINT>       VECTOR 0 0 1  RETURN
```

creates 210 (21 * 10) points on a cylinder whose axis is along the (0., 0., 1.) vector, whose radius varies continuously from 1 in the first section, to 2 in the middle section and 3 in the last section, and whose length is 4. There are 10 subdivisions around the circle and 20 along the length of the cylinder.

- **BOX**

A box–like field point mesh can be defined by the complement **BOX**. For example, the command:

```
RAYNOISE> POINT  BOX  0 0 0  TO  1 1 1  DIVIDE  3 4 5  RETURN
```

creates 224 field points and 294 field elements on a box whose extreme points are (0., 0., 0.) and (1., 1., 1.). There are 3, 4 and 5 sub–divisions in the X, Y and Z directions, respectively.

- **MESH**

A field point mesh defined in an external pre–processor can be defined by either of the two commands:

```
RAYNOISE> INPUT  POINT  FILE filename  FORMAT format_label  RETURN
```

or

```
RAYNOISE> POINT  MESH  FILE filename  FORMAT format_label  RETURN
```

- **Reading points from a file**

Points can be read from a file containing their coordinates using the **POINT** command followed by the complement **FILE**. For example:

```
RAYNOISE> POINT FILE POINT.DAT RETURN
```

reads point coordinates in file POINT.DAT which is an ASCII unformatted file containing the point coordinates (3 coordinates per line). Note that these points will be isolated (not connected by a field–point mesh) so the same remarks apply as in *Creating individual field points*, Section 6.3.2 on Page 6–12.

6.3.12 Remarks about field points

6.3.12.1 Cumulative definition of field points

Successive POINT commands are cumulative and keep on adding field points and field elements to existing elements and points. Unless the option **NoMerge** is applied (*Off* by default) any field points created coincident with existing field points will be merged into them and any corresponding field element topologies will be updated.

6.3.12.2 Deleting field points

All existing points can be cleared by the command:

```
RAYNOISE> POINT  RESET  RETURN
```

See the **Note!** and related **Hint** in *Field point mesh, Definition*, Section 6.3.1 on Page 6–11.

6.3.12.3 Displaying field points

Field points and corresponding field–point mesh elements will be displayed as soon as they are created, unless automatic updating of the display is turned off. However, isolated field points will not be apparent unless they are marked ('highlighted') as mentioned in *Creating individual field points*, Section 6.3.2 on Page 6–12. See also *Highlight Node*, section 12.1.4.8 Page 12–8.

To view the field point mesh inside an acoustic mesh, select the wireframe mode for the acoustic mesh and the face–painted mode (or wireframe mode) for the field point mesh. See *Corresponding Rendering Method*, Section 12.1.4.2 on Page 12–7.

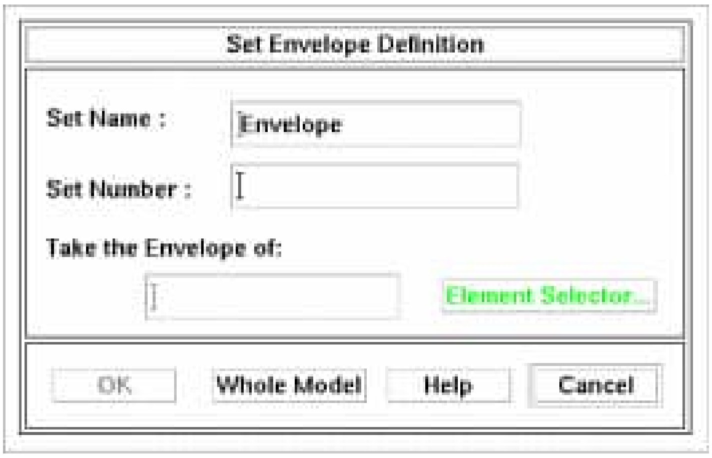
6.3.12.4 If the field point mesh passes through acoustic surfaces

It is recommended to avoid the situation in which elements of the field point mesh pass through acoustic surfaces, since there will be a 'jump' in the results between the two sides. This can produce odd effects when making contour plots, for instance.

6.4 Free edges

Free edges may be due to a deliberate choice of the user, to have an open or partly–open geometry, or may be due to errors in the mesh geometry (such as lack of connectivity or even a missing element). It is therefore of some interest to detect and display such free edges. It may also be useful to use free edges to define diffraction behavior, using selected free edges as diffraction edges.

Free edges are grouped in a set of faces using the **Geometry, Sets, Envelope Generation** dialog:



Set Envelope Definition

Set Name : Envelope

Set Number :

Take the Envelope of:

Figure 6.4–1 Geometry, Sets, Envelope Generation dialog

Figure 6.4–2 shows the free edges of a rectangular plate standing above a base plane. The top edge may then be identified as a diffraction edge.

The free edges are identified by a heavy line, colored according to the group color for the set.

Note! In most situations, the edges to be picked as diffraction edges will be free edges and therefore part of the Envelope set. If other edges are to be used (see *Equivalent screen approach*, Section on Page 3–21) the edges to be picked are not necessarily free edges: in this case, the Environment Variable INCLUDEALLFACES must be made TRUE. (See Section 11.8 on Page 11–19).

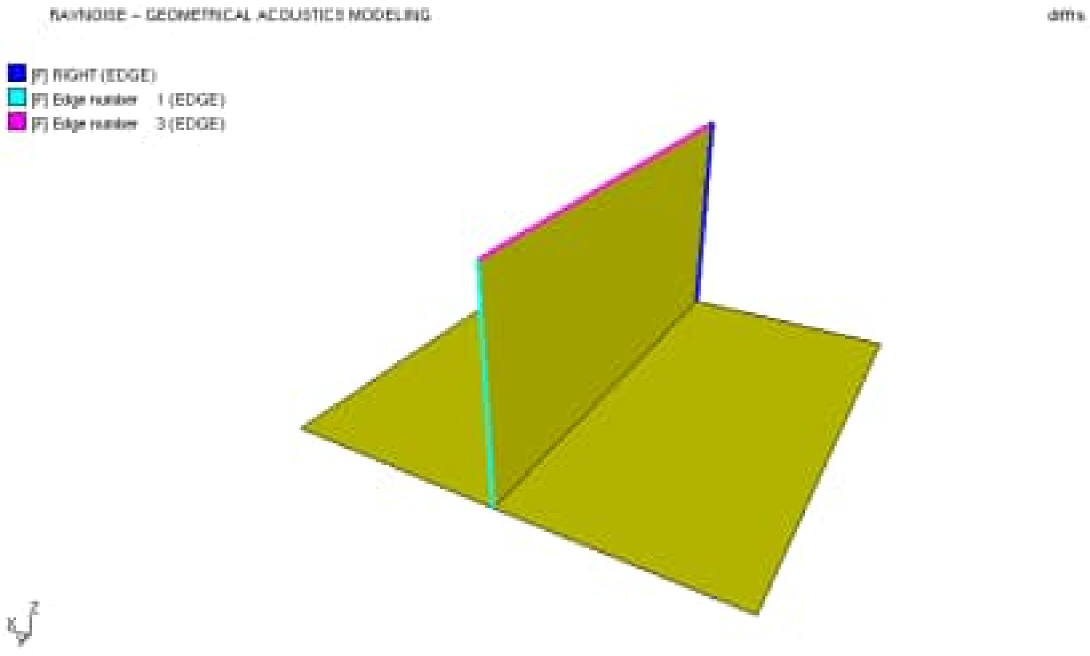


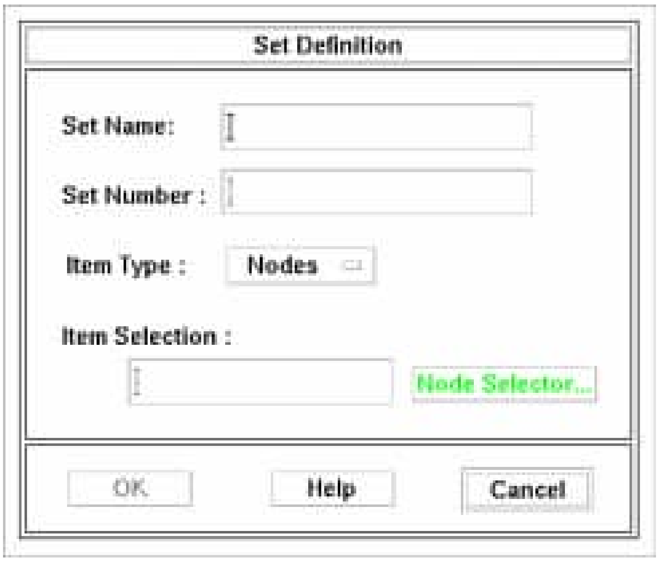
Figure 6.4–2 Free edges which may be picked as diffraction edges

6.5 Sets

6.5.1 Function of sets

The concept of Sets allows the user to manipulate parts of the model with only a short reference label, rather than having to repeat long item–picking sequences. The sets can be used for modifying geometry, adding materials and other characteristics, and for display purposes. Logical operations between sets allow new sets to be created rapidly.

Sets are created using the **Sets** command in the **Geometry** menu, followed by a selection of the type of entity to be put into the set, which opens the relevant dialog. This is also opened automatically, for the purpose of entering the set number and name, when launching a graphical–picking tool directly.



Item Type : Nodes

Figure 6.5–1 Example of Set–creation dialog: Geometry, Sets, Nodes

For more information, see:

- *Sets of nodes* (Section 6.5.2 on Page 6–28)
- *Sets of elements* (Section 6.5.3 on Page 6–29)
- *Layers* (Section 6.5.4 on Page 6–29)
- *Sets of faces* (Section 6.5.5 on Page 6–30)
- *Sets of field points* (Section 6.5.6 on Page 6–31)
- *Operations between sets* (Section 6.5.7 on Page 6–31)

6.5.2 Sets of nodes

6.5.2.1 Node selection

Nodes can be selected by using the graphical picking tools (see *Graphic picking tools*, Section 4.2.3.4 on Page 4–7) or menu– or command–based selection, using:

- node numbers (the label Internal means that internal rather than external, user–defined, numbers will be used; default is external numbers)
- node nearest to a specific coordinate (by a spherical searching process)
- all nodes at a specific coordinate (*ie* on a plane normal to x– y– or z–axis)
- nodes between parallel planes at specified x/y/z coordinates (see also *Geometrical tolerances*, Section 6.2.8 on Page 6–9)
- nodes attached to specified elements or faces, which form another set.

Node selector... opens the node selection sub–dialog.

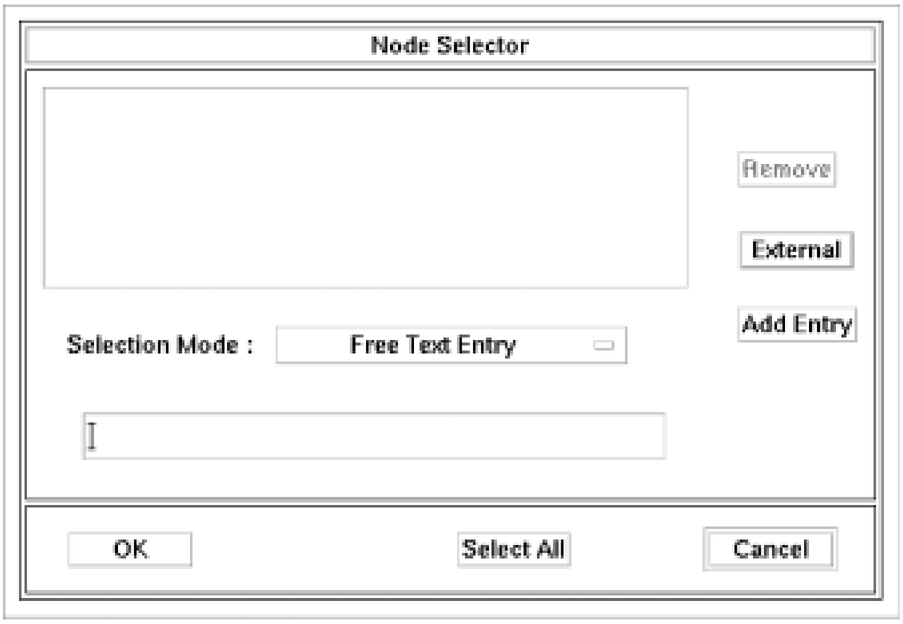


Figure 6.5–2 Node selector sub–dialog

6.5.2.2 Nodes within a set of another type

Nodes can be selected by reference to the nodes associated within another set which contains another type of entity, in particular elements. For example: *Nodes Set 4*, where *Set 4* is a set of elements. This can be done within the Sets creation, in order to create a set of nodes, or the reference can be used directly in another action.

6.5.3 Sets of elements

6.5.3.1 Element selection

Elements can be selected by using the graphical picking tools (see *Graphic picking tools*, Section 4.2.3.4 on Page 4–7) or menu– or command–based selection, using:

- element numbers (the label *Internal* means that internal rather than external, user–defined, numbers will be used; default is external numbers)
- element whose centroid is nearest to a specific coordinate (by a spherical searching process)
- elements between parallel planes at specified x/y/z coordinates (see also *Geometrical tolerances*, Section 6.2.8 on Page 6–9)
- elements with a specified material property
- elements of a particular type (only with a command, not via the GUI). Possible types are PLG*n*, where *n* is the number of vertices.

Note! The whole of an element (*ie* all its nodal points) must be within the specified volume (whether defined numerically or graphically) for that element to be selected.

Element selector... opens the element selection sub–dialog (similar to the node selector).

6.5.3.2 Importing sets of elements from an external file

Sets of elements can be imported from an external DXF–format file, if layers (AutoCAD term) are listed in the file.

See *Importing sets of elements from AutoCAD layers*, Section 5.3.7.2 on Page 5–14, for more information.

6.5.4 Layers

6.5.4.1 Definition of a layer

A layer is a special form of a set of elements. Layers have the property that an element can be a member of one and only one layer. If an element is later defined to be part of another layer, it loses its membership of the first layer (*ie* definitions are substitutive, whereas definitions of sets are cumulative, as regards the status of a specific element).

6.5.4.2 Selection of elements for a layer

Elements can be selected and put into layers using exactly the same techniques as for sets of elements: see *Sets of elements*, Section 6.5.3 on Page 6–29, for details.

6.5.5 Sets of faces

6.5.5.1 Face selection

Faces can be selected by using the graphical picking tools (see *Graphic picking tools*, Section 4.2.3.4 on Page 4–7) or by menu or command–based selection, using:

- face numbers
- the face with its centroid nearest to some specified coordinates
- all faces at a specific coordinate (*ie* on a plane normal to x– y– or z–axis)
- faces between parallel planes at specified x/y/z coordinates (see also *Geometrical tolerances*, Section 6.2.8 on Page 6–9)
- faces of a selection of elements: only the free edges of the elements are selected, *ie* those not shared with another element (even if that element is not in the selection for which faces have been requested)
- all faces of a particular type (a list of topological types is offered)
- all faces of a given dimension (point, line, surface...)

Face selector... opens the face selection sub–dialog.

6.5.5.2 Envelope faces

The envelope is a special set of faces, consisting of the free edges of the selected elements. The envelope set is created in the same way as other sets of faces, but using the separate **Envelope** command on the **Geometry** menu.

Hint: If all elements of the mesh are selected, this can be of particular use as a check that the mesh is closed, without gaps or disconnected elements (which may allow rays to escape from the interior during the ray–tracing).
The Envelope of the whole model is also used to provide the set of free edges from which diffraction edges can be selected. See Diffraction edges, Section 7.6 on Page 7–25.

6.5.6 Sets of field points

6.5.6.1 Field point selection

Field points can be selected by using the graphical picking tools see *Graphic picking tools*, Section 4.2.3.4 on Page 4–7) or by menu or command–based selection, using:

- point numbers (the label Internal means that internal rather than external, user–defined, numbers will be used; default is external numbers; in practice, internal and external numbers are often the same)
- the point nearest to a specified coordinate
- all points at a specific coordinate (*ie* on a plane normal to x– y– or z–axis)
- points between parallel planes at specified x/y/z coordinates (see also *Geometrical tolerances*, Section 6.2.8 on Page 6–9)

6.5.7 Operations between sets

6.5.7.1 Operations between named sets

Operations between sets can be used to create new sets. The operations are performed using Boolean logic. The possible operators are:

- union
- intersection
- difference

The commands are found in **Sets, Operations between sets** in the Geometry menu.

6.5.7.2 Graphical operations between sets

It is also possible to make immediate operations between sets, when these are displayed (colored) and the new set is being created by graphic picking. The button **Intersection with currently displayed sets** in the graphic picking dialog is used. The newly–created set becomes the resulting sub–set of the picked entities. This may be useful, for example, to pick a set of elements on one side of structure, using a graphic selection, without picking the elements within the selection region which are hidden behind: first make a set of elements for the visible surface, rotating the model as appropriate if using graphic picking, then pick the required elements on the visible side of the model.

It is important to note that several of the set–creation commands, in particular the graphical picking operations, select all the entities within a defined region which is infinite in one direction, for instance along the eye–line from the viewpoint.

7

Model data

This chapter describes the various data used to define a model, including:

○ *Name*

○ *Medium*

○ *Materials*

○ *Sources and background noise*

○ *Diffraction*

○ *Using tables of data*

○ *Units and Reference values*

7.1 Model name

Each model can be given a specific name using the dialog **Model, Model Name**. This dialog modifies the title of the currently–active model. The model name is stored with the model data in the database file, but does not influence the database file name.

The name of the currently–active model is used as a header in all graphic windows and in some output files.

Note! Several models can have the same name. The user should be careful of the possible confusion which may result from this: in general, it is recommended to give each model a unique name.

The RAYNOISE Command Language syntax of the corresponding command is:

```
RAYNOISE> TITLE "model name"  RETURN
```

Note that the *model name* must be enclosed in quotes if it contains blanks. This is done automatically if the dialog is used.

7.2 Properties of the medium

7.2.1 Definition

Medium properties refer to the properties of the medium through which the acoustic wave fronts propagate, such as: sound velocity, mass density, temperature, relative humidity, air absorption coefficients.

The medium properties can be entered in two modes:

- if the medium is air, by determining the acoustic data based on the atmospheric conditions
- for any medium, by giving the acoustic data directly

7.2.2 Medium properties for air

Enter the temperature and the relative humidity: this will fix the sound velocity, the air absorption coefficients and the mass density, by derivation from standard data.

- See *Units in RAYNOISE* (Section 7.8 on Page 7–35) for information on the units of the values being entered.

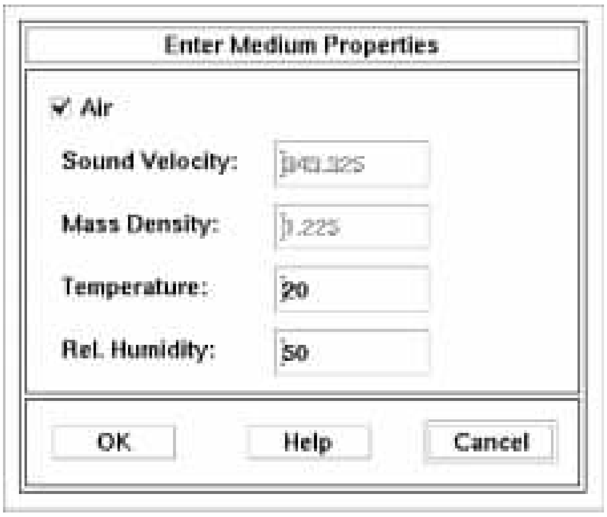
7.2.3 Medium properties for any medium

For any medium (typically, other than air – but including ’air’ if non–standard properties are required): enter the sound velocity and mass density directly.

- See *Units in RAYNOISE* (Section 7.8 on Page 7–35) for information on the units of the values being entered.

Note! The volumic absorption is *zero* for non–air cases.

7.2.4 Procedure



Enter Medium Properties

☑ Air

Sound Velocity: 343.325

Mass Density: 1.225

Temperature: 20

Rel. Humidity: 50

Figure 7.2–1 Medium properties dialog

Choose **Medium** from the **Model** menu. Set the Air/other toggle. Enter the values for the medium properties. Click **OK** to set the medium properties.

7.3 Material properties

7.3.1 Defining materials

7.3.1.1 Definition and types of property

A material can be defined with certain acoustical properties for absorption, diffusion and transmission. These properties can be entered as octave band values or by means of a table defining frequency−dependence. (See *Tables*, Section 7.7 on Page 7−28)

- See *Units in RAYNOISE* (Section 7.8 on Page 7−35) for information on the units of the values being entered.

7.3.1.2 Material property databases

Material properties can also be selected from a list of predefined properties stored in the files RAYNOISE.SAB, RAYNOISE.DIF, RAYNOISE.TL (for absorption, diffusion and transmission, respectively). The location of these files is defined by the environment variables DATASABINEFILE, DATADIFFUSIONFILE and DATATRANSMISSIONFILE (See *Environment variables*, Section 11.8 on Page 11−19)

Note! Materials have a unique number and name in the particular model in which they are defined and used. These are not the same as the labels in the databases: in particular, the material number in the model is not the same as its sequence number in the database.

7.3.1.3 To create a material

Choose **Materials...** from the **Model** menu. Click on **New** to create a new material. Click the property types to be active. Choose between octave−wise or table−wise mode. If octave−wise, enter values manually or pick values from the pre−defined list by clicking on **Database**. This brings up the scrollable database dialog (second figure below). Highlight a material from the database and click on **Select** to return to the first dialog: the properties, name *etc* are inserted into the entry fields. Then, click on **Add** to save the material in the currently−open model database.

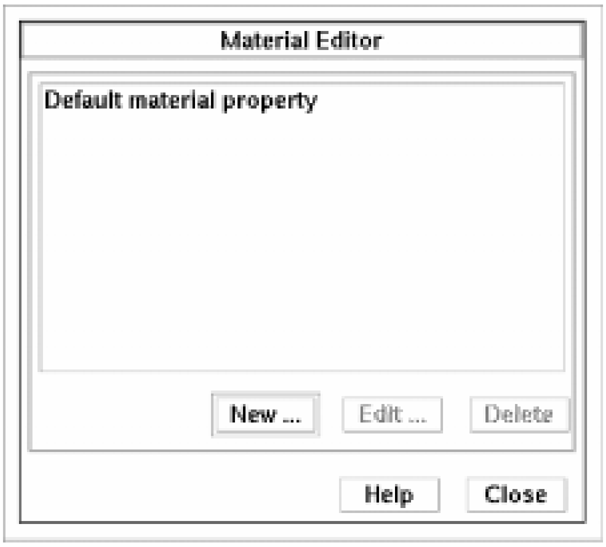


Figure 7.3–1 Model, Materials dialog

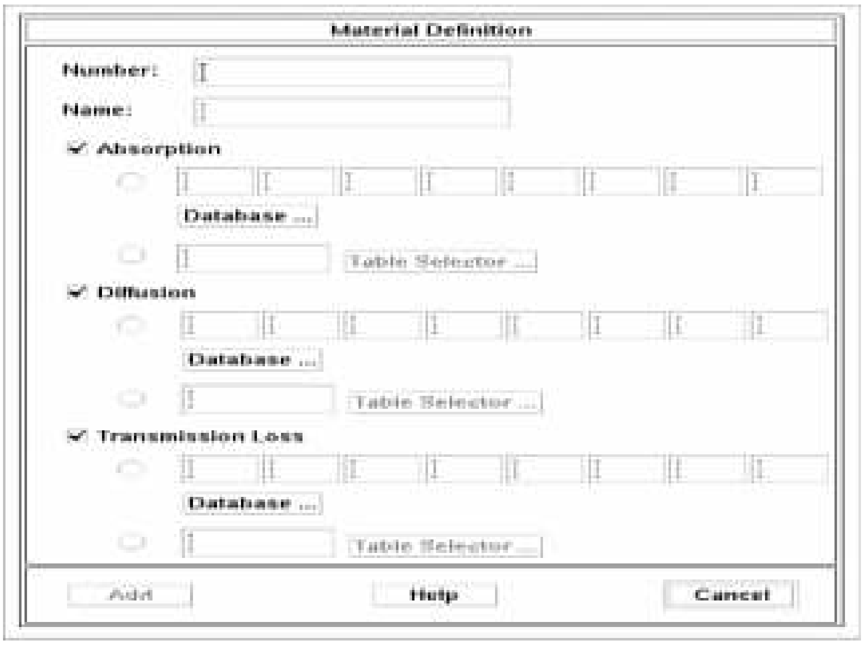
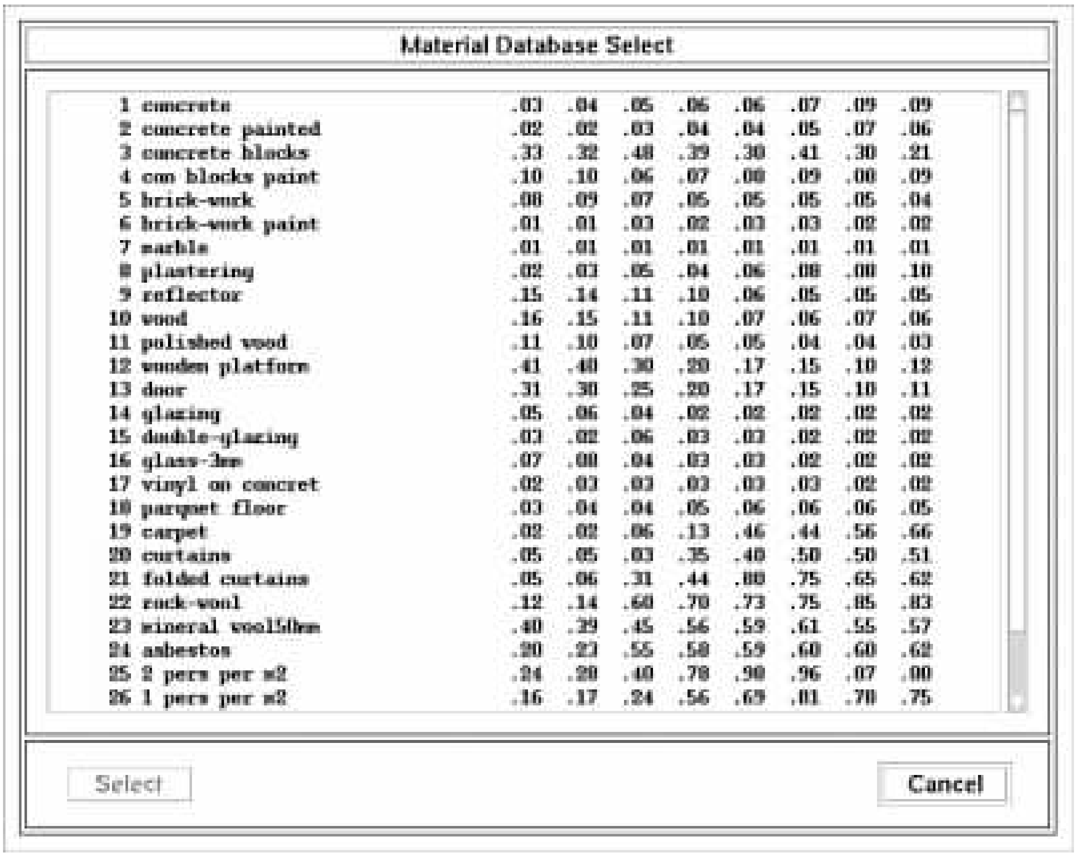


Figure 7.3–2 Model, Materials Definition dialog



```
 1 concrete                  .03  .04  .05  .06  .06  .07  .09  .09
 2 concrete painted          .02  .02  .03  .04  .04  .05  .07  .06
 3 concrete blocks           .33  .32  .48  .39  .30  .41  .30  .21
 4 con blocks paint          .10  .10  .06  .07  .08  .09  .08  .09
 5 brick-work                .08  .09  .07  .05  .05  .05  .05  .04
 6 brick-work paint          .01  .01  .03  .02  .03  .03  .02  .02
 7 marble                    .01  .01  .01  .01  .01  .01  .01  .01
 8 plastering                .02  .03  .05  .04  .06  .08  .08  .10
 9 reflector                 .15  .14  .11  .10  .06  .05  .05  .05
10 wood                      .16  .15  .11  .10  .07  .06  .07  .06
11 polished wood             .11  .10  .07  .05  .05  .04  .04  .03
12 wooden platform           .41  .40  .30  .20  .17  .15  .10  .12
13 door                      .31  .30  .25  .20  .17  .15  .10  .11
14 glazing                   .05  .06  .04  .02  .02  .02  .02  .02
15 double-glazing            .03  .02  .06  .03  .03  .02  .02  .02
16 glass-3mm                 .07  .08  .04  .03  .03  .02  .02  .02
17 vinyl on concret          .02  .03  .03  .03  .03  .03  .02  .02
18 parquet floor             .03  .04  .04  .05  .06  .06  .06  .05
19 carpet                    .02  .02  .06  .13  .46  .44  .56  .66
20 curtains                  .05  .05  .03  .35  .40  .50  .50  .51
21 folded curtains           .05  .06  .31  .44  .80  .75  .65  .62
22 rock-wool                 .12  .14  .60  .70  .73  .75  .85  .83
23 mineral wool50mm          .40  .39  .45  .56  .59  .61  .55  .57
24 asbestos                  .20  .23  .55  .58  .59  .60  .60  .62
25 2 pers per m2             .24  .28  .40  .78  .90  .96  .87  .80
26 1 pers per m2             .16  .17  .24  .56  .69  .81  .78  .75
```

Figure 7.3–3   Model, Materials Definition, Database selector

7.3.1.4 To edit a material

Choose **Materials...** from the **Model** menu. First, select the material by highlighting it in the list of defined materials, then click on **Edit**. **Change** to apply the changes, then **Close** the dialog.

7.3.1.5 To delete a material

Choose **Materials...** from the **Model** menu. First, select the material by highlighting it in the list of defined materials, then click on **Delete**, then **Close** the dialog.

7.3.1.6 Default material

When reading in a new geometry file (*ie* a mesh), a default material is created having 0.0 absorption, no diffusion and no transmission properties. This default material, named ‘default_material’, is automatically assigned to all elements.

Hint: You can inquire about the presently–defined materials by using **Inquire, Materials**.

7.3.2 Assigning materials to elements

Before a calculation can be performed, materials must be assigned to all elements. Any element which does not have a material assigned to it explicitly will have the default material (perfect reflector) assigned to it.

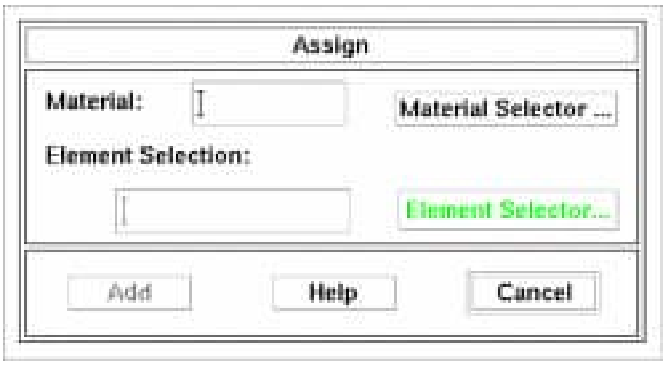


Figure 7.3–4 Model, Assign... dialog

To assign a material to a selection of elements: Select **Assign...** from the **Model** menu. Enter the material number, or click on **Material Selector...** to select a material by name, by highlighting it in the pop–up list. Enter element numbers, or click on **Element Selector...** or use the graphical picking buttons from the Toolbar, to select the elements to which this material will be assigned. Click **Add** to apply the assignment or **Cancel** to abort.

The assigned materials can be viewed on the mesh display, by choosing **Materials** from the **View** menu (see *Selection of materials*, Section 8.1.3 on Page 8–4)

7.4 Sound sources

7.4.1 Available source types

Before initiating a calculation, at least one sound source has to be defined.

There are 5 source types :

- point sources
- line sources
- area sources
- panel sources
- transmission sources

Every source is characterized by its position in the Cartesian coordinate system, its sound power level, its time delay, its orientation, its directivity pattern and its immission zone.

Sources are created and modified via the Source Editor dialog:

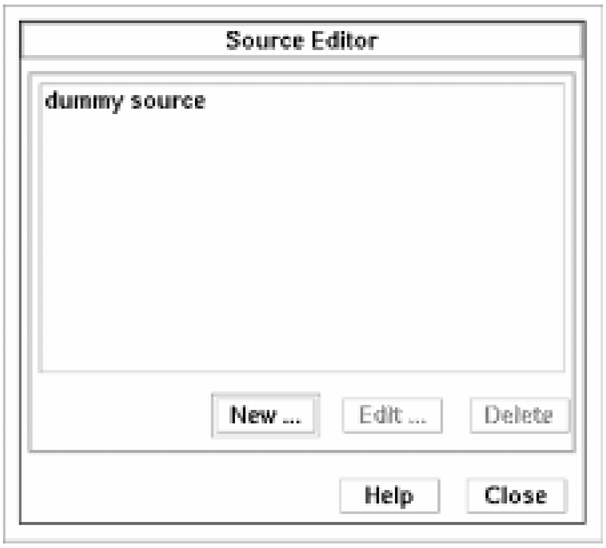
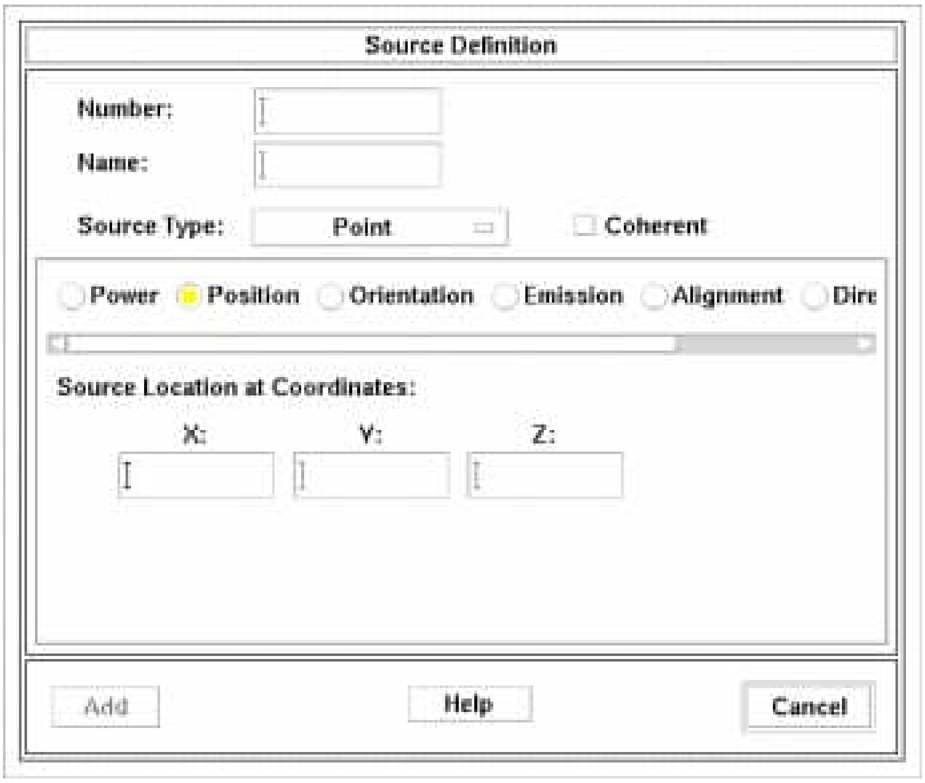


Figure 7.4–1 Model, Sources... dialog

See *To create a source*, Section 7.4.4 on Page 7—22, for a description of the procedure.

7.4.2 Source type and position

7.4.2.1 General dialog



Source Definition

Number:

Name:

Source Type: Point

□ Coherent

○ Power ● Position ○ Orientation ○ Emission ○ Alignment ○ Dire

Source Location at Coordinates:

X: Y: Z:

Figure 7.4–2 Model, Source Definition dialog (Position)

7.4.2.2 Point source

A point sources is defined as a single ’monopole’ at a given location in space. Its characteristics conform to a free–field spherical source modified by any directivity and/or emission angle limits which may be defined.

Menu option and required data: **Point** *x,y,z*

7.4.2.3 Line source

A line sources is defined as a group of point sources along a straight line from point a1 to b1, all with the same characteristics. The number of point sources in this group is user–defined, and equals $n+1$, where n is the number of divisions along the line.

Menu option and required data: **Line** *x1,y1,z1* TO *x2,y2,z2* DIVIDE *n*

7.4.2.4 Area source

An area source is defined as a group of point sources along a parallelogram, defined by three points a2, b2 and c2 (see Figure 7.4–3).

Menu option and required data:

Area *x1,y1,z1* TO *x2,y2,z2* DIVIDE *n1* TO *x3,y3,z3* DIVIDE *n2*

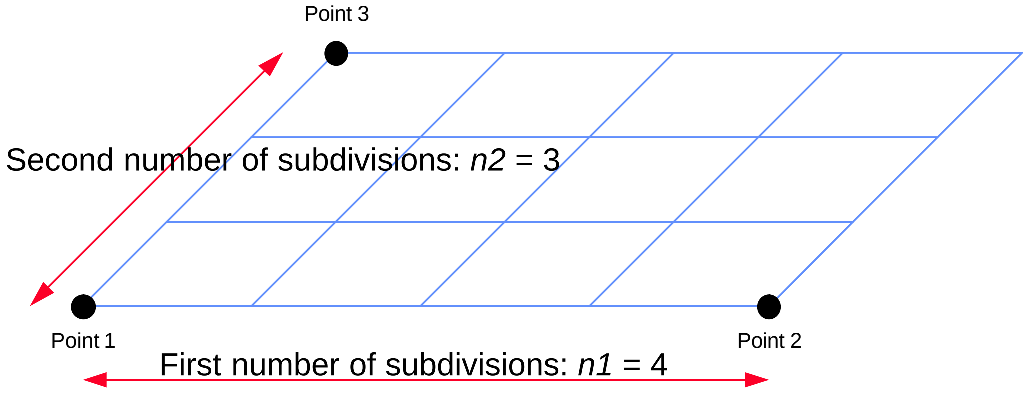
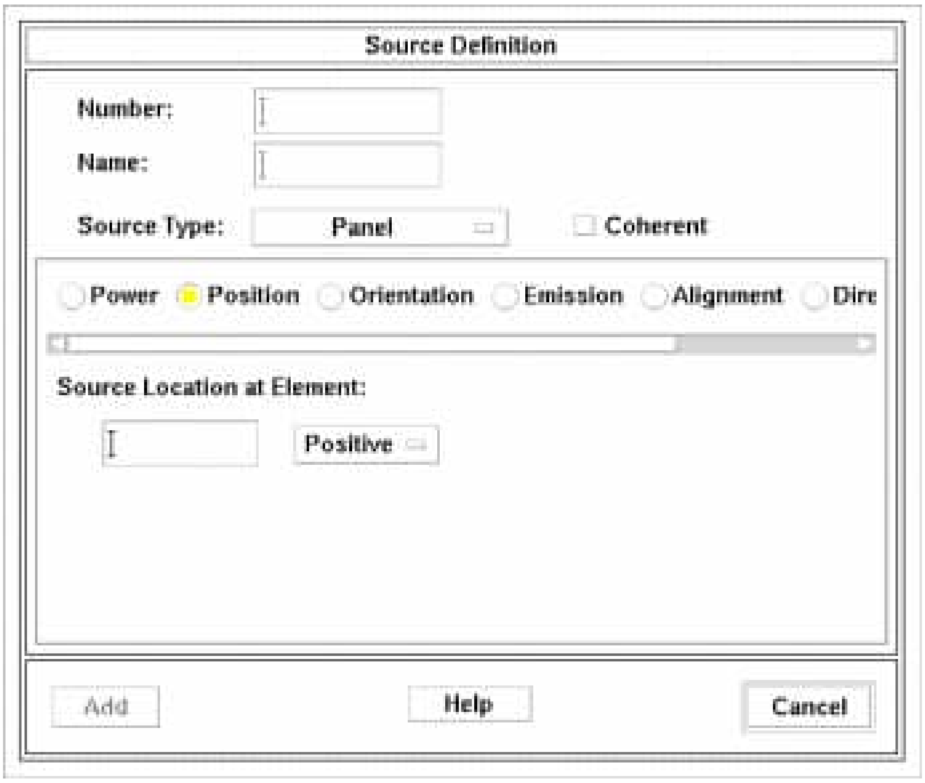

Point 3
Second number of subdivisions: *n2* = 3
Point 1
Point 2
First number of subdivisions: *n1* = 4

Figure 7.4–3 Geometry of an area source

7.4.2.5 Panel and transmission sources

Panel sources and transmission sources are defined as hemispherical monopoles located at the centroid of an element (surface) and oriented according to the element normal (positive or negative). The centroid location is found automatically by RAYNOISE. See *Reversing element orientations*, Section 6.2.6 on Page 6–7, for further information on how to control the orientation of elements (*eg*, to make them all point into a building, for convenience).



Source Definition

Number:

Name:

Source Type: Panel

□ Coherent

○ Power ● Position ○ Orientation ○ Emission ○ Alignment ○ Dire

Source Location at Element:

Positive

Figure 7.4–4 Model, Source Definition dialog (Panel source)

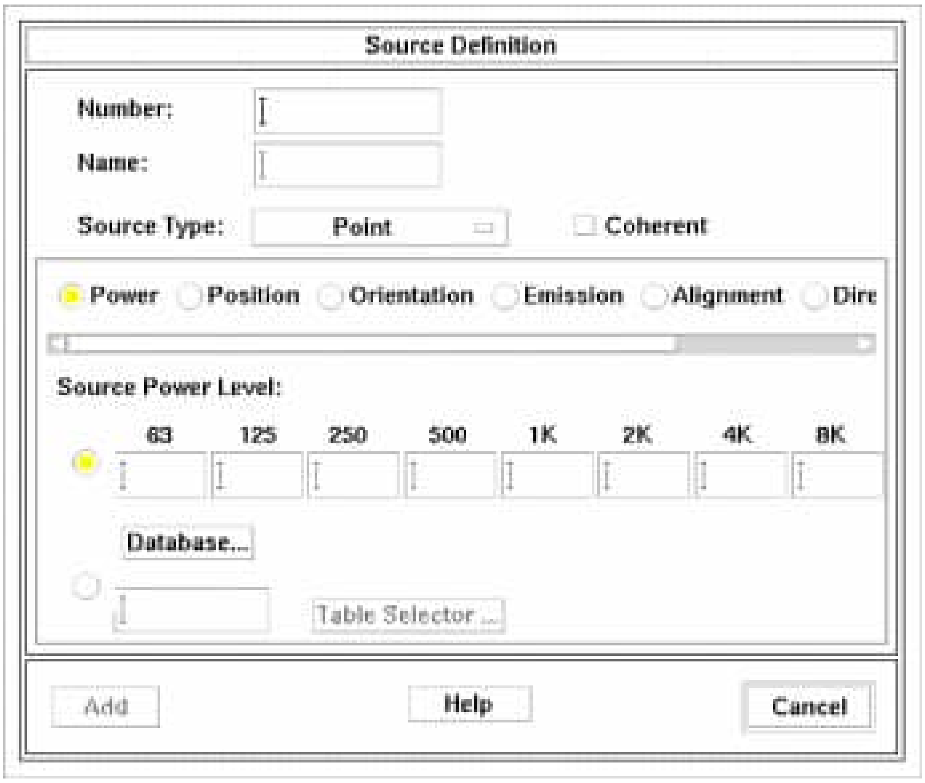
Menu option and required data:

(Transmitting) **Panel**, ELEMENT *e* , POSITIVE (NEGATIVE)

7.4.3 Source properties

7.4.3.1 Sound power level of sources

The sound power level of a source can be given in the 8 frequency octave bands 63Hz to 8kHz or in a table, containing any list of values at any given frequencies. (See *Tables*, Section 7.7 on Page 7–28, for more information on defining and using tables).



Source Type: Point

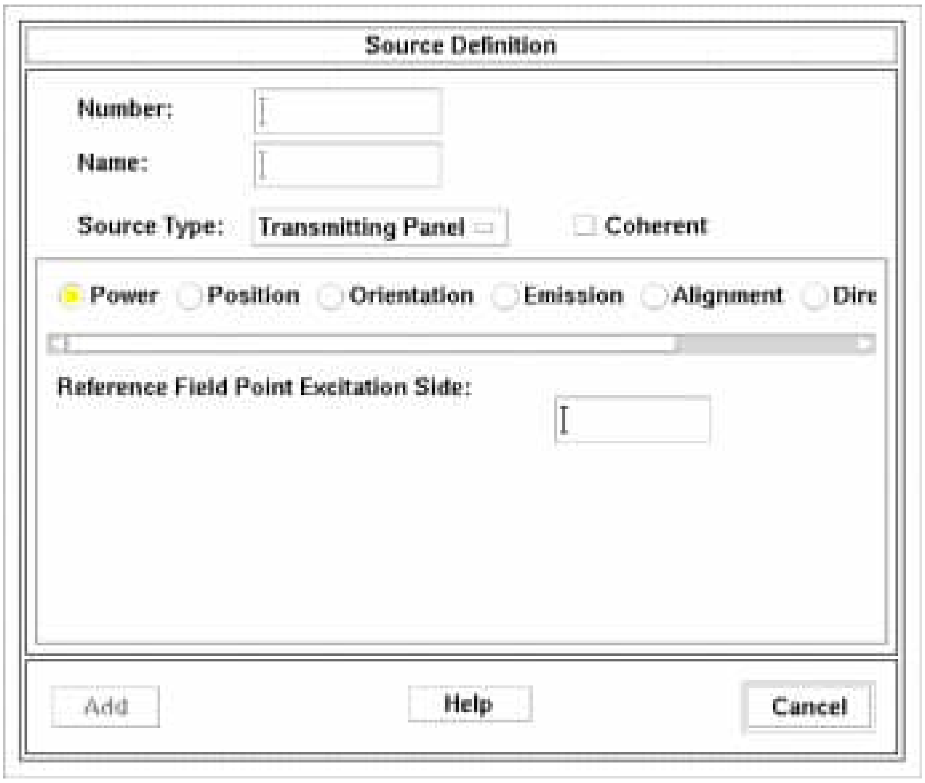
● Power

Figure 7.4–5 Model, Source Definition dialog (standard Power)

Note! The power of a line source or an area source is divided equally amongst its constituent point sources.

This procedure is opposite to the one in RAYNOISE Revision 2.1 in which the defined sound power level was applied at *each* of the point sources generated by the line or plane definition.

The sound power level of a transmission source depends on the excitation level and transmission level of the panel with which it is associated. A field point number needs to be entered, whose sound pressure level is taken to represent the incident field at the excitation side. See *Sound transmission modeling*, Section 3.6 on Page 3–29.



Source Definition

Number:

Name:

Source Type: Transmitting Panel

□ Coherent

● Power ○ Position ○ Orientation ○ Emission ○ Alignment ○ Dire

Reference Field Point Excitation Side:

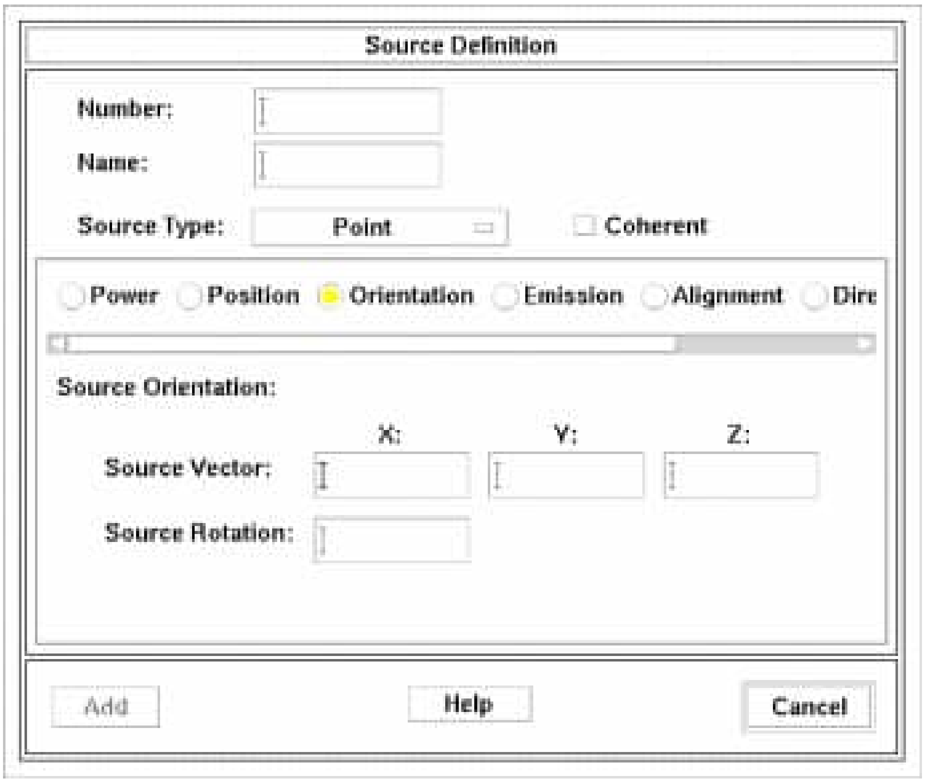
Figure 7.4–6 Model, Source Definition dialog (Transmission Power)

Note! For an omnidirectional source, the power relates to a full sphere at each constituent point, or a hemisphere in the case of panel or transmission sources. If emission angles are used to limit the solid angle into which the source radiates, the total sound power which the source radiates into the model will be reduced in proportion to the fraction of a sphere (or hemisphere) so defined. The intensity of each ray within the actual radiating region remains the same as it would have been had it been part of the corresponding omnidirectional (or hemispherical) source. Thus, the total power radiated into the model will be affected by any subsequent changes to emission angles.

For a source where directivity is specified, *ie* reduction in the sound intensity in rays off the central axis of the source, a similar effect occurs: the intensity on the central axis relates to the specified sound power radiating omnidirectionally over a full sphere, but the intensities in all other directions are reduced in proportion to the attenuations interpolated from the directivity diagram. The total sound power radiated into the model will thus be reduced compared to an omnidirectional source, and will also be affected by subsequent changes to the directivity data.

7.4.3.2 Source orientation

A directional source can be aimed in a specified direction. This orientation is defined by a vector (direction cosines) defining the principal axis (local z–axis) and the rotation angle around it. The local x–axis is set to be parallel to the global XY plane (and in the indeterminate case, which is also the default, parallel to the global X–axis) before applying any such rotation. A positive rotation is applied in a right–handed sense about the principal axis. The default orientation has direction cosines 0,0,1 (principal axis parallel to global Z–axis) and no rotation. Self–evidently, orientation is irrelevant to an omnidirectional source.



Source Definition

Number:

Name:

Source Type: Point

□ Coherent

○ Power ○ Position ● Orientation ○ Emission ○ Alignment ○ Dire

Source Orientation:

Source Vector: X: Y: Z:

Source Rotation:

Figure 7.4–7 Model, Source Definition dialog (Position)

7.4.3.3 Emission angles

The emission angles define the boundaries of an area on the the unit sphere, centered at the origin of the source, through which the rays are radiated. These angles are defined in a local xyz–system which has its center at the sound source, its z–axis according to the principal axis orientation, and the other two axes likewise (see Source orientation, above, and Figure 7.4–8 below). The 'horizontal' emission angles (*ie* in the local xy–plane) define the emission region around the principal axis, and must lie between –180 and +180 degrees. The vertical emission angles (*ie* in the local 'xz–plane') define how far off the principal axis (at +90 degrees) the radiation occurs: they must lie between −90 and +90. For example, horizontal emission angles of −180/+180 and vertical emission angles of +70/+90 degrees will define a radiation cone with a 20−degree half−angle, centered on the principal axis. If the vertical angles were the same and the horizontal emission angles were 0/+90 degrees, the result would be a quadrant of a cone, in the local +x+y−quadrant. Default emission angles are −180/+180, −90/+90 respectively, *ie* a complete sphere.

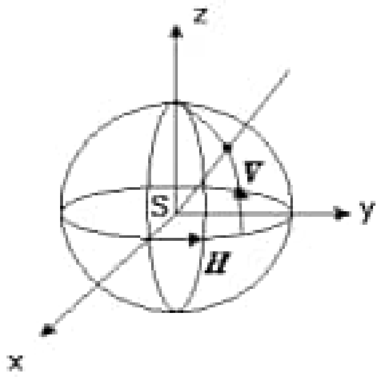


Figure 7.4–8 Definition of emission angles

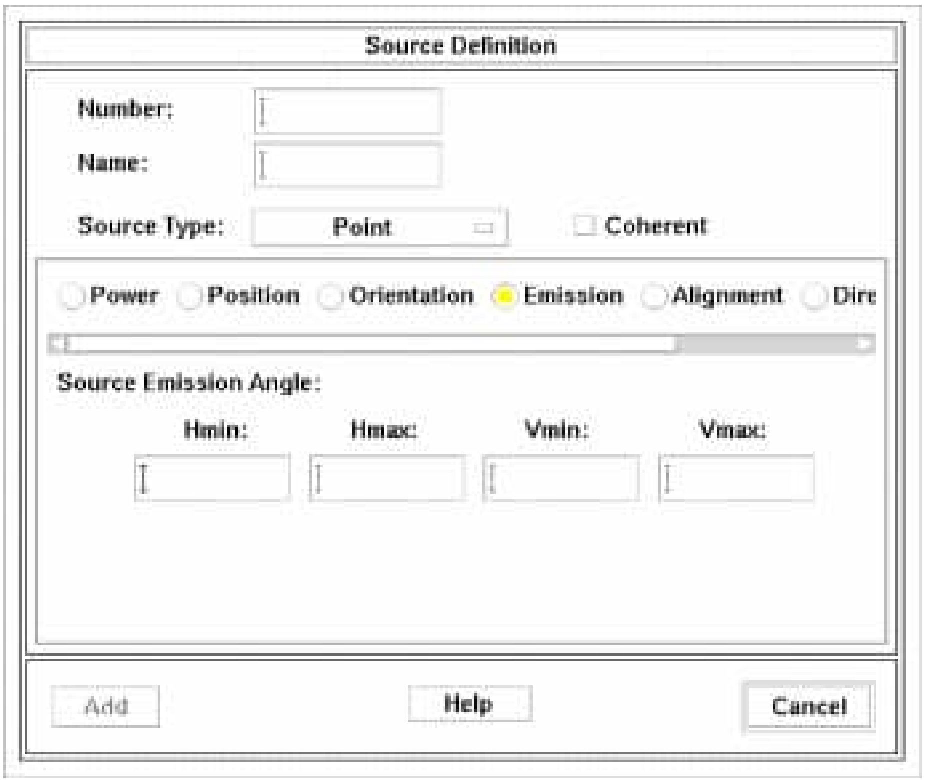


Figure 7.4–9 Model, Source Definition dialog (Emission angles)

Hint: Using emission angles reduces the number of rays to be calculated, since the solid angle of each ray is found by dividing a complete sphere by the calculation parameter 'number of rays per sphere'. Hence, it can be useful to limit the emission angles of sources near to surfaces, with hemispherical, quadrant or octant emission.

7.4.3.4 Directivity pattern

Another property of a sound source is its directionality, *ie* the fact that it does not emit sound with equal intensity in all directions. Usually, the directionality of a source is characterized by its directionality in a horizontal and a vertical plane (see Figures 7.4–10 and 7.4–11).

The horizontal and vertical planes are fixed in a similar manner to the local axes for Emission angles (Section 7.4.3.3 on 7.4.3.3), using direction cosines to fix the local principal axis of the source, plus a rotation about that axis if required.

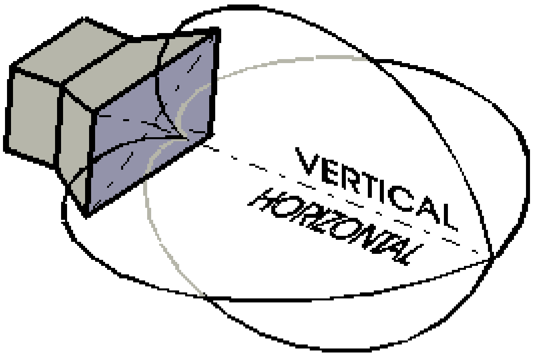


Figure 7.4–10   Directivity of a source in space

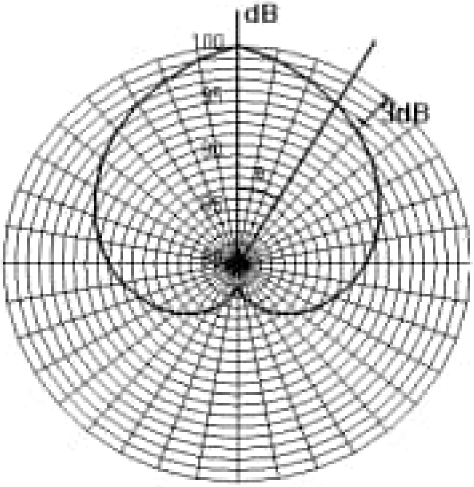
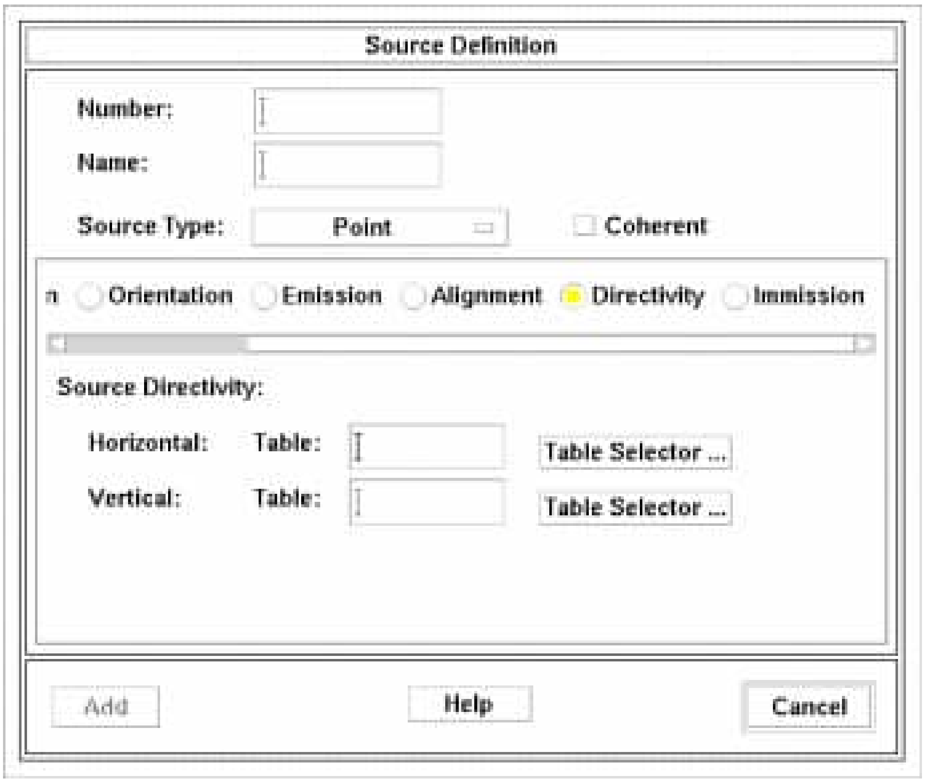


Figure 7.4–11   Directivity pattern of a source

The horizontal or vertical directivity pattern (Figure 7.4–11) is determined by a directivity table, which contains a list of attenuation and phase values as functions of frequency and angle. Positive attenuation values are the reduction in sound power, at the given angular direction with respect to the unattenuated sound power in the direction of the major axis (local z–axis) – for instance, values of 3 (dB) at 30 degrees, 6 at 60 and so on, for the directivity shown in Figure 7.4–11. (Note that it is nevertheless possible to have attenuation on the major axis itself, and that the attenuation can vary arbitrarily and is not limited to a cardioid). Positive phase angles imply a delay in the emission of an impulse, *ie* they are equivalent to a positive Alignment, with respect to the emission of an impulse in the direction of the major axis.

Hint: The directivity tables can be defined before defining the source, or the table definition can be launched as a sub–dialog from within the source definition sequence, using the **Table Selector...** button, followed by **New**. It is recommended usually to define the tables in advance, by using the **Model, Tables** menu entries (or reading command files with the corresponding commands) or by importing the tables from files. Retaining the directivity data in separate files is especially useful if it is at all complicated (complex angular– and frequency–dependencies).



Source Definition

Number:

Name:

Source Type: Point

□ Coherent

n ○ Orientation ○ Emission ○ Alignment ● Directivity ○ Immission

Source Directivity:

Horizontal: Table:

Vertical: Table:

Figure 7.4–12 Model, Source Definition dialog (Directivity)

The directivity in a direction between the ’horizontal’ (local xz−plane) and ’vertical’ (local yz−plane) is interpolated between the values from the two tables for these directions. Hence, if the two tables are different, an ellipsoidal directivity can be produced, around the principal axis of the source, with the local x− and y−axes as its major and minor axes.

If no directivity table is given, the source is considered omnidirectional.

7.4.3.5 Time delay

Sources can be given a user–defined time delay (or ‘alignment’) in order to model electronic delay units as often applied in public address systems. If no value is given, the source is considered to emit at the same instant as (or ‘in phase with’) the first source defined with zero time delay.

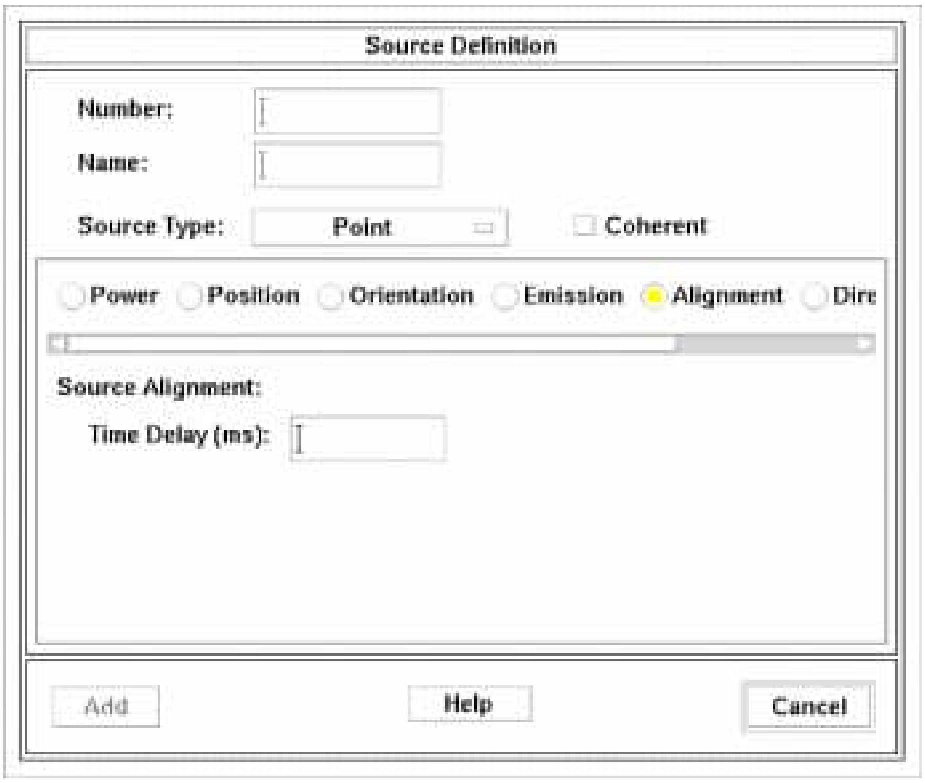
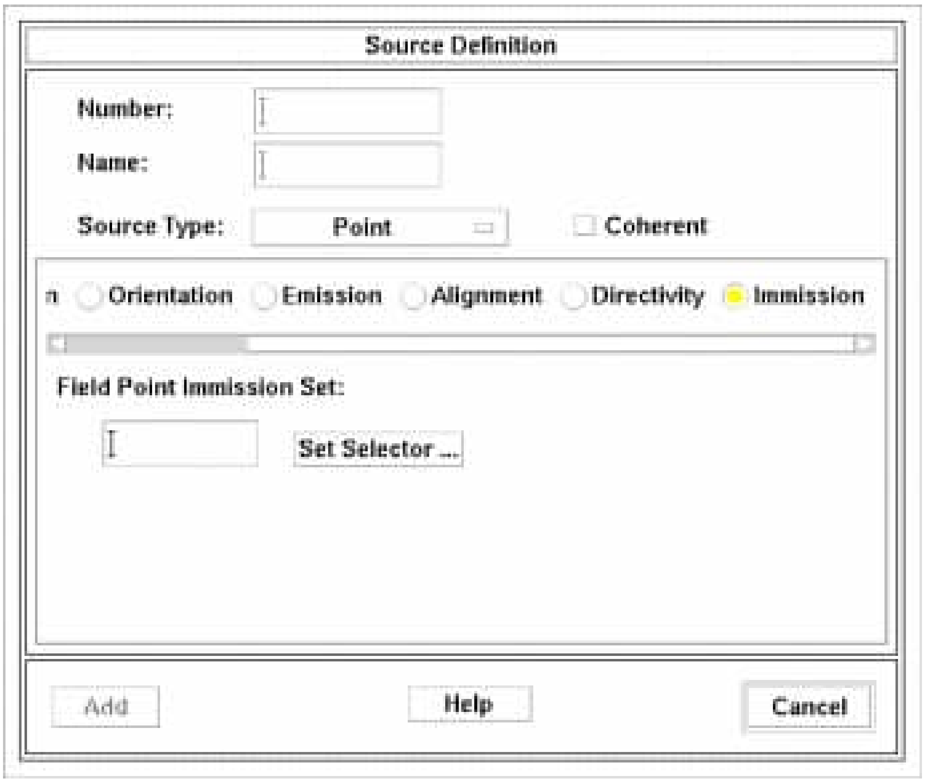


Figure 7.4–13 Model, Source Definition dialog (time delay)

7.4.3.6 Immission zone

Sources can be given an immission zone, defined by a set of field points. If so, only field points belonging to this set will be given sound contributions originating from the source. This feature is only of use in multi–domain situations: it enables calculations to benefit from the speed of approximate beam tracing, without suffering from poor accuracy in corners due to the ray ‘leakage’ effects (see *Approximate beam–tracing*, Section 3.1.5 on Page 3–9). Enter a set of field points in the dialog field, using the normal item–reference methods, selection from a previously–defined set, or graphical picking.



Source Definition

Number:

Name:

Source Type: Point

□ Coherent

n ○ Orientation ○ Emission ○ Alignment ○ Directivity ● Immission

Field Point Immission Set:

Figure 7.4–14 Model, Source Definition dialog (Immission zone)

7.4.3.7 Coherence *vs* incoherence

Sources are labeled coherent or incoherent. This has an impact on the calculated pressure distributions. Coherent means including phase information. Incoherent means without phase. Coherent sources and incoherent sources can be mixed in one calculation. See *Phase ray−tracing*, Section 3.4 on Page 3−25 for more information. STI calculations usually have a combination of coherent source(s) (to model the speech signal) and incoherent source(s) to model the noise signal. If all sources are coherent, only late reflections from the coherent sources, and badly−adjusted time delays, will affect the STI. For a calculation of STI with Storelevel = 2 (see *Histogram parameters*, Section 9.5.4 on Page 9−15) there must be at least one coherent source to represent the speech signal.

7.4.4 To create a source

Choose **Sources...** from the **Model** menu. Click on **New** to create a new source. Fill in the appropriate data, selecting dialog sub−regions with the related toggle buttons. All data remain 'volatile' and can be changed until the source is saved. Click on **Add** to save the source in the model which is currently open and active. Finally, **Close** the dialog when all source definitions are done.

7.4.5 To edit a source

Choose **Sources...** from the **Model** menu. First, select the source by highlighting it in the list of defined sources, then click on **Edit**. Make the changes to the data, **Change** to apply them, then **Close** the dialog.

7.4.6 To delete a source

Choose **Sources...** from the **Model** menu. First, select the source by highlighting it in the list of defined sources, then click on **Delete**, then **Close** the dialog.

7.4.7 Displaying sources

An omnidirectional source (*ie* a point source with an empty directivity table) is shown in the model viewer by means of a blue circle.

A directional source is displayed by means of a long blue line, representing the main axis orientation (local z–axis), and a short line representing the local x–axis.

A line source is represented by a blue line, and an area source by a blue polygon, in their defined locations.

A panel source is represented by a blue line, originating from the centroid of the panel and oriented in the direction of the radiating hemisphere (= normal to panel).

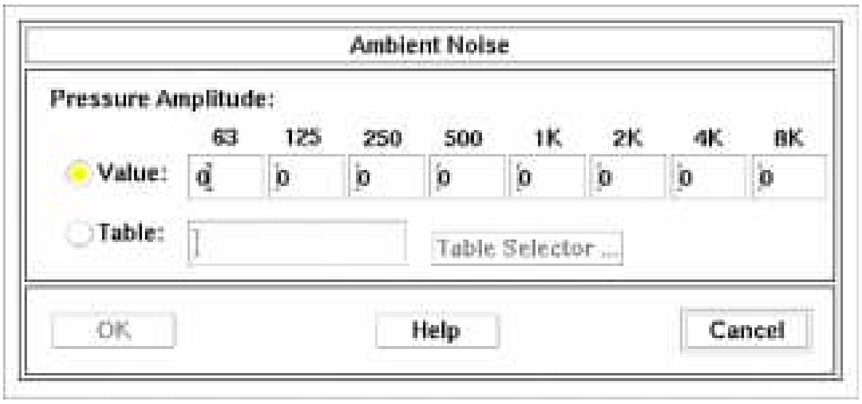
A transmitting source is represented by a red line, in the same way as a panel source. A second line from the origin of the source to a field point, indicates the connection to a value defining the excitation of the transmitting panel.

The sizes of the symbols used for displaying sources is a function of the maximum dimension of the model, except that it remains constant when the view zooms in/out.

7.5 Background noise

Background noise is the noise heard when all controllable sound sources are shut down. It can be useful for instance to take into account audience noise. The background noise level is position–independent.

Choose **Background Noise...** from the **Model** menu and enter values in the standard octave bands, or give a reference to a table. By default the background noise is 0dB for all frequencies.



Ambient Noise

Pressure Amplitude:

	63	125	250	500	1K	2K	4K	8K
● Value:	0	0	0	0	0	0	0	0

○ Table:

Figure 7.5–1 Model, Background noise definition dialog

7.6 Diffraction edges

7.6.1 General principles

RAYNOISE allows the modelling of diffraction phenomena across sharp edges, such as screens. The user has to select the edges to be used as diffraction edges.

Only first−order diffraction paths are taken into account. In other words, diffracted energy cannot reflect and/or diffract again and a ‘diffracted ray’ will ‘die’ at its first collision. However, diffraction contributions can originate from both sources and image sources. In cases with multiple diffraction along the same paths, it is the user’s responsibility to choose ‘equivalent’ diffraction edges.

- See *Diffraction modeling*, Section 3.3 on Page 3−19 for more information on the theoretical basis.

Diffraction edges can be mouse−picked from the mesh by selecting an element face. (See *Graphic picking tools*, Section 4.2.3.4 on Page 4−7). Faces are edges of polygon elements, for instance an element consisting of 4 nodes contains 4 edges. There is a distinction between free edges and connected edges. The free edges can be put into a set, by using ‘Envelope Generation’. (The envelope set is also interesting to detect holes in a supposedly−closed geometry model). The edges contained in this set are the ones which can be selected when defining diffraction edges. If you want diffraction edges to be defined on the basis of connected edges, set the Environment Variable INCLUDEALLFACES to TRUE when the mesh is imported. (See Section 11.8.3.40 on Page 11−28). In this case, all edges are available to be mouse−picked as diffraction edges.

7.6.2 To create a diffraction edge

Choose **Diffraction Edges...** from the **Model** menu. Select **New** to create a new diffraction edge. Mouse−pick an edge of an element. (An edge number can also be entered manually). Enter the name and/or diffraction−edge reference number if desired, in the sub−dialog. Click on **Add** to save the diffraction edge. **Close** the main dialog when done.

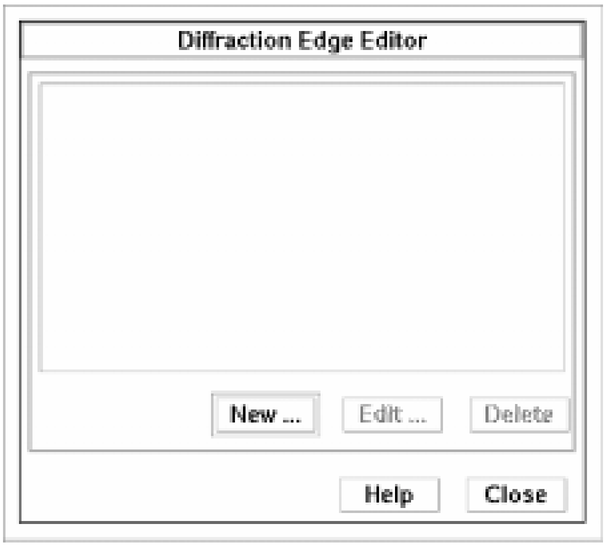


Figure 7.6–1 Model, Diffraction edge create/edit dialog

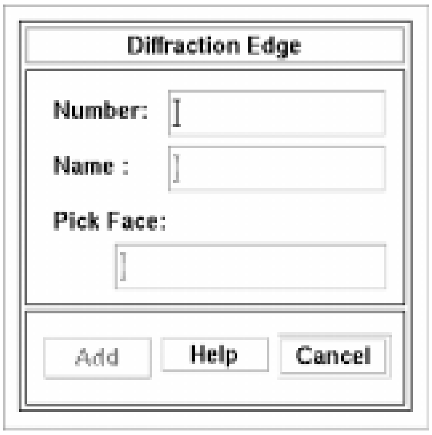


Figure 7.6–2 Model, Diffraction edge picking dialog

7.6.3 To edit a diffraction edge

Choose **Diffraction Edges...** from the **Model** menu. Select the diffraction edge by highlighting it in the list of current edges, then **Edit** to alter its data. Mouse–pick an edge of an element. (An edge number can also be entered manually). Enter the name and/or diffraction–edge reference number if desired, in the sub–dialog. Click on **Change** to save the changes. **Close** the main dialog when done. Note that if the edge number is changed manually, the previous edge selected for Edit will remain and a new edge will be created.

Hint: Be careful not to use identical edge names, or later identification may be difficult.

7.6.4 To delete a diffraction edge

Choose **Diffraction Edges...** from the **Model** menu. Select the diffraction edge by highlighting it in the list of current edges, then **Delete** to remove it. **Close** the main dialog when done.

7.7 Tables

7.7.1 The function of tables

In many problems, many characteristics such as source powers are a function of frequency. The definition of such parameters can be performed using tables. The **Model, Table** menu offers two possibilities:

- Creating tables manually
- Creating tables from a file

More information is also found in the related sections:

- *Using a table*, Section 7.7.5 on Page 7–34
- *Plotting a table*, Section 7.7.6 on Page 7–34
- *Creating a table using RCL commands*, Section 7.7.4 on Page 7–33
- *Import and Export of Tables*, Section 5.3.9 on Page 5–17 (regarding multiple tables from one file).

The use of tables is referred to in:

- *Material properties*, Section 7.3 on Page 7–6
- *Source properties*, Section 7.4.3 on Page 7–13

7.7.2 Manual definition of a table

7.7.2.1 Creating a table

Note! Manual creation of a table *using dialogs* is non–operational on PC. The corresponding commands are nevertheless available. It is in any case recommended normally to create tables by Importing from an external text file.

A table is created in the **Model, Tables, Manual...** dialog:

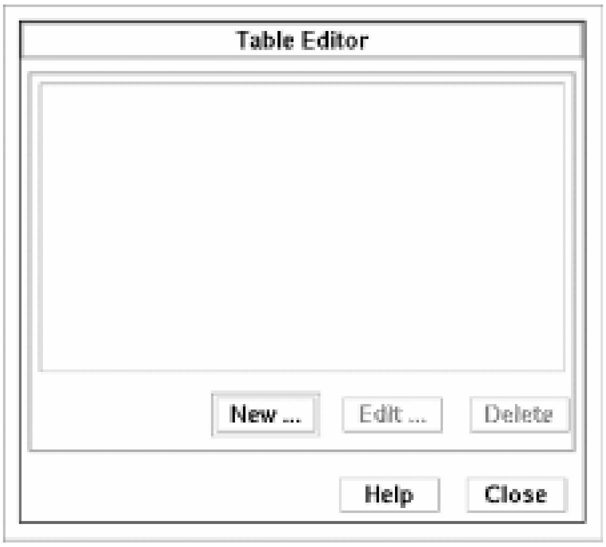
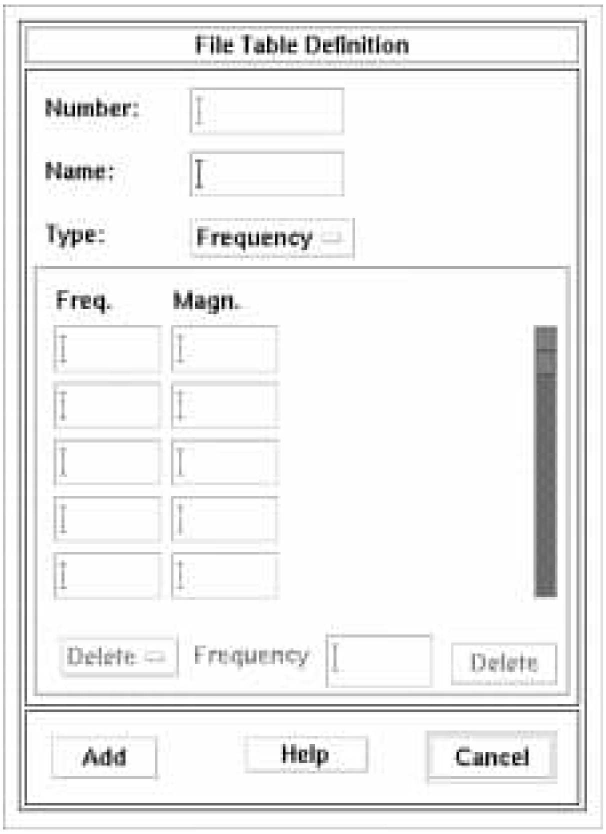


Figure 7.7–1 Model, Tables, Manual dialog

which shows a list of existing tables (if any). Selecting **New** opens an empty table−definition dialog:



Type: Frequency

Figure 7.7–2 Model, Table definition dialog (Frequency table)

which expects the following information:

- a table **number** for reference in further commands (if not given, defaults to last table number plus one)

- a table **name** for identification (if not given, defaults to *Type*Table # n)

- the **type** of table: possibilities are *Frequency*, *Directivity*, or *General*. See *Using a table*, Section 7.7.5 on Page 7–34, regarding the type of table for different applications.

- a sequence of (**frequency** / **value**) couples.

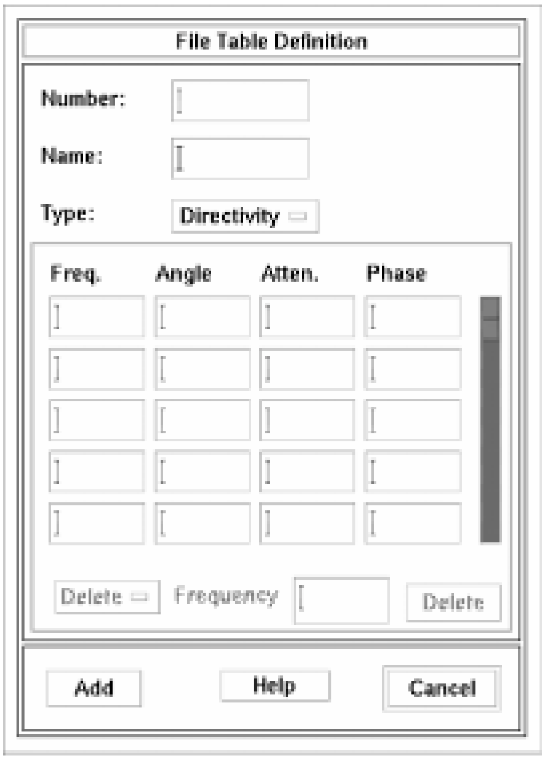
A new value is entered by giving a frequency value in the first entry field and a value in the one, two or three following fields on the row:

- **Frequency** tables require only one value: *magnitude* (a real number).

- **Directivity** tables require three values: *angle* from major axis (degrees), *attenuation* (reduction in intensity, dB) and relative *phase* (degrees).

- **General** tables require two values: *magnitude* (in the units where used, *eg* sound power level in dB) and *phase* (degrees).

The variants of the dialog for Directivity and General tables are shown below:



Type: Directivity

Figure 7.7–3 Model, Table definition dialog (Directivity table)

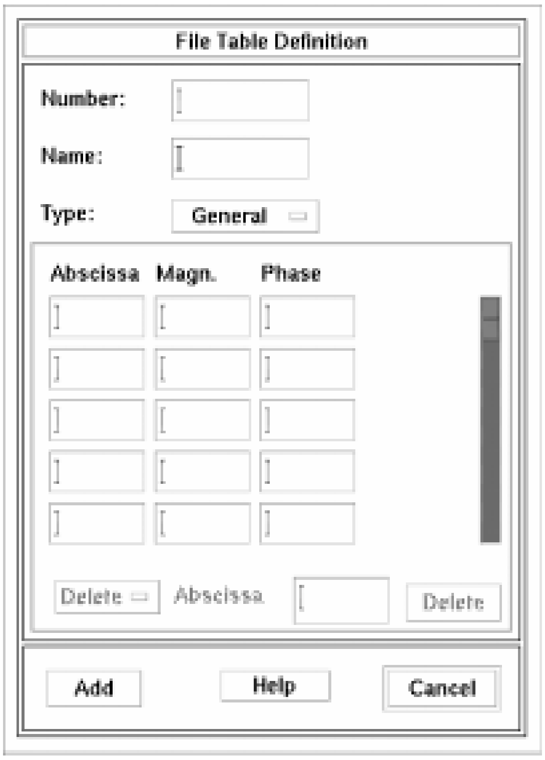


Figure 7.7–4 Model, Table definition dialog (General table)

The values entered are shown in the dialog and can be freely altered until the table is complete. If more entries are needed than the number of rows offered in the dialog, it can be scrolled, after adding more frequencies. Set the left–hand **Delete/Insert** toggle button to *Insert*, enter a new frequency, then click on the right–hand **Insert** button to add it to the table, after which the related value(s) can be entered in the row. Likewise, an unwanted frequency can be removed with *Delete*. Note that the frequency is inserted at the appropriate point, with frequencies in ascending order.

Click on the **Add** button to create the table, or **Cancel** to abort and close the dialog.

Note! If a table already exists, with the same table identification number specified in the new definition, the previously–defined table is deleted and a message is issued. It can be recommended to avoid this by using default table numbers and just giving new table names, unless such overwriting is definitely wanted.

The **General** type of table also corresponds to the tables (and frequency–response function results, *etc*) of SYSNOISE.
For instance, these data may have been output from SYSNOISE after processing with the **Combine** command in SYSNOISE, which has the equivalent in RAYNOISE (see Section 11.4 on Page 11–10).

7.7.2.2 Editing a table

Select an existing table by highlighting it in the **Model, Tables, Manual...** dialog and clicking **Edit**. The existing data will be displayed in the table–definition dialog and can be freely edited as described under *Creating a table*, except that the table number and type are fixed.

Click on the **Add** button to redefine the table, or **Cancel** to abort and close the dialog.

7.7.2.3 Deleting a table

Select an existing table by highlighting it in the **Model, Tables, Manual...** dialog and click **Delete** to remove it from the model. The table editor dialog list is updated.

7.7.3 Reading a table from a file

It is also possible to define a table using numerical values stored in a file using the **Model, Table, From file** dialog:

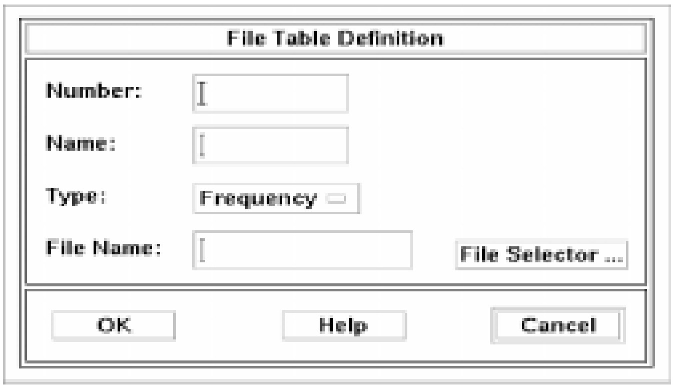


Figure 7.7–5 Model, Table definition from a file

The **number** and **name** of the table are entered in the relevant fields and the file containing the data is selected with the usual **File Selector...** action. The type of data in the file is selected with the *Frequency/Directivity/General* toggle, which also determines the number of values per row of the table. The specified file is read in free–format, with two, three or four numerical values per line. The first column contains the frequency and the other columns contain real values:

- **Frequency** tables require only one value: *magnitude* (a real number).
- **Directivity** tables require three values: *angle* from major axis (degrees), *attenuation* (reduction in intensity, dB) and relative *phase* (degrees).

- **General** tables require two values: *magnitude* (in the units where used, *eg* sound power level in dB) and *phase* (degrees).

The necessary number of values per line must be entered even if the values are zero (*eg*, in a General table, real values may be required, but the trailing zero imaginary part must also be entered).

For example, the following listing shows the contents of a data file to be read in as a General table:

```
  17   26.9   -4.2
  12   26.3    5.4
  15   26.5   -0.9
  11   24.2    8.7
  10   22.3   12.4
  19   42.1   -9.2
  20   38.6    0.
  16   25.5   12.1
  16   25.5   -3.4
  18   32.2   -5.1
  13   28.8    2.2
```

Note that the table can be edited once it is loaded into the RAYNOISE model. (See *Editing a table*, Section 7.7.2.2 on Page 7–32).

See also *Import and Export of Tables*, Section 5.3.9 on Page 5–17, for information on importing multiple tables from a single file.

A table can also be Exported to a file. See *Free format table files*, Section 15.1.5 on Page 15–10, for details of the file format when a table is written in RAYNOISE Free format.

7.7.4 Creating a table using RCL commands

An alternative to reading a table from a file is to create it using RCL commands (which may be held in a file). In this case, the file must have the necessary leading and trailing keywords to be read as a command file. For example, the following listing shows the contents of a command file which creates a table (Note that the table type *General* which can be selected in the dialog is termed Unknown in the RAYNOISE Command Language):

```
TABLE  NAME ´source123´
TYPE  UNKNOWN
  17   26.9   -4.2
  12   26.3    5.4
  15   26.5   -0.9
  11   24.2    8.7
  10   22.3   12.4
  19   42.1   -9.2
  20   38.6    0.
  16   25.5   12.1
  16   25.5   -3.4
  18   32.2   -5.1
  13   28.8    2.2
RETURN
```

7.7.5 Using a table

When defining materials data or sources, it is possible to refer to a table by choosing the **Table** switch, as in the **Model, Materials** and **Model, Sources** definition dialogs shown in previous sections. Table selection rather than direct data entry is activated by a toggle button, and then the table number can be entered, or the **Table selector...** button opens a dialog where a table can be picked by name: in this case, highlight the table and click **Select** to make the selection or **Done** to close the dialog. **New** opens the dialog for defining a new table.

Note that only tables of the appropriate type are offered by the selection dialog, depending on the application (*Frequency* tables for materials coefficients, source powers and other data requiring one value per frequency, *Directivity* tables for directivities, ...). During a calculation, frequency–dependent data such as source powers at successive calculation frequencies will be taken from the table. If the frequency lies between two table entries, the value is interpolated linearly, treating it as a complex value with interpolation of magnitude and phase if necessary. If the frequency is smaller (or greater) than the first (or last) entry in the table, the first (or last) entry value is selected. Similar interpolation is used in an angular sense for directivities.

The Combine tool (see Section on) can be used to manipulate tables and convert them from one form to another (eg from General type to Frequency type).

7.7.6 Plotting a table

The content of a table can be plotted using the **Postprocess, Table function** dialog.

- For further information, see *Results display, Plotting a table*, Section 10.3.7 on Page 10–39.

7.8 Units in RAYNOISE

7.8.1 General principles

Values representing physical data (coordinates, sound power, absorption coefficient, ...) are usually entered into RAYNOISE as real numbers. Unit descriptors (m, N, kg/m^3 ...) are never added to such values when entering them.

The values entered must therefore represent a *consistent* set of data. The usual assumption is that SI units are used, more specifically MKS (metres, kilograms, seconds). For some values (*eg* absorption and diffusion coefficients which are ratios) this is irrelevant. For other values, like coordinates, properties of the fluid medium, and sound powers, the user must take care to enter correctly−scaled values.

7.8.2 Changing units

It is recommended not to change the units system in your RAYNOISE model: preferably, scale the entered values to match the MKS system. In the case of coordinates, this can be done using the **Geometry, Scale** command (see *Scaling the mesh*, Section 6.2.7 on Page 6−8) after entering the acoustic mesh in another set of units. For all other data, the values must be scaled to the RAYNOISE system before being entered into the relevant dialogs.

However, expert users may decide to change the effective units used internally by RAYNOISE. This can be done, by considering the following assumptions:

- The unit of time is always assumed to be seconds (sec) and the unit of frequency is likewise always hertz (Hz) (and will be displayed as such).
- The sound speed in the fluid has the units of length/second: the sound speed and the length unit for coordinates then determine the arrival times of reflections.
- Sound pressure level and sound power level are related by the normal assumptions of classical acoustics in air in MKS units, unless the *Reference values* are changed (see Section 7.8.3 on Page 7−36).
- Phase angles (and geometrical angles, *eg* for emission) are always in degrees.
- Temperatures (for the *Medium properties for air*, see Section 7.2.2 on Page 7−4) are in degrees Celsius: the properties of air are looked−up in a table which gives sound speed in m/sec and air absorption in fraction/metre.

7.8.3 Reference values

The reference values used when calculating SPL dB levels and relating these to intensities and sound powers are Environment Variables, given in the *Set–up file* RAYNOISE.STP (see Section 11.8.4 on Page 11–39) and in the *Profile file* RAYNOISE.PRF file (see Section 11.8.5 on Page 11–40) which are read at start–up.

See also *Changing environment variables* (Section 11.8.2 on Page 11–19).

8

Data display

This chapter describes the methods for displaying the model, using the View menu, including:

8.1 View menu

8.1.1 General description

The menu associated with the **View** keyword in the Menubar contains command entries that activate dialogs where **Objects** and **Groups** can be selected for display in the graphics window.

The **Object** concept refers to a RAYNOISE entity that is considered as a single item for display in the graphics window. The following RAYNOISE entities belong to the objects category:

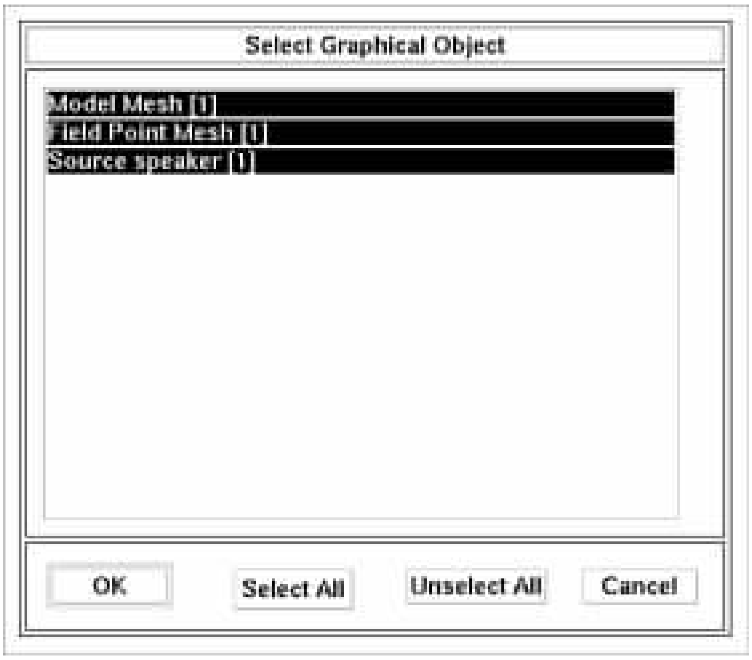
- Acoustic Mesh
- Field Point Mesh
- Diffraction edges
- Sources

The **Group** concept refers to a collection of RAYNOISE entities that are considered to form a single graphical item for display. The following RAYNOISE entities belong to the group category:

- Sets (collections of Nodes, Elements, Faces or Field Points)
- Materials
- Layers (collections of Elements)

8.1.2 Selection of *Objects*

The **View, Objects...** entry activates a dialog where the objects to be displayed are selected by highlighting the corresponding entries in a list of available objects. Figure 8.1–1, for instance, shows the **Objects** dialog where the Acoustic Mesh has been selected for display along with the Field Point Mesh and a Source.



Select Graphical Object

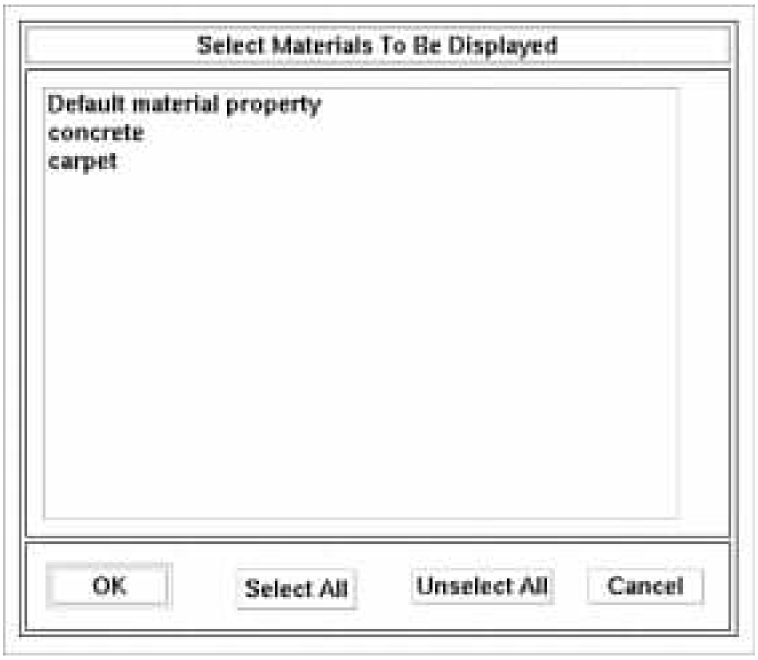
Model Mesh [1]
Field Point Mesh [1]
Source speaker [1]

Figure 8.1–1 View, Objects... dialog

8.1.3 Selection of *Sets, Layers, Materials* and *Edges*

8.1.3.1 General dialog layout

The **Sets... Materials... Layers...** and **Diffraction Edges...** entries activate similar dialogs where the corresponding item(s) can be selected. Figure 8.1–2 illustrates this for the **Materials...** dialog.


Select Materials To Be Displayed

Default material property
concrete
carpet

Figure 8.1–2 View, Materials... dialog: any combination of materials can be selected for display

All dialogs contain four buttons at the bottom:

- ***OK***: confirms the present selection
- ***Select All***: selects all listed entries
- ***Unselect All***: deselects all listed entries
- ***Cancel***: cancels the selection operation

8.1.3.2 Sets

View, Sets... enables the display of sets. The set(s) to be displayed are selected from the sub–dialog, which shows the currently–defined sets: a combination of sets of any type is permitted. The sets are colored using a basic color palette, related to the sequence in which they were defined in the model. The color used for any set can be changed, using **Display, Group Colors...** with group type **Sets**. (See Group colors, Section 12.1.10 on Page 12–14).

8.1.3.3 Layers

Layers are selected and displayed in the same manner as for Sets. Their colors can also be changed in the same way. (See Group colors, Section 12.1.10 on Page 12–14).

8.1.3.4 Materials

View, Materials... enables the display of materials. The material(s) to be displayed are selected from the sub–dialog, which shows the currently–defined materials. The materials are colored using a basic color palette, according to the sequence in which the materials were defined in the model. The color used for any material can be changed, using **Display, Group Colors...** with group type **Materials**. (See *Group colors*, Section 12.1.10 on Page 12–14).

8.1.3.5 Diffraction Edges

View, Diffraction Edges... enables the display of diffraction edges. The edge(s) to be displayed are selected from the sub–dialog, which shows the currently–defined edges. The edges are colored using a basic color palette, according to the sequence in which they were defined in the model. The color used for any edge can be changed, using **Display, Group Colors...** with group type **Edges**. (See *Group colors*, Section 12.1.10 on Page 12–14).

8.1.4 Hide All Groups

The **Hide All Groups** entry of the **View** menu deactivates all the groups selected for display: all groups become invisible and only Objects selected for display appear in the graphics window.

is the corresponding tool bar button, *which also removes all results displays*.

9

Calculation procedures

This chapter describes the calculation procedures of RAYNOISE, including:

9.1 Overview of calculation procedures

RAYNOISE offers 6 calculation procedures, also activated by the following commands:

- REVERBERATION: statistical reverberation calculation
- MAPPING: calculation of pressure contributions, with optional echogram saving in user−defined points
- POSTPROCESS: calculation of acoustic quantities in field points and echograms at field point positions, derived from the mapping data (=raw results)
- RESPONSE: calculation of complex, narrow−band frequency response functions at field points, derived from the mapping data
- BINAURAL: calculation of binaural impulse response, based on the mapping data
- CONVOLVE: convolution of sound material with binaural impulse response

While REVERBERATION and CONVOLVE are independent operations, POSTPROCESS, RESPONSE and BINAURAL need the raw results from MAPPING to take any effect.

9.2 Statistical Reverberation Time calculation

9.2.1 Purpose

Before beginning detailed studies of the spatial distributions of acoustic parameters such as Definition, Clarity, *etc*, it is often desirable to have a good assessment of the overall reverberation time of the room, independent of position. A position–independent reverberation time can be calculated using formulas from statistical acoustics. See *Statistical reverberation time*, Section 3.2 on Page 3–17 in the chapter on *Modeling principles*, for more information.

9.2.2 Procedure to calculate statistical RT

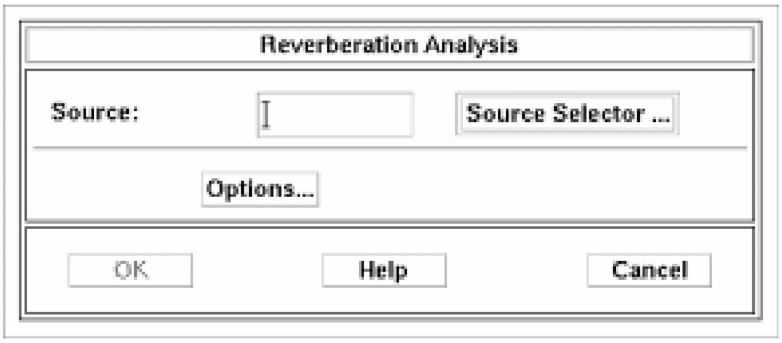


Figure 9.2–1 Analysis, Reverberation dialog

A statistical reverberation calculation is initiated by **Analysis, Reverberation...** It needs only one complement, selection of a source. This defines one source that will be used as ray generator. The ray–tracing process thus started is only sensitive to the order of reflection and the number of rays. See *Calculation parameters*, Section 9.5 on Page 9–13, for more information.

9.2.3 Extracting reverberation time results

The results of the statistical RT calculation are tabulated in the Echo Window, with **Inquire, Reverberation.** An example of the output is shown below:

```
REVERBERATION
-------------------------------------------------------------------
 Total Area (m2)      :  2583.5
 Volume (m3)          :  4914.7
 Mean Free Path (m)   :     7.6
 Relative variance    :     .74
 Number Mat_id       Name        Area(m2)  A-weight  S-weight
      1      1 CONCRETE            1116.5      .43       .34
      2      2 WOODENPLATFORM       369.3      .14       .24
      3      3 DOUBLE-GLAZING        15.5      .01       .00
      4      4 PARQUETFLOOR         133.8      .05       .12
      5      5 CARPET               374.7      .15       .10
      6      6 CURTAINS              63.7      .02       .07
      7      7 PERSPERM2            510.0      .20       .14
                        63 - 125 - 250 - 500 -  1K -  2K -  4K -  8KHz
Sabine area (m2):    282.  295.  320.  505.  688.  752.  785.  807.
mean absorption :    .138  .142  .134  .186  .233  .253  .255  .260
 RT(Sabine) :        2.21  2.16  2.29  1.65  1.31  1.21  1.20  1.17
 RT(Eyring) :        2.06  2.00  2.13  1.49  1.15  1.05  1.04  1.01
 RT(statis) :        2.16  2.11  2.24  1.59  1.25  1.15  1.14  1.12

Preferred Reflection Order :     97
-------------------------------------------------------------------
```

*Figure 9.2–2 Output from **Inquire, Reverberation***

‘A–weight’ refers to the fraction of the total area of each material type, according to its area. ‘S–weight’ is the weighted fraction of the total area, of each material type, according to the number of ray collisions on that material type. A major difference between these two weighted fractions indicates that certain elements take more or less part in the reverberation process than their area would indicate, for instance because certain elements are partly obstructed by others.

The wall diffusion coefficients are not accounted for, as the statistical reverberation theory assumes a fully–diffusive behavior of the walls.

The **Inquire, Reverberation** output also includes the information Preferred Reflection Order which indicates the order which should be used in a full Mapping calculation in order to obtain high–accuracy echogram results. The value is entered in the Options dialog of the **Analysis, Mapping** command. This Preferred Reflection Order is derived from the highest RT_{60} reverberation time, *RT*, across all frequencies, the mean free path, *l*, and the speed of sound, *c*, using: $Order = c.RT/l$

Statistical reverberation times are calculated using the statistically–weighted fractions of the various material types, not the simple area fractions.

Hint: If you re–assign other material types to the elements (Model, Assign... menu, command ASSIGN) the reverberation times will be updated automatically, immediately, without having to re–initiate a reverberation calculation.

Reverberation times and related results are saved in the permanent database, when Saving.

See *Extracting information*, Section 4.8 on Page 4–17, for more general information about the **Inquire...** menu.

9.3 Mapping calculation

9.3.1 Purpose

During a mapping calculation, the Conical Beam Method or the Triangular Beam Method is used to calculate pressure contributions at field points, as functions of time, space, frequency and angle of incidence. Results from Mapping calculations are needed to enable the second step, called Postprocessing, *ie* the derivation of acoustical parameters such as SPL, STI, EDT *etc*. (See *Postprocess calculation*, Section on Page 9–10 for more information). Mapping results are also needed for the *Frequency response calculation*(see Section 9.6 on Page 9–20) and the *Binaural impulse response calculation* (see Section 9.7 on Page 9–22).

9.3.2 Mapping calculation procedure

9.3.2.1 Mapping parameters

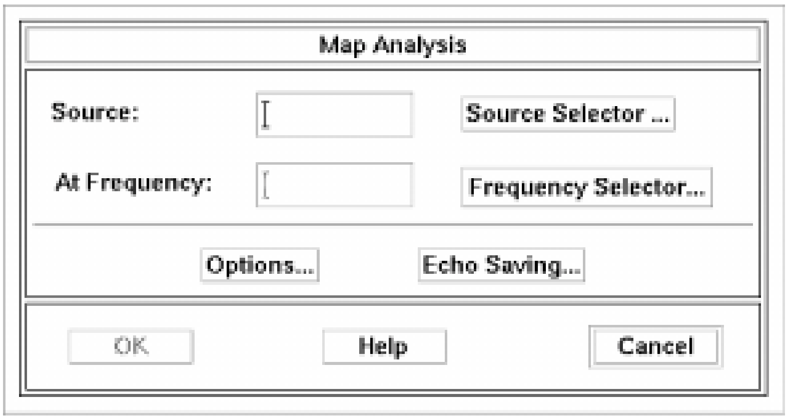


Figure 9.3–1 Analysis, Mapping, main dialog

Analysis, Mapping needs a source selection and a frequency selection. A maximum of 128 frequencies can be chosen for one mapping calculation. If no frequencies are specified (void text field), by default the 8 standard octave–band center frequencies are picked, from 63Hz to 8kHz. For each of the selected sources, a cone– or pyramid–tracing calculation is performed. All frequencies are processed in parallel (since ray paths are independent of frequency) except when wall diffusion is explicitly required. In that case a randomly–based ray–tracing calculation will be initiated for each frequency separately, in addition to the deterministic ray–tracing calculation which simultaneously models specular reflections for all frequencies. (See *Calculation parameters*, Section 9.5 on Page 9–13, for more information).

9.3.2.2 Echo saving option

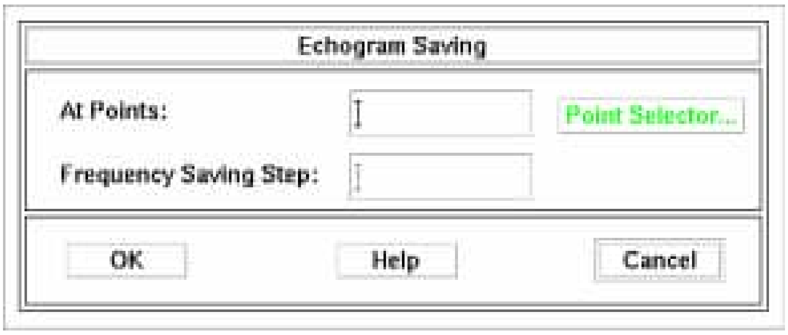


Figure 9.3–2 Analysis, Mapping, Echo saving dialog

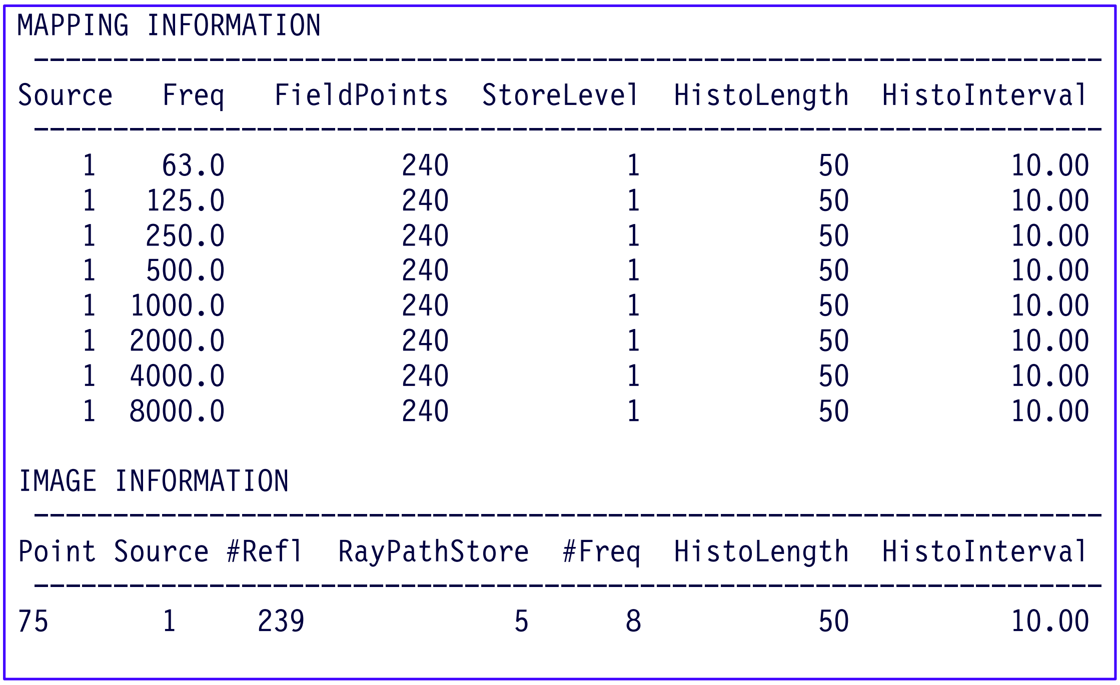
When detailed analysis of the echogram is required in a local point, the **Echo Saving** option can be used. Enter one or more field point numbers, or select them with the mouse using the graphical picking tools.

See *Echogram parameters*, Section 9.5.3 on Page 9–14, and *Histogram parameters*, Section 9.5.4 on Page 9–15, for more information on stored reflection order and histogram dimensions.

9.3.3 Extracting mapping results

The results of a mapping calculation are stored in the database. A directory of the results can be found with **Inquire, Mapping**. It is grouped in two parts:

- MAPPING INFORMATION, *ie* pressure as a function of time in histogram style in all field points, kept separately for each source and each frequency.
- IMAGE INFORMATION, *ie* full echogram/histograms as functions of frequency, stored per selected field point and per source, including ray trajectories for individually–stored reflections.



```
MAPPING INFORMATION
 -------------------------------------------------------------------
Source    Freq   FieldPoints  StoreLevel  HistoLength  HistoInterval
 -------------------------------------------------------------------
    1     63.0           240           1           50          10.00
    1    125.0           240           1           50          10.00
    1    250.0           240           1           50          10.00
    1    500.0           240           1           50          10.00
    1   1000.0           240           1           50          10.00
    1   2000.0           240           1           50          10.00
    1   4000.0           240           1           50          10.00
    1   8000.0           240           1           50          10.00

IMAGE INFORMATION
 -------------------------------------------------------------------
Point Source #Refl  RayPathStore  #Freq  HistoLength  HistoInterval
 -------------------------------------------------------------------
75        1     239            5      8           50          10.00
```

Figure 9.3–3   Output from ***Inquire, Mapping***

See *Database operations*, Section 5.2.4 on Page 5–6, regarding how to delete mapping results from the database.

9.4 Postprocess calculation

9.4.1 Purpose

The postprocess calculation derives results, *ie* acoustical parameters such as SPL, STI and EDT, from the mapping results stored in the database. Naturally, results can only be derived at frequencies and for sources for which the mapping results are available.

9.4.2 Postprocess calculation procedure

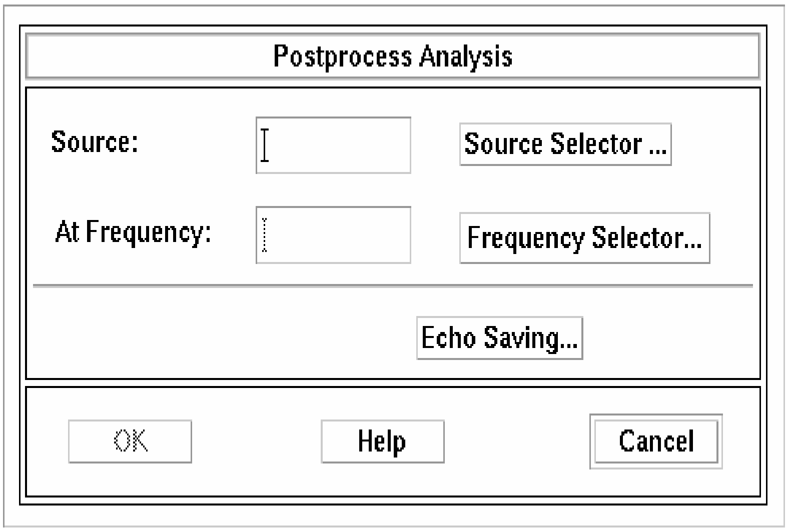


Figure 9.4–1 Analysis, Postprocess dialog

The command POSTPROCESS needs a user–specified source selection, as well as a frequency selection. The mapping results for each of the selected sources, for each selected frequency, must be present in the database: if not, a warning message will be given. The dialog is opened by **Analysis, Postprocess...** The source and frequency settings from the last use of the Mapping dialog in the current RAYNOISE session are recalled as default entries (often, the Mapping and Postprocess commands require the same information).

The acoustical parameters which are calculated by Postprocess relate to the selection of sources made in that Postprocess operation: this may be a sub–set of all the sources in the model. The source properties (power, time delay, ...) used in the Postprocess are those valid at that time. See further remarks under *Efficient use of mapping and postprocess*, Section 9.4.3 on Page 9–11.

Detailed echogram/histograms can be derived from the IMAGE section of the Mapping results using the **Echo Saving** option, if the required echo results were stored during the Mapping calculation. Enter one or more field point numbers, or select them graphically. Usually, point numbers are automatically pre−entered from the previous use of the Mapping dialog. The Frequency Saving Step (FSS) determines the frequency values at which echogram/histograms are prepared for graphical display: FSS = 1 means every frequency, FSS = 2 means every other frequency, etc.

9.4.3 Efficient use of mapping and postprocess

9.4.3.1 General cases

While the Mapping calculation is computationally intensive (it contains the actual ray−tracing process) the Postprocess calculation is relatively very fast.

The advantage of this two−step approach is that the contribution of each source to the resulting noise map is stored separately. In that way, the mapping does not have to be re−calculated in order to know the influence on the acoustical field of only some of the sources. Likewise, it is possible to alter the sound power levels, time delays and (in)coherence of the sources after the Mapping step, without a repeat of the Mapping being needed. Only a re−run of the Analysis, Postprocess sequence is needed to get updated results.

9.4.3.2 Particular case with transmission sources

In the particular case where transmission sources have been defined, it is possible to carry out first the Mapping and Analysis, Postprocess field−points steps for the ’primary’ sources (those representing real acoustic sources, as opposed to the panel transmission sources). Then, the Mapping step can be repeated for the transmission source(s) only, in which step the powers of the transmission source(s) will be derived from the field−point results already calculated. Finally, the Analysis, Postprocess step can be performed for all sources, to get the total result at all field points.

This may be useful, for instance, in the case of noise break−out from a building: first, carry out the interior calculations, ignoring the transmission sources; make any iterations necessary (changes to materials, primary sources, *etc*); finally, run a Mapping for the transmission sources and Postprocess all results together.

Such a procedure can be expected to work correctly in the case of two distinct domains (*ie*, a closed interior) and can be computationally efficient, especially if the sources have immission limited to certain field−point sets. However, in the case of partly−closed, partly−open domains, diffraction and similar effects, the user should exercise great care to assess the source−receiver relationships and the necessary calculation sequence.

See *Sound power level of sources*, Section 7.4.3.1 on Page 7−13, for more information on transmission sources and *Immission zone*, Section 7.4.3.6 on Page 7−21, regarding how to limit source immission.

9.4.4 Extracting the results of Postprocess

After the Postprocess step, results can be displayed graphically (see *Results display* , Chapter 10) in the form of color maps and/or spectra, and the echogram display can now be used (see *Echogram display*, Section 10.4 on Page 10−41).

- Results values at field points can be listed using **Inquire, Results**.
- Detailed echograms can be listed using **Inquire, Echograms**.

9.5 Calculation parameters

9.5.1 Dialog areas and general procedure

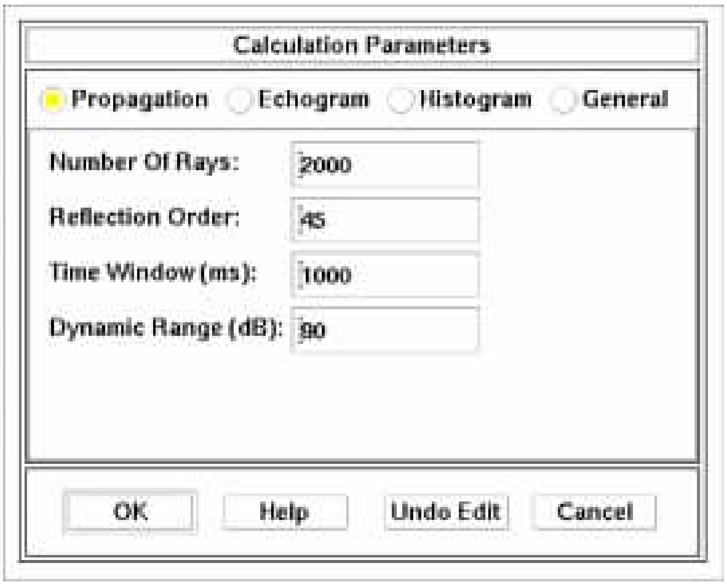
The Calculation Parameters dialog sets various parameters regarding the beam propagation model, accuracy of results, amount of detail in results, *etc*.

The Calculation Parameters dialog is opened by the **Options**... button on the relevant higher–level dialogs. The dialog has four areas, activated by the radio buttons at the top:

- Propagation
- Echogram
- Histogram
- General

After entering the desired values, they are applied by pressing **OK**, which returns to the higher–level dialog. **Cancel** returns without applying the values. **Undo Edit** resets the entry fields to their previous values (*eg* Environment Variables from the RAYNOISE.STP or RAYNOISE.PRF files, see Section 11.8 on Page 11–19 or values applied earlier in this RAYNOISE session) after which OK or Cancel can be used.

9.5.2 Propagation parameters



● Propagation ○ Echogram ○ Histogram ○ General

Number Of Rays: 2000

Reflection Order: 45

Time Window (ms): 1000

Dynamic Range (dB): 90

Figure 9.5–1 Propagation parameters dialog

One parameter just influences accuracy (and conversely, calculation time):

- The **Number Of Rays** per source = number in a complete sphere, even if the source is limited in extent by Emission Angles. It is applied uniformly to all sources.

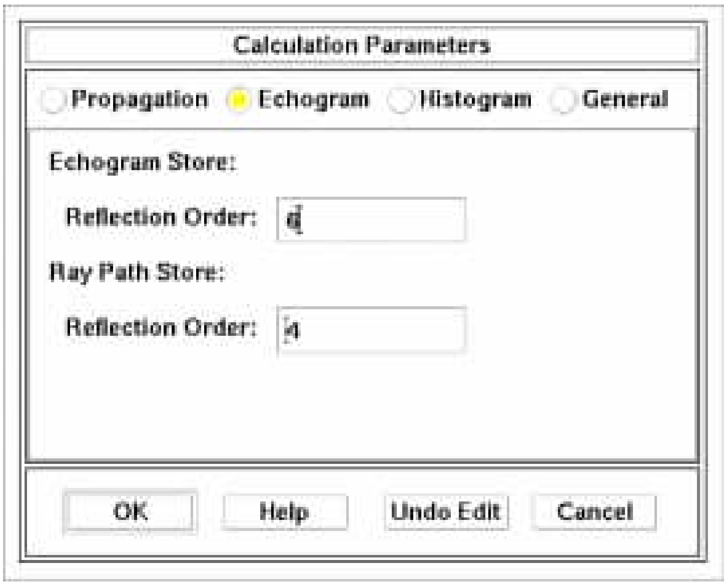
Three criteria have more complex influences. They are used to decide when to abandon the propagation of a cone/pyramid and to begin tracing a new one:

- The current order of reflection exceeds **Reflection Order**
- The current travel time exceeds **Time Window**
- The current energy content of the cone/pyramid is lower than a threshold value, determined by the initial energy content minus **Dynamic Range**.

A Reflection Order of 0 implies calculation of the direct energy only.

See also *Effect of calculation parameters on accuracy*, Section 9.5.8 on Page 9–18.

9.5.3 Echogram parameters



Calculation Parameters

○ Propagation ● Echogram ○ Histogram ○ General

Echogram Store:

Reflection Order: 0

Ray Path Store:

Reflection Order: 4

Figure 9.5–2 Echogram parameters dialog

These parameters are only relevant to the optional **Echo Saving** procedure in Mapping and Postprocess calculations.

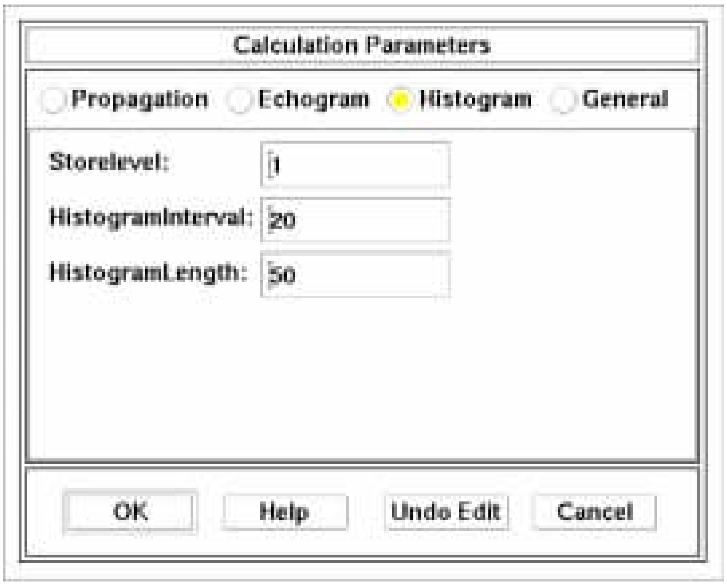
- **Echogram Store Reflection Order**: indicates up to what reflection order the reflections are stored explicitly in the echogram as discrete reflections with precise arrival times and frequency–dependent amplitudes. Above this order, reflections are 'integrated' into the histogram. It is meaningless to enter a value greater than the Reflection Order.

- **Ray Path Store Reflection Order**: indicates up to what reflection order the ray trajectories of individual reflections (*ie* the coordinates of the ray intersections at the reflecting surfaces) are stored explicitly in the echogram. This number must always be less than or equal to the Echogram Store Reflection Order and less than or equal to 10 in absolute terms. (Default value is normally 5).

See also *Effect of calculation parameters on data storage*, Section 9.5.7 on Page 9–17.

See also *Effect of calculation parameters on accuracy*, Section 9.5.8 on Page 9–18.

9.5.4 Histogram parameters



Calculation Parameters

○ Propagation ○ Echogram ● Histogram ○ General

Storelevel: 1

HistogramInterval: 20

HistogramLength: 50

Figure 9.5–3 Histogram parameters dialog

These parameters relate to the dimensions of the histogram as used during Mapping and Postprocess calculations:

- **HistogramInterval**: ’bin’ width of the histogram in milliseconds, typically 10 to 20ms.

- **HistogramLength**: number of time intervals (’bins’) to cover the full length of the transient response.

If the temporal behavior is of no particular interest, histograms can be omitted from being stored in the database, saving a lot of RAM and disk space. If so, transient parameters (Definition *etc*) will have zero values after Postprocess. On the other hand, if the Speech Transmission Index (STI) needs to be assessed accurately, histogram information is insufficient: Modulation Transfer Functions (MTF) need to be calculated and stored in the database as well.

The above options are controlled with the **Storelevel** parameter:

Storelevel value	Data stored
0	no storage of transient behavior
1	storage of transient behavior (= histogram)
2	storage of transient behavior and MTFs for precise STI computation

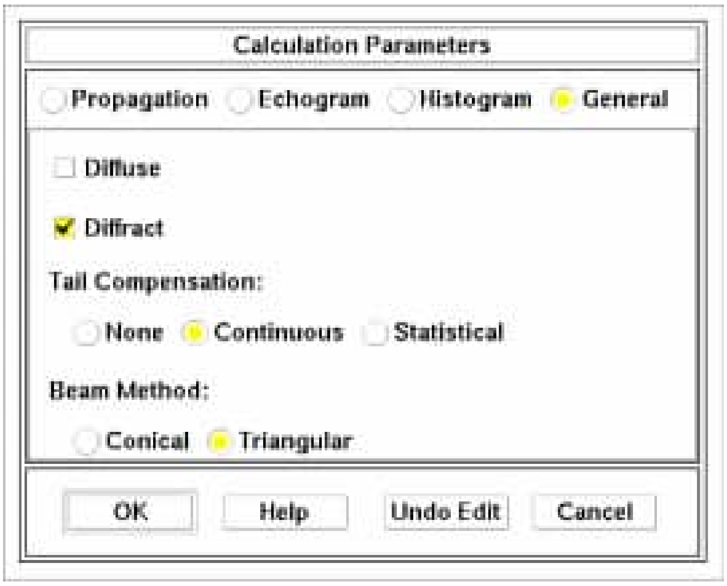
The use of Storelevel 2 implies a requirement for precise STI calculation which in turn implies that at least one source is defined to be Coherent, to represent the ’speaker’. (See also *Source properties, Coherence vs incoherence*, Section 7.4.3.7 on Page 7–22).

The availability of Storelevel 1 or higher is only possible if the module ‘Transient’ is installed. (Installation– and license–dependent).

See also *Effect of calculation parameters on data storage*, Section 9.5.7 on Page 9–17.

See also *Effect of calculation parameters on accuracy*, Section 9.5.8 on Page 9–18.

9.5.5 General parameters



Calculation Parameters

○ Propagation ○ Echogram ○ Histogram ● General

□ Diffuse

☑ Diffract

Tail Compensation:

○ None ● Continuous ○ Statistical

Beam Method:

○ Conical ● Triangular

Figure 9.5–4 General parameters dialog

The General calculation parameters relate to optional procedures in the algorithm to be used.

- **Diffuse** toggle: when *On*, the program models diffuse reflections by means of the hybrid deterministic–random method (see *Diffuse reflections*, Section 3.5 on Page 3–27).

- **Diffract** toggle: when *On*, the program compensates for diffraction by applying the inverse Kurze–Anderson attenuation on the user–defined diffraction edges (see *Diffraction modeling*, Section 3.3 on Page 3–19).

- **Tail Compensation, None/Continuous/Statistical** toggles: applies the continuous or statistical or no echogram tail compensation (see *Tail correction*, Section 3.1.6 on Page 3–11).

- **Beam Method**: switches between Conical and Triangular beam–tracing. (See *Hybrid methods*, Section 3.1.4 on Page 3–6, for a comparison of the methods).

See also *Effect of calculation parameters on data storage*, Section 9.5.7 on Page 9–17.

9.5.6 Calculation parameters as environment variables

Calculation parameters are also Environment Variables and can therefore be set independently of the calculation dialogs, using the **Tools, Environment Var.** dialogs (see *Changing environment variables*, Section 11.8.2 on Page 11–19). The values or switches set in the calculation parameter dialogs or in the environment variables dialog are inter–linked. Note that this means that if a calculation parameter is set in the calculation parameters dialog, and subsequently Store In User Profile is executed in the environment variables dialog, the values set in the calculation parameters dialog will be stored in the RAYNOISE.PRF Profile file. (See Section 11.8.5 on Page 11–40).

9.5.7 Effect of calculation parameters on data storage

The Echogram parameters and the Histogram parameters interact in the storage and presentation of impulse response results in the following way:

- An echo arriving after no more reflections than the *Echogram Store Reflection Order* is stored individually, provided its arrival time is less than *Time Window* and its final energy is greater than its initial energy less *Dynamic Range*. Such an echo appears as an individual pulse in the Echogram display.

- Other echoes may arrive at around the same time as the one described above, but after more reflections than the *Echogram Store Reflection Order*. They are added to the Histogram ‘bin’ corresponding to their arrival time. The histogram energy is displayed in the Echogram display as bars with a finite width on the time axis, plotted behind the individual echo pulses. If *Echogram Store Reflection Order* is set to zero, only a histogram will be stored and displayed.

- Diffuse rays may produce echoes arriving at any time in the Time Window. Because there can be a large number of such diffuse reflections (*ie*, large data volumes) and because they are in any case statistical in nature, with a random initiation, they are always added to the Histogram storage.

For a given model, with a given number of field points and frequencies, the volume of data to be stored increases with the number of Histogram bins to be stored, and especially with the number of discrete echoes in Echograms.

See also *Effect of calculation parameters on accuracy*, Section 9.5.8 on Page 9–18.

See *Echogram displays*, Section 10.4 on Page 10–41, for more information on plotting echograms.

See also *Resource requirements*, Section 9.5.8 on Page 9–18.

9.5.8 Effect of calculation parameters on accuracy

9.5.8.1 In general

In general, a calculation will yield more accurate results, the higher the maximum order of the reflection. This is particularly so in cases with hard reflective linings, because then the energy content of reflections of high order (*eg* 20 or 30) can still be very significant. This effect may be greater if the source is Coherent, since explicitly–computed echoes are summed with phase effects included, whereas energy in histogram 'bins' or derived from a tail correction implictly lose phase information. If a Statistical Reverberation calculation is carried out before a Mapping, the Inquire, Reverberation command gives an indication of the recommended reflection order which should be used to get high–accuracy results. This is especially true if an auralization is to be performed. See *Extracting reverberation time results*, Section 9.2.3 on Page 9–4.

The accuracy also depends significantly on the number of rays emitted, although to a lesser extent. The higher the number of rays, the smaller the solid angle of the cones/pyramids and thus the smaller the errors due to edge effects (see *Approximate beam–tracing*, Section 3.1.5 on Page 3–9) but also the higher the computation time.

9.5.8.2 Accuracy of results based on the impulse response

The accuracy of the predicted reverberation time and other results derived from the impulse response results at a local receiver, depend strongly on the settings of HistogramInterval and HistogramLength. It is advisable to choose HistogramInterval and HistogramLength such that their product is approximately equal to the average reverberation time.

Furthermore, if HistogramInterval is too wide, a coarse integration will result. This may be acceptable if only certain acoustic quality results are of interest (for instance, Definition) and a compatible value is chosen for the HistogramInterval (for instance, 50 ms). If the response may be far from a standard exponential decay, or greater accuracy is required, HistogramInterval must be made narrower and a much larger number of Histogram 'bins' will be used. The default HistogramInterval is 10ms.

9.6 Frequency response calculation

9.6.1 Purpose

A frequency response calculation is a narrow–band computation of a Frequency Response Function (FRF). It relies on the IMAGE result section from the Mapping results. The procedure is particularly useful for narrow–band analyses of transfer functions between a coherent source and a receiver point.

The Frequency Response calculation procedure is only possible if the module 'Transient' is installed. (Installation– and license–dependent).

9.6.2 Frequency response calculation procedure

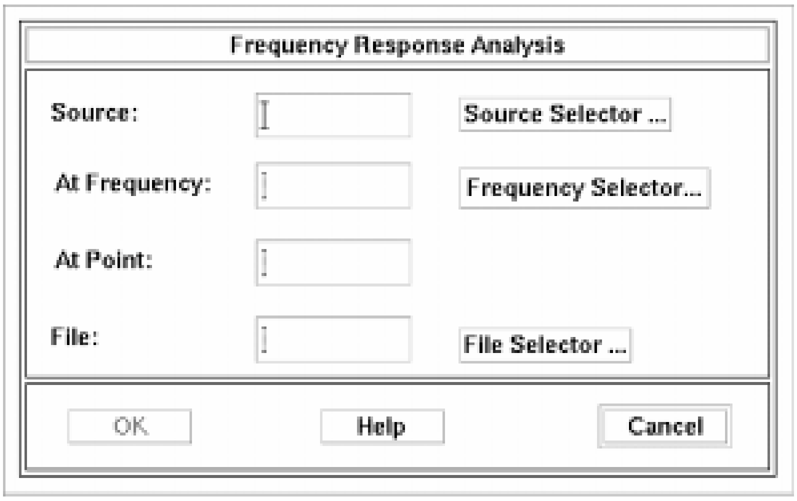


Figure 9.6–1 Frequency response dialog

Select **Response...** from the **Analysis** menu. Enter a field point number, a source selection, and a frequency selection.

The field point number must exist in the IMAGE result section of the Mapping results. Likewise, the selected sources must exist in the same results for that field point.

The frequency selection is arbitrary: it is not limited to the frequency selection chosen in the Mapping calculation. The *Echogram interpolation* technique (see Section 3.4.5 on Page 3–26) automatically interpolates between frequencies for which Mapping results exist, in order to give a full narrow–band analysis.

The process is controlled by the LinInterpolate (or LogInterpolate) parameter in the frequency–selection dialog. The linear/logarithmic toggle determines whether the progression of intermediate frequencies between each pair of adjacent master frequencies is linear or logarithmic on the frequency axis: the parameter is the number of divisions between the lower and upper master frequencies; in the linear case, the divisions are equal; in the logarithmic case, the divisions are logarithmic (*ie* a geometric progression) with the spacing automatically determined such that the requested number of slave frequencies are created between the lower and upper master frequencies.

- See *Frequency selection*, Section 9.9 on Page 9–26, for more details on defining frequency ranges and steps.

The results are written in an 'FRF file'. The format of the FRF file corresponds with the format used in the SYSNOISE program. The default file name is 'default.RES'.

Hint: The FRF can also be read by the **Combine** tool (see Section 11.4 on Page 11–10) of both RAYNOISE and SYSNOISE, for instance to merge an FRF calculated with FEM or BEM in SYSNOISE with the FRF from RAYNOISE. An FRF for the complete frequency range, including low–frequency wave phenomena and mid–/high–frequency effects, will result.

The FRF obtained from this procedure can be displayed using **Postprocess, File Function** (see *Frequency response function curves*, Section 10.3 on Page 10–28).

Remarks:

- An FRF contains complex pressures. The phase of each pressure contribution is derived from the time corresponding to the distance traveled by the ray and the initial phase at the source. Histogram results (as opposed to Echogram results) from a Mapping calculation, do not contain individual rays and their travel times (since they are integrated, within the short time intervals of the histogram 'bins') so phase cannot be determined. A random sampling operation results, which gives a pressure contribution with random phase (somewhat like white noise).
- As in the Postprocess calculation, source power levels and time delays may be altered, without having to re–do the Mapping calculation, and new frequency response functions calculated.

9.7 Binaural impulse response calculation

9.7.1 Purpose

A binaural impulse response calculation consists of transforming the echogram/histogram data into a Binaural Finite Impulse Response (BFIR) which is sampled at a user–defined sampling frequency and stored in a sound file. A BFIR is the core of the auralization process, which can make any source material audible taking into account any given physical and acoustical boundaries. See *Modeling principles, Auralization*, Section 3.7 on Page 3–32, and *Convolution*, Section 9.8 on Page 9–24, for more information.

The Binaural calculation procedure is only possible if the 'Auralization' module is installed. (Installation– and license–dependent).

9.7.2 Binaural impulse response calculation procedure

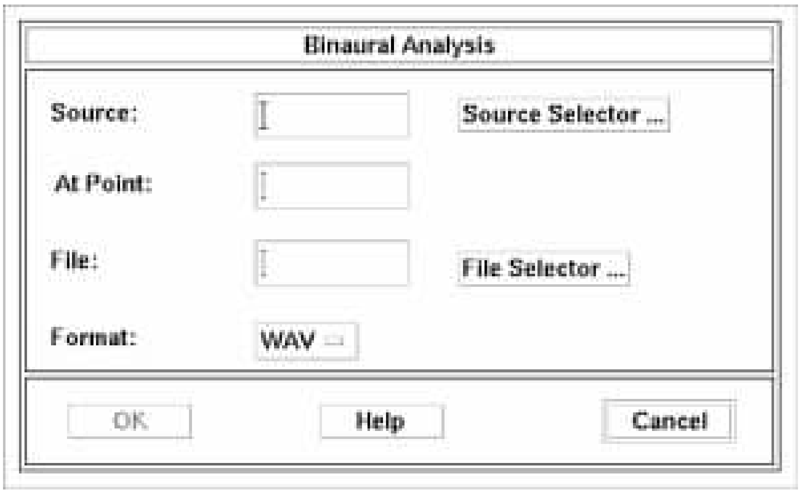


Figure 9.7–1 Binaural transformation dialog

Select **Binaural...** from the **Analysis** menu. Enter a field point number and a source(s) selection.. **OK** starts the processing, or **Cancel** to abort.

The field point number must exist in the IMAGE result section of the Mapping results. Likewise, the selected source(s) must exist in the same list for that field point.

Two parameters determine the BFIR:

- *Time length*, determined by the **Time Window** which was set in the Propagation parameters area of the Calculation Parameters dialog, called by **Analysis, Options...** (see Section 9.5.2 on Page 9–13).

- *Sampling frequency*, set by the Environment Variable SAMPLEFREQUENCY (see *Environment variables*, Section 11.8 on Page 11–19). Its default value is 44.1kHz, corresponding to CD–audio quality.

The sound file containing the BFIR can be stored in WAV, AIFF or AU format, selected from the **Format** option list.

Note! The Binaural calculation is produced using the ray–arrival information for all explicitly–stored (individual) rays, combined with Head Related Transfer Functions that are built into RAYNOISE. (These HRTFs are taken from published literature). The orientation of the nominal head is 'horizontal' (a line between the ears is parallel to the XY plane of the model) and 'looking' at the first source defined in the database. It is also assumed that the Z–axis is vertical upwards, for the purposes of establishing the left/right relationship. Therefore, to rotate the head to different positions, it is recommended to define the first source as a 'dummy' (*ie*, with negligible power) and then to change its position between one Binaural computation and another.

Hint: The normalization factor used when writing the BFIR sound file is displayed in the Echo Window. By default, normalization uses a variable value, so as to fill the maximum dynamic range of the 16–bit file, but a fixed normalization can be defined by the user with the Environment Variable NORMIRF (see Environment Variables, Section 11.8 on Page 11–19). Note that NORMIRF will not appear in the list of Environment Variables given by the **Tools, Environment Var.** dialog, unless it has been set previously. Therefore, to set it, use a command:

```
RAYNOISE> ENVIRONMENT NORMIRF value  RETURN
```

9.8 Convolution

9.8.1 Purpose

The convolution process enables the convolution of any source material stored in a sound file with a BFIR, *eg* as calculated in the Binaural impulse response calculation (see Section 9.7 on Page 9–22).. The result is a new sound file, which lets the listener hear how the original source material would be heard at the chosen location within the given physical and acoustical boundary conditions.

The Convolution calculation procedure is only possible if the 'Auralization' module is installed. (Installation– and license–dependent).

9.8.2 Convolution procedure

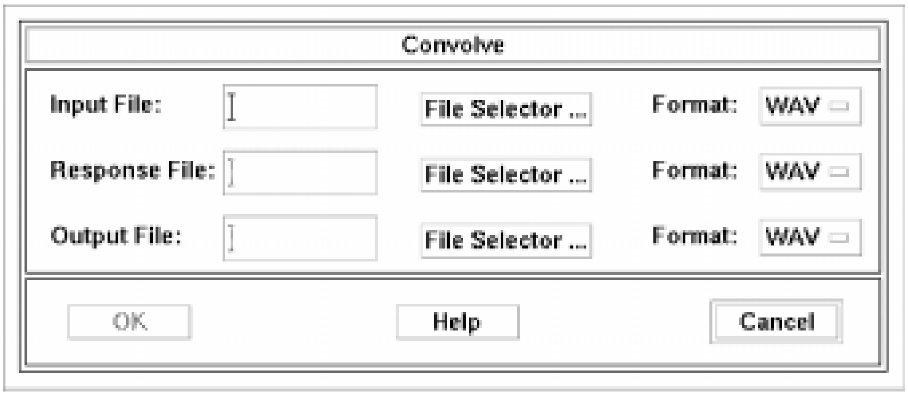


Figure 9.8–1 Sound file convolution dialog

Select **Convolve...** from the **Analysis** menu. Enter two existing files and their formats: the Input file and the Response file. Specify an Output file and its format to store the output signal. **OK** starts the processing, or **Cancel** to abort.

Any suitable sound editor/player can be used to replay the sound files.

Note! The Input sound file and the Response sound file must use the same sample frequency.

Hint: The normalization factor used when writing the sound file is displayed in the Echo Window. By default, normalization uses a variable value, so as to fill the maximum dynamic range of the 16–bit file, but a fixed normalization can be defined by the user with the Environment Variable NORMIRF (see Environment Variables, Section 11.8 on Page 11–19). Note that NORMIRF will not appear in the list of Environment Variables given by the **Tools, Environment Var.** dialog, unless it has been set previously. Therefore, to set it, use a command:

```
RAYNOISE> ENVIRONMENT NORMIRF value  RETURN
```

9.9 Frequency selections

9.9.1 General principles

A sub–dialog for frequency selection can be opened from all dialogs where frequencies must be selected. The dialog is described below. Alternatively, a single frequency or a selection can be entered directly into the frequency data field.

Frequency selections are arbitrary (not limited to octaves or fractions of octaves). In the absence of any frequency definition (*ie* with an empty frequency data entry field) RAYNOISE defaults to the standard octave–band center frequencies from 63Hz to 8kHz.

It must be emphasized that all other calculations depend on the basic results from a Mapping, therefore the frequency selections in any dependent calculation can only be the same as (or a sub–set of) what was used for Mapping. The exception is the Frequency Response calculation, where the echogram interpolation technique enables narrow–band results to be derived from Mapping data with relatively wide frequency steps.

Hint: The available frequencies in the database can be checked using **Inquire, Mapping**.

9.9.2 The frequency selection dialog

9.9.2.1 Frequency definitions

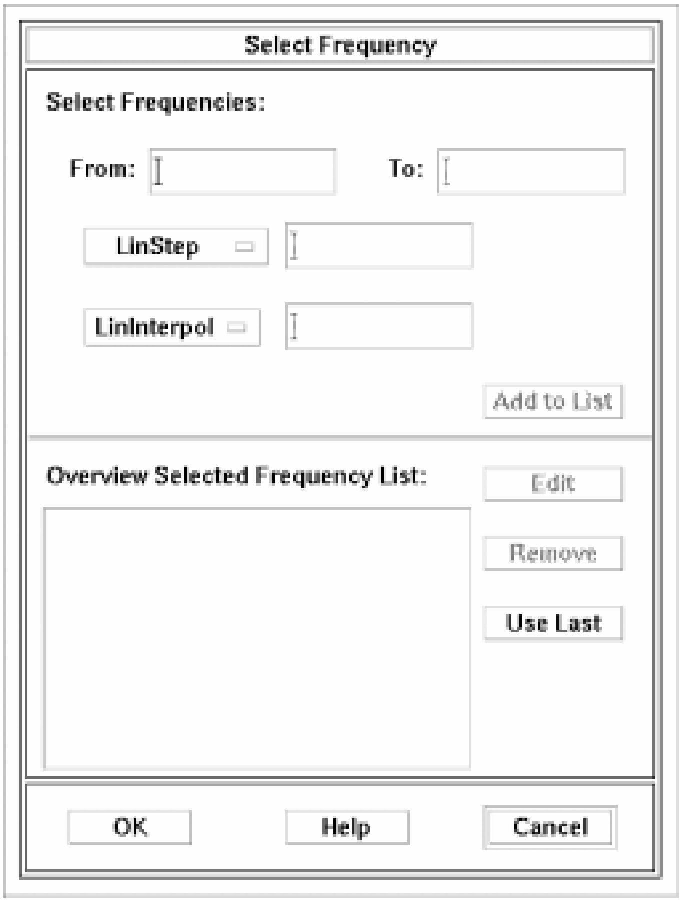


Figure 9.9–1 Frequency selection dialog

Typically, a frequency selection is based on a lower value, an upper value and a step. The step may be linear (in Hz: *freq2 = freq1 + Linstep...*) or logarithmic (*freq2 = freq1.Logstep, freq3 = freq1.Logstep*^{2}*...).* The last interval is adjusted downwards if necessary, to make the final frequency in a series equal to the defined upper value.

The **Lin(Log)Interpolate** entries are irrelevant to all procedures except Frequency Response function calculations, for which they define the fineness of the interpolation between the frequencies for which Mapping results are available. (If a value is entered in a frequency definition for another calculation, it just refines the selection of 'normal' frequencies which is used). The value to be entered is the number of interpolation intervals between each pair of adjacent 'master' frequencies (*eg*, a value of 10 will produce 9 'slave' frequencies).

It is possible to define multiple sequences of master and slave frequencies, with the Frequency Selector dialog: the master and slave frequencies are established from the selection, but if the sequences overlap, the frequencies are sorted in ascending order, such that interpolation at any slave frequency eventually uses the immediately–adjacent master frequencies from the complete selection.

9.9.2.2 Dialog procedure and cumulative frequency list

The dialog works interactively, until the frequency selection is completed, when **OK** will apply it to the main dialog, or **Cancel** returns without using the selection. After entering starting and ending frequencies, step and interpolation (if applicable), **Add to List** places the definition in the frequency list. Entries in the list can be Removed or Edited: in the case of **Edit** the values are placed back into the entry fields for alteration. The final sequence of frequencies is built in ascending order from the sequence(s) thus defined. Overlapping definitions are permitted. (See above, regarding the effect of overlaps on Interpolation).

The **Use Last** button recalls the complete frequency selection previously–used in this session of RAYNOISE, which may then be edited if desired.

It is thus possible to have a refined frequency step for a part of the range, for instance in the vicinity of a troublesome low–frequency mode, and wide steps for the remainder.

Example of use, in Postprocess command:

```
RAYNOISE> POSTPROCESS
POSTPROCESS> SOURCES 1 2
POSTPROCESS> FREQUENCY 250 TO 500 LINSTEP 25
POSTPROCESS> FREQUENCY 500 TO 1000 LINSTEP 50
POSTPROCESS> FREQUENCY 1000 TO 8000 LOGSTEP 2
POSTPROCESS> RETURN
```

9.10 Resource requirements

9.10.1 Memory usage

9.10.1.1 Memory usage during normal calculations

In normal calculations (*ie* Mapping) the main processing is a geometrical searching procedure. Therefore, the memory required is primarily related to the size of the model in terms of numbers of elements, with an approximately linear relationship. It is desirable to have all the data for the ray–tracing procedure in physical memory ('in core') to maximize the calculation speed. Therefore, the maximum practical memory should be allocated to RAYNOISE on starting the program (see *Allocating memory to RAYNOISE*, Section 9.10.2 on Page 9–29).

Some messages regarding memory use are issued during calculations. The amount of information increases with Debug level (see Section 11.2.2 on Page 11–4). The user can check on allocated space at any time using **Tools, Job Info, Space** (see Section 11.3.3 on Page 11–8).

9.10.1.2 Memory usage during Convolution

The Convolution calculation is the exception to the rule that maximum memory should be allocated to RAYNOISE, because, for the digital signal data being processed, the procedure uses part of the memory not allocated to RAYNOISE internal data. Therefore, it can be necessary to Exit RAYNOISE, Saving the model and results, and re–start using a much lower memory allocation, before opening the model and starting the Convolution procedure.

9.10.1.3 Memory usage during Animation

Animations using frames stored in memory (rather than in files: see *Results animation*, Section 10.2.6 on Page 10–25) can require much more memory than the ray–tracing calculations themselves. The requirement can be reduced by reducing the size of the RAYNOISE Main Window, before starting the animation. Otherwise, an uneven animation can result.

9.10.2 Allocating memory to RAYNOISE

Memory is used by RAYNOISE by dynamic allocation. The maximum memory which may be used by RAYNOISE is allocated when starting the program, using the parameter –m*X* on the command which launches RAYNOISE (or in the **raynoise.bat** file, in the case of a PC). *X* is the maximum memory in megawords. Please refer to the installation instructions for more information.

9.10.3 Disk usage

Disk usage is primarily determined by the amount of data storage required, which depends on the numbers of sources, receiver field points, frequencies, calculation parameters, Storelevel and (especially) numbers of echograms requested. See *Effect of calculation parameters on data storage*, Section 9.5.7 on 9.5.7, for more information.

It may be recommended to remove unwanted data and compress a database.

10

Results display

This chapter describes the methods for displaying results with RAYNOISE post–processing, including:

- ○ *Spatial plots: contour plots, deformed geometry and vectors*
- ○ *Animations*
- ○ *Frequency response curves*
- ○ *Echograms*
- ○ *3D ray–path diagrams*

10.1 General concepts in results display

10.1.1 Types of results display

Numerical simulations of acoustics can produce large amounts of results.

RAYNOISE includes a powerful and integrated graphical post–processor for easy and straightforward visualization, interpretation and analysis of these large amounts of acoustic data. The seamless integration of the post–processor with the RAYNOISE model viewer and the RAYNOISE analysis modules is designed to give the user access to a powerful tool for checking models and reviewing their results.

The graphical post–processing capabilities can be sub–divided into three categories:

- Spatial display (contour plots, deformed–geometry plots, vector plots)
- Frequency–response function curve plots
- Echogram plots

10.1.2 Spatial display: field–point post–processing

This type of post–processing produces displays of acoustic data and results on the complete RAYNOISE Field Point Mesh, in one of the following representations:

- **Color Contour/Isoline** representation of all acoustic data and results, including acoustic pressures, phases and acoustic quality measures.
- **Deformed Geometry** representation of acoustic results, for example a 3–d representation of sound pressure levels.
- **Vector** representation of acoustic results, for example arrows attached to the field points, with the length of the arrows proportional to the magnitude of the result.

It should be noted that there is no method of representing results on the acoustic mesh itself, since no results of any sort are calculated there. A nearby field–point mesh must be defined if results are wanted at (or just above) some surface. In fact, it is recommended not to put any field points too near to a surface, but to place them a finite distance away (*eg*, 0.3 metre) so that it is very clear which side of the surface they are on, both for the calculation processes and in displays.

10.1.3 Curve plotting

This type of post–processing produces two–dimensional XY–graphs of acoustic data and results, at selected points, in one of the following forms:

- **Frequency Response Functions** for plotting the variation of acoustic results as a function of frequency, for example an octave−band spectrum. (See Section 10.3 on Page 10−28).
- **Echogram Display** for plotting the arrival times, amplitudes, phases and directional data, for the echoes computed in the ray−tracing process. The echogram display window can be linked directly to a visualization of the corresponding ray path in the display of the model in the main window. (See Section 10.4 on Page 10−41).

10.1.4 Post−processor access

All graphical post−processor functions are accessed through the pull−down menu associated with the **Postprocess** key word in the Menubar.

Each entry in the **Postprocess** menu opens up a dialog to define the settings and to execute the display of a particular type of results display (see Section 10.1.1 on Page 10−3).

In general this requires a selection of:

- Result type
- Frequency (except for wide−band result types)
- Data format (if the result type is *Pressure*)
- Options
- The object(s) and viewport(s) to which the selections apply.

10.2 Spatial display on the field–point mesh

10.2.1 Color map contour/isoline plots

10.2.1.1 General dialog procedure

For the representation of acoustic data and results in the form of color contours or color isolines superimposed on the model, select the **Postprocessing, Color Map...** menu entry. In the associated *Color Map Display* dialog, select the settings for displaying data and results in the form of contours or isolines on the Field Point Mesh:

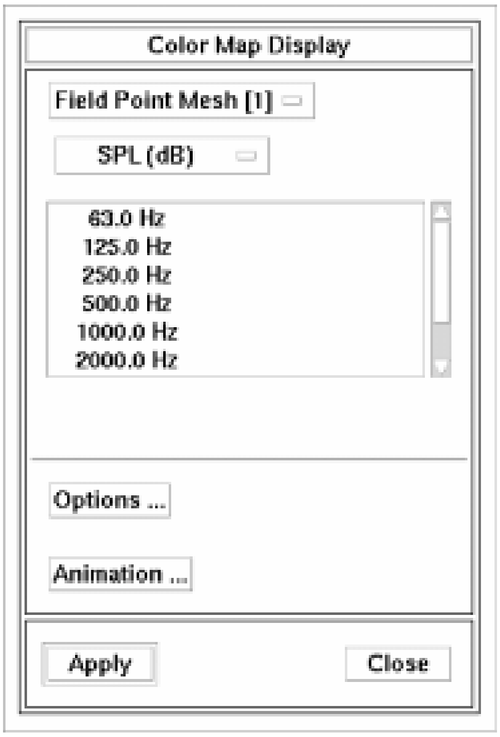


Figure 10.2–1 Color map display dialog

The upper option menu button allows the Object to be selected: this is only relevant if more than one Model is open, in which case a pull–down menu with a choice of *Field Point Mesh [n]* will be offered.

The second upper button of the *Color Map Display* dialog selects the *Data Type* of the acoustic quantity to be displayed (*eg* SPL with/out A–weighting, RT, STI, ...). If results at multiple frequencies are available, these are listed in a scrolling window, from which one *frequency* can be selected.

Note! Certain types of results may not be available for display, depending on the installed modules of RAYNOISE or the Storelevel used in the Mapping and Postprocess calculations. If only stationary sound calculations were selected (*ie* Storelevel 0) only the steady–state results, down to NC and NR in the list of data types, can be selected for display.

The settings in the dialog shown above (Figure 10.2−1) will produce a color map plot that shows the acoustic pressure in dB−format on the field point surfaces, at the chosen frequency, as illustrated in Figure 10.2−2.

The **Options...** button, in the lower part of the dialog, opens a new dialog for setting display parameters that affect any Color Map plot: see *Display options*, Chapter 12, beginning on Page 12−1, for a general explanation of graphical options, and *Color representation control*, Section 10.2.1.4 on Page 10−8, and following sections, for more details on color map options. A further button enables Animation of the results, if appropriate (see Section 10.2.6 on Page 10−25).

The two buttons at the bottom have the following functions:

- **Apply** displays the Color Map plot according to the present settings, without closing the dialog
- **Close** closes the dialog (without Applying).

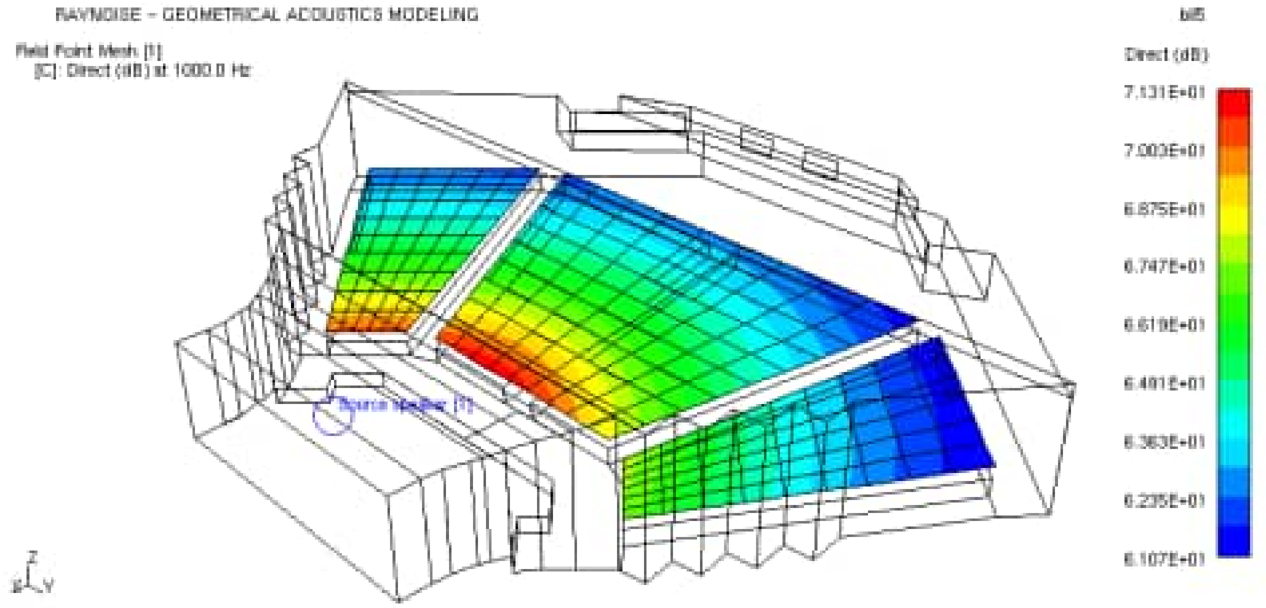


Figure 10.2–2 Example of a color map plot on field point surfaces (color map representation scheme)

10.2.1.2 Data type

The second upper button of the *Color Map Display* dialog selects the *Data Type* of the acoustic quantity to be displayed. The type of data and results that can be represented through a Color Map plot on the Field Point Mesh depends on the selected RAYNOISE calculation parameters. Certain results may not be available if they have not been calculated and stored in the database, depending on the options which were selected during the Analysis steps.

The available results are, in general:

- None
- Pressure results
- Energy ratio results (total:direct, ...)
- Noise criteria results (NC, NR)
- Acoustic quality results (Definition, Lateral efficiency, ...)
- Speech intelligibility results (STI, RaSTI, ...)
- Reverberation results (position−dependent RT60, EDT, EC, ...)

See *Acoustic quantities*, Section 13.1 on Page 13−3, for definitions of the result types.

Where the selected data type has been calculated at multiple frequencies, the frequencies of the results available in the database are listed in a scrolling window, from which one *frequency* can be selected. Where no frequency selection is required (*eg* for broadband results) this window does not appear.

10.2.1.3 Data format

The data formats available for the representation of the *Pressure* data type on the Field Point Mesh are:

- Amplitude
- Phase
- Real Part
- Imaginary Part
- dB
- Given Phase

Note that, in many cases, the Imaginary part is zero, so Amplitude and Real part are identical and it is meaningless to display Phase or results at a Given phase. This will always be the case if all sources are incoherent (*ie*, they have no defined phase relationship and results from multiple sources are added incoherently, as sound energy levels only).

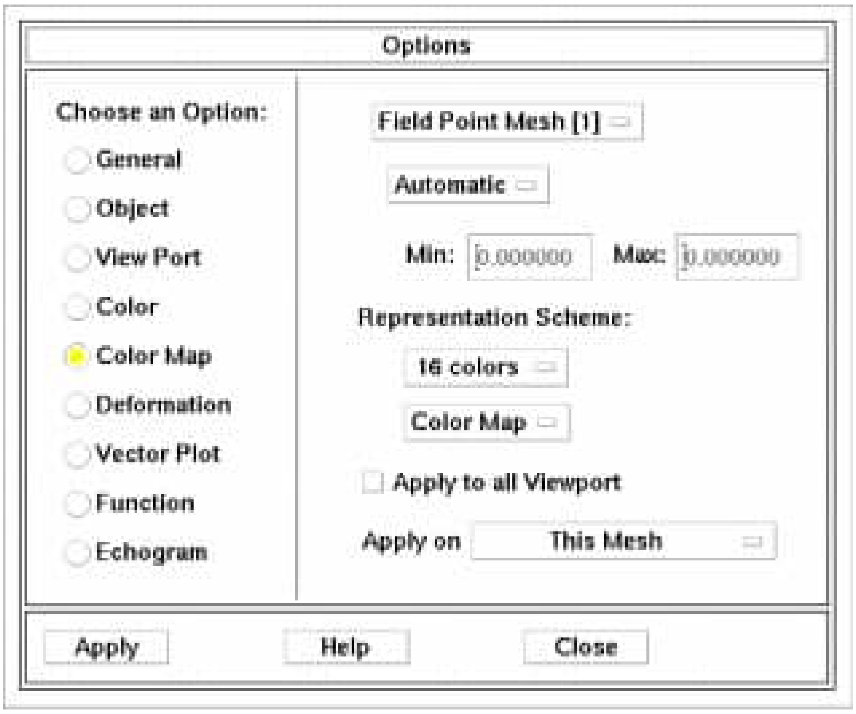
For all other Data Types, the Data Format is implicit in the Data Type (*eg* SPL in dB...) so this option menu does not appear.

10.2.1.4 Color representation control

RAYNOISE Graphical Postprocessing uses a fixed color scheme for color map representations. Low or small data values are represented by a deep−blue (‘cold’) color, high or large data values are represented by a bright−red (‘hot’) color. Between these extremes, the color varies with increasing values from deep−blue, over light−blue, cyan, green, yellow, orange to bright−red.

The relation between color and data values is visualized by means of a vertical color bar scale at the right−hand side of the viewport (see Figure 10.2−2). The scale for the data values is displayed at the side of the color bar.

Clicking **Options...** in the *Color Map* dialog opens a multi−dialog for setting parameters that control the color use of any Color Map plot (Figure 10.2−3). This dialog is also activated through the **Display, Graphical Options...** menu entry with the *Color Map* switch setting. The effect of the different buttons and scales in this dialog are described in the following sections.



Field Point Mesh [1]

Automatic

Min: 0.000000 Max: 0.000000

Representation Scheme:

16 colors

Color Map

□ Apply to all Viewport

Apply on This Mesh

Figure 10.2–3 Graphical Options dialog with Color Map switch

The buttons at the bottom of the Options dialog have the following functions:

- **Apply** passes the selected options, scaling *etc*, to the Color Map Display dialog, without closing the Options dialog

- **Close** closes the Options dialog (without Applying).

10.2.1.5 Scale values for field point mesh

This option menu defines the color bar scaling for the displayed data values on the Field Point Mesh, as follows:

Automatic causes the total range between the minimum and maximum value of the displayed data to be divided by the number of colors used by the color map, and defines equal intervals for each color.

Manual causes the range between the values specified by the user in the *Min* and *Max* fields to be divided by the number of colors used by the color map, and defines equal intervals for each color.

Hint: Manual scaling must be chosen, if an identical scale is required for several successive plots, in which the actual ranges of values vary.

10.2.1.6 Representation scheme

Two option menus control the number of colors and the type of visualization used by the color map plot:

The upper push–button allows a choice between 4, 8, 16, 64, 128 Colors for use by the color map display. The maximum number of colors which can be used is installation–dependent.

The lower push–button allows a choice between two representation schemes:

- *Color Map*, for a visualization through continuous bands of color
- *Iso–Line*, for a visualization through colored contour lines.

For example, Figure 10.2–2 above shows the color map representation scheme with 16 colors, while Figure 10.2–4 illustrates the iso–line representation scheme with 16 colors.

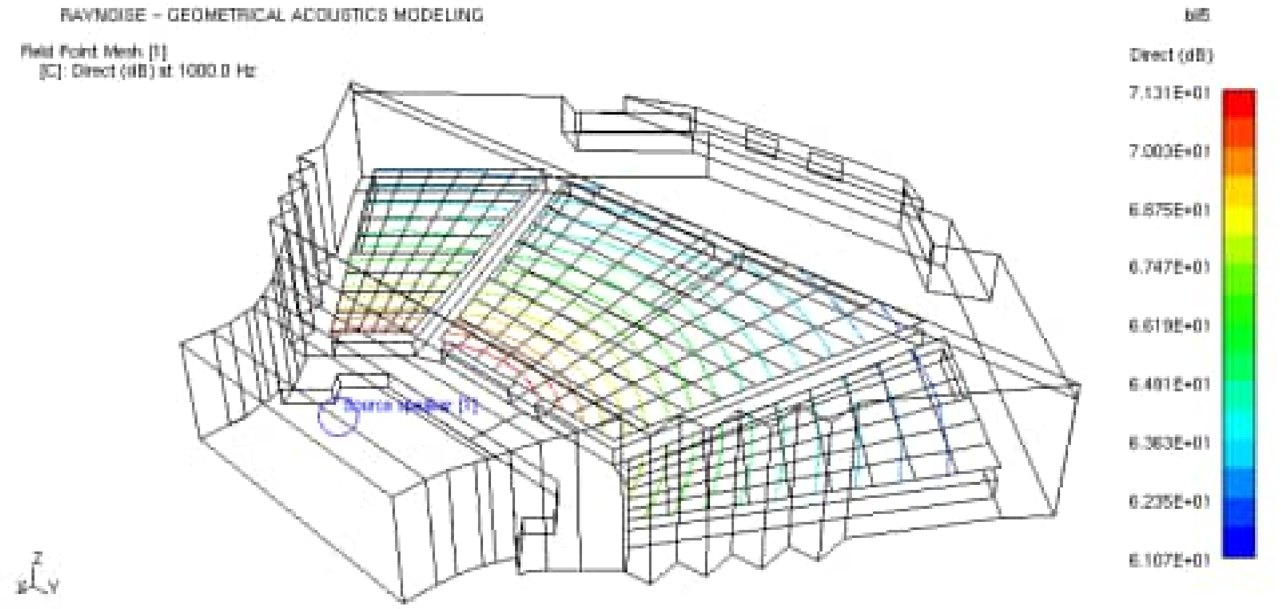


Figure 10.2–4 Example of a color map plot with iso–line representation

10.2.1.7 Apply to all Viewport

When toggled *On*, this causes the chosen options to be applied to all viewports in the Main Window, if multiple viewports are being used. When *Off* (default) the selected parameters only apply to the active viewport.

10.2.1.8 Apply on

This option menu allows selection of the object(s) to which the chosen options will be applied. This is only relevant if more than one field point mesh is available for display (*ie* more than one model is open).

The settings for each object can be controlled independently (*This Mesh* menu option) or propagated to all displayed objects (*All Meshes of this Type* or *All Meshes* have the same effect).

10.2.1.9 Clearing a color map display

In order to clear the display of a color map plot and return to the normal display of the model, select the data type *None* and apply the **Update** button.

Alternatively, the ‘Hide All’ button on the toolbar can be used, which clears all results and also any groups (sets, materials, ...) from the display.

10.2.2 Field point mesh deformation

10.2.2.1 General dialog procedure

The use of a deformed−geometry plot on the field point mesh may not be as self−evident as a color map plot, but can be of value, especially if a composite display is wanted (see Section 10.2.4 on Page 10−23).

The deformed−geometry representation of results is derived from the SYSNOISE program, in which some acoustic and vibrational results have spatial vector content. In RAYNOISE, there are no spatial vector results: all the acoustic values are scalar in the sense of having no direction. However, it can be interesting to show some results using a form of three−dimensional contour plot.

In this kind of post−processing display, the result at each point is represented by the displacement of the point from its original position. The displacement is a function of the data value at the point. Because this data value has no spatial (directional) attributes, the displacement vectors are all placed normal to the field point mesh at each point. This format gives a ‘surface’ indicating the distribution of the selected result, in which the field point mesh is displaced from its original position. An example of a deformed−geometry plot, showing the acoustic pressure in dB−format on the field point surfaces, can be seen in Figure 10.2−5, which is a sort of ’directivity balloon’ for a loudspeaker in near free−field conditions.

Note that it may be necessary to take care of the orientation of the elements of the field−point mesh, if they are created by more than one **Geometry, Field Points...** command, since the normals may be reversed in some parts of the field point mesh. The deformation scaling factor can also be useful. (See *Deformation scale control*, Section 10.2.2.4 on Page 10−14).

This type of representation only works for a field−point mesh containing surface elements. It should not be used with isolated field points. Field points joined by line elements may give odd effects.

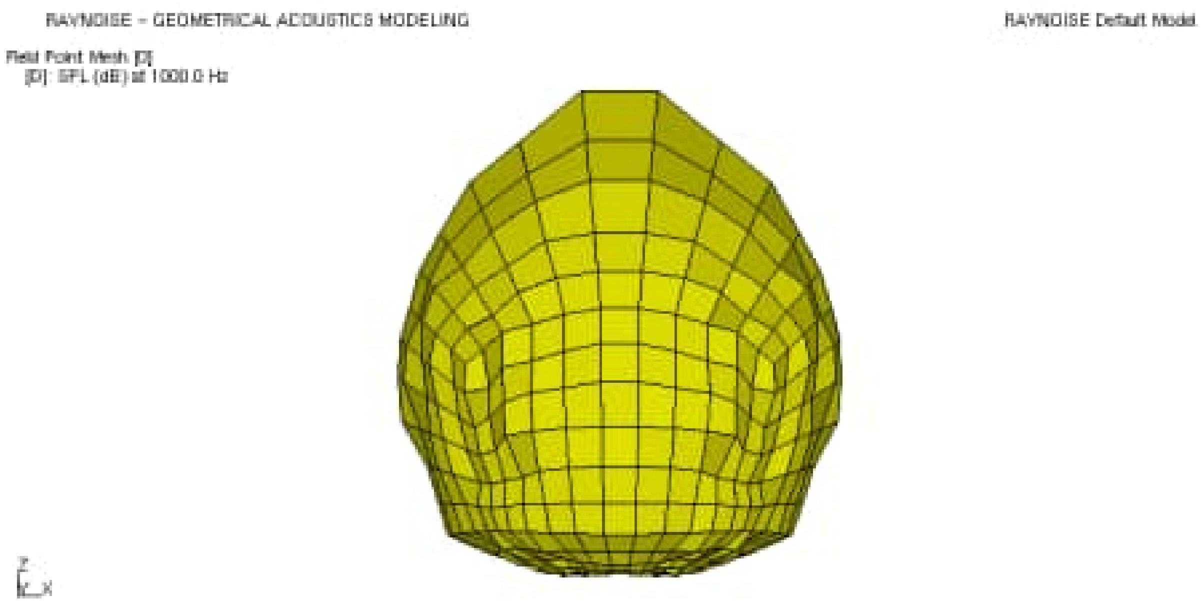

RAYNOISE – GEOMETRICAL ACOUSTICS MODELING
RAYNOISE Default Model
Field Point Mesh [D]
[D]: SPL (dB) at 1000.0 Hz
Z
X

Figure 10.2–5 Example of a deformed–geometry plot on field point surfaces

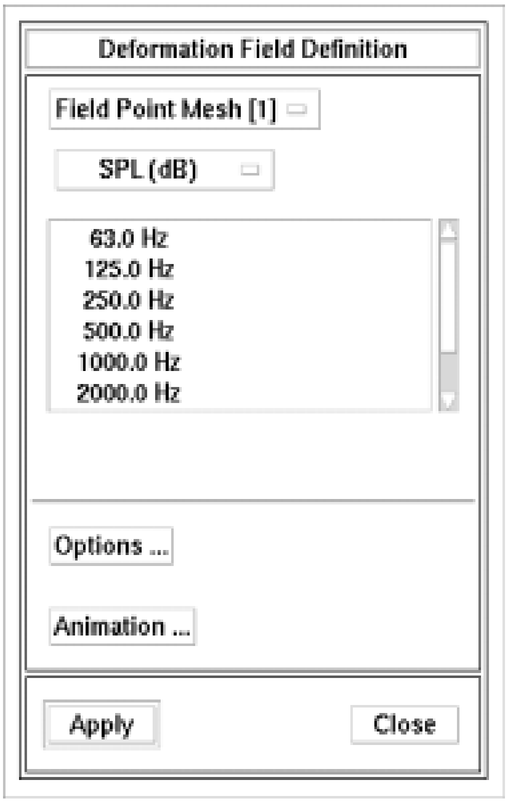


Figure 10.2–6 Deformation field definition dialog

Select the **Postprocessing, Deformation...** menu entry. In the associated *Deformation Field Definition* dialog, select the settings for displaying data and results in the form of a deformation of the Field Point Mesh:

The upper option menu button allows the Object to be selected: this is only relevant if more than one Model is open, in which case a choice of *Field Point Mesh [n]* will be offered.

The second upper button of the *Deformation field definition* dialog selects the *Data Type* of the acoustic quantity to be displayed (*eg* SPL with/out A–weighting, RT, STI,...). If results at multiple frequencies are available, these are listed in a scrolling window, from which one *frequency* can be selected.

Note! Certain types of results may not be available for display, depending on the installed modules of RAYNOISE or the Storelevel used in the Mapping and Postprocess calculations. If only stationary sound calculations were selected (*ie* Storelevel 0) only the steady–state results, down to NC and NR in the list of data types, can be selected for display.

The **Options...** button, in the lower part of the dialog, opens a new dialog for setting display parameters that affect any Deformation plot: see *Display options*, Chapter 12, beginning on Page 12–1, for a general explanation of graphical options, and *Deformation scale control*, Section 10.2.2.4 on Page 10–14, and following sections, for more details on deformed–geometry options. A further button enables Animation of the results, if appropriate (see Section 10.2.6 on Page 10–25).

The two buttons at the bottom have the following functions:

- **Apply** displays the Deformation plot according to the present settings, without closing the dialog

- **Close** closes the dialog (without Applying).

10.2.2.2 Data type

The type of result that can be assigned as a deformation variable for the Field Point Mesh depends on the selected RAYNOISE calculation parameters. Certain results may not be available if they have not been calculated and stored in the database, depending on the options which were selected during the Analysis steps.

The available results are, in general:

- None

- Pressure results

- Energy ratio results (total/direct, ...)

- Noise criteria results (NC, NR)

- Acoustic quality results (Definition, Lateral efficiency, ...)
- Speech intelligibility results (STI, RaSTI, ...)
- Reverberation results (RT60, EDT, EC, ...)

See *Acoustic quantities*, Section 13.1 on Page 13–3, for definitions of the result types.

Where the selected data type has been calculated at multiple frequencies, the frequencies of the results available in the database are listed in a scrolling window, from which one *frequency* can be selected. Where no frequency selection is required (*eg* for broadband results) this window does not appear.

10.2.2.3 Data format

The data formats available for the representation of the *Pressure* data type on the Field Point Mesh are:

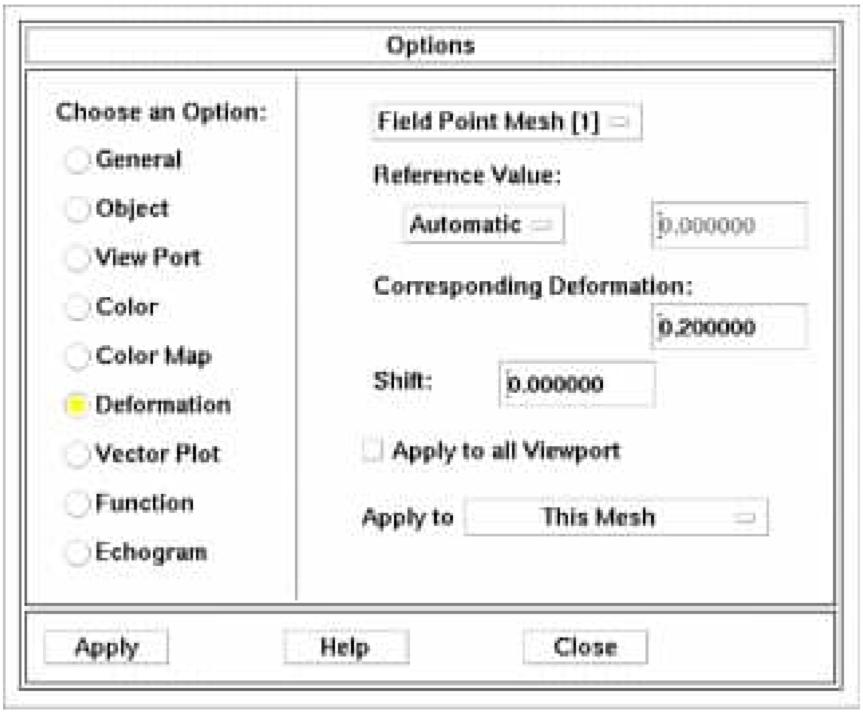
- *Amplitude*
- *Phase*
- *Real Part*
- *Imaginary Part*
- *dB*
- *Given Phase*

Note that, in many cases, the Imaginary part is zero, so Amplitude and Real part are identical and it is meaningless to display the Imaginary part, Phase or results at a Given phase. This will always be the case if all sources are incoherent (*ie*, they have no defined phase relationship and results from multiple sources are added incoherently, as sound energy levels only).

For all other Data Types, the Data Format is implicit in the Data Type (*eg* SPL in dB...) so this option menu does not appear.

10.2.2.4 Deformation scale control

Clicking the **Options...** button in the *Deformation* dialog opens a multi–dialog for setting parameters that control the scale of deformation applied to the Field Point Mesh in the graphics window (Figure 10.2–7). This dialog is also activated through the **Display, Graphical Options...** menu entry with the *Deformation* switch setting.



Field Point Mesh [1]

Reference Value: Automatic 0.000000

Corresponding Deformation: 0.200000

Shift: 0.000000

□ Apply to all Viewport

Apply to This Mesh

Figure 10.2–7 Graphical Options dialog with Deformation switch

The buttons at the bottom of the Options dialog have the following functions:

- **Apply** passes the selected options, scaling *etc*, to the Deformation Field Definition dialog, without closing the Options dialog
- **Close** closes the Options dialog (without Applying).

10.2.2.5 Scale values for field point mesh deformation

The relative scale of deformation in the graphics window is governed by three different parameters:

- **Reference Value**

This option menu sets the data value that will be used as reference value for the definition of the scale of deformation:

Automatic will assign the maximum value of the selected deformation data as reference value.

Manual allows the user to define an arbitrary reference value in the entry field next to the button.

- **Corresponding Deformation**

This entry field allows the user to specify a scaling value relative to the maximum model (acoustic mesh) size. (Model size is displayed upon Input of the acoustic mesh). This value determines the relative magnitude of deformation to be applied to the reference data value defined above.

- **Shift**

This entry field allows a shift of the data value that corresponds to a zero deformation.

When no shift is applied (zero entry in this field) a zero data value at a point will correspond to zero deformation of that point, and the magnitude of the deformation will be distributed linearly between the zero data value and the maximum data value, *ie* over the entire data range.

When a non−zero shift is applied, this shift value will correspond to zero deformation of the point, and the magnitude of deformation will be distributed linearly between the shift data value and the maximum data value, *ie* over a restricted data range. A data value lower than the shift data value will produce a deformation of the point in the opposite (negative) direction.

The application of shift is especially useful in situations where the data or results only show little variation (*eg* in dB−format).

In other words, the Reference Value and Corresponding Deformation parameters fix the relative magnitude of deformation, while the Shift parameter fixes the zero deformation reference value. Together, these parameters fix a common scale of deformation for all field point results.

The deformed geometry plot in Figure 10.2−5, for instance, has a scale of deformation where the maximum displacement of a point corresponds to a relative deformation of 10% (0.10) of the maximal model size.

Note that the direction of motion of the field−point mesh in a deformed−geometry plot depends on the orientation of the positive normal to the field−point elements. Take care when defining the field−point mesh to get consistently−oriented elements. (See *Reversing element orientations*, Section 6.2.6 on Page 6−7).

10.2.2.6 Apply to all Viewport

When toggled *On*, this causes the chosen options to be applied to all viewports in the Main Window, if multiple viewports are being used. When *Off* (default) the selected parameters only apply to the active viewport.

10.2.2.7 Apply to

This option menu allows selection of the object(s) to which the settings underneath will be applied. This is only relevant if more than one field point mesh is available for display (*ie* more than one model is open).

The settings for each object can be controlled independently (*This Mesh* menu option) or propagated to all displayed objects (*All Meshes of this Type* or *All Meshes* have the same effect).

10.2.2.8 Clearing a deformation display

To clear the display of a deformed−geometry representation and to return to the normal display of the model, select the data type *None* and apply the **Update** button. Alternatively, the 'Hide All' button on the toolbar can be used, which clears all results and also any groups (sets, materials, ...) from the display.

10.2.3 Field point vector displays

10.2.3.1 General dialog procedure

The use of a vector plot on the field point mesh may not be as self−evident as a color map plot, but can be of value, especially if a composite display is wanted (see Section 10.2.4 on Page 10−23).

The vector representation of results is derived from the SYSNOISE program, in which some acoustic and vibrational results have spatial vector content. In RAYNOISE, there are no spatial vector results: all the acoustic values are scalar in the sense of having no direction. However, similarly to the deformed field−point mesh plots, it can be interesting to show some results using a vector (arrow) plot.

In this kind of post−processing display, the result at each point is represented by an arrow displayed at the point. The magnitude and color of the arrow are a function of the data value at the point. Because this data value has no spatial (directional) attributes, the vectors are all placed normal to the field point mesh at each point. Typically, this results in a series of parallel arrows (on a plane field point mesh) or radial arrows (on a sphere). As with a deformed−shape plot of the field point mesh, this format gives a 'surface' formed by the arrow−heads, indicating the distribution of the result selected to be shown, but in this case the underlying field point mesh remains in its original location. An arrow plot, showing the acoustic pressure in dB−format on the field point surfaces, as illustrated in Figure 10.2−8, related to the directivity balloon of the loudspeaker as in the deformed−shape plot (Figure 10.2−5).

Note that it may be necessary to take care of the orientation of the elements of the field−point mesh, if they are created by more than one **Geometry, Field Points...** command, since the normals may be reversed in some parts of the field point mesh. The arrow size control can also be useful. (See *Arrow control*, Section 10.2.3.4 on Page 10−21).

This type of representation only works for a field–point mesh containing surface elements. It should not be used with isolated field points. Field points joined by line elements may give odd effects.

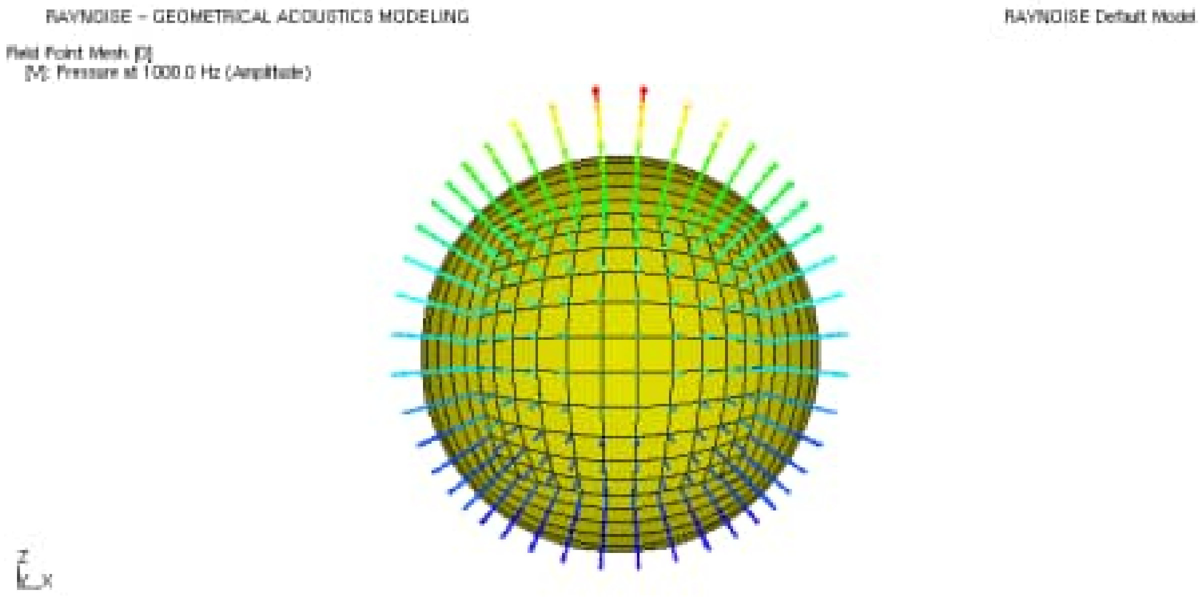

RAYNOISE – GEOMETRICAL ACOUSTICS MODELING
RAYNOISE Default Model
Field Point Mesh [D]
[V]: Pressure at 1000.0 Hz (Amplitude)
Z
X

Figure 10.2–8 Example of an arrow plot on field point surfaces

Select the **Postprocessing, Vector Field...** menu entry. In the associated *Vector Field Definition* dialog, select the settings for displaying data and results in the form of vectors on the Field Point Mesh:

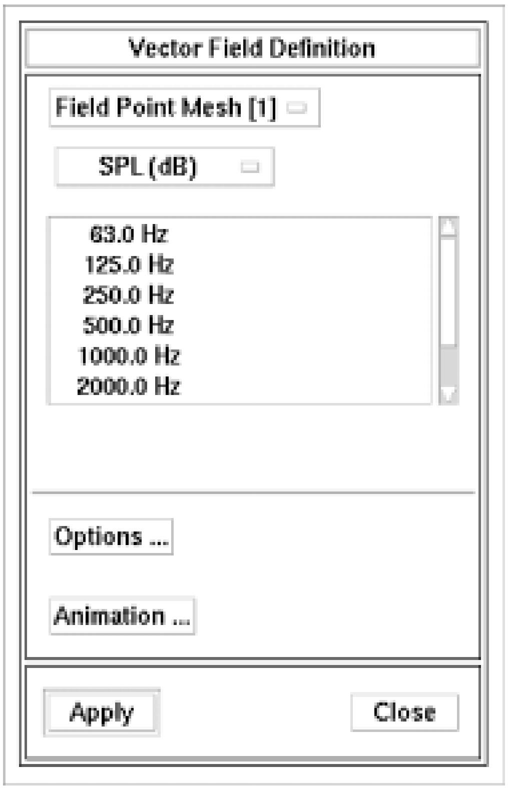


Figure 10.2–9 Vector field definition dialog

The upper option menu button allows the Object to be selected: this is only relevant if more than one Model is open, in which case a choice of *Field Point Mesh [n]* will be offered.

The second upper button of the *Vector field definition* dialog selects the *Data Type* of the acoustic quantity to be displayed (*eg* SPL with/out A–weighting, RT, STI, ...). If results at multiple frequencies are available, these are listed in a scrolling window, from which one *frequency* can be selected.

Note! Certain types of results may not be available for display, depending on the installed modules of RAYNOISE or the Storelevel used in the Mapping and Postprocess calculations. If only stationary sound calculations were selected (*ie* Storelevel 0) only the steady–state results, down to NC and NR in the list of data types, can be selected for display.

The two buttons at the bottom have the following functions:

- **Apply** displays the Vector plot according to the present settings, without closing the dialog
- **Close** closes the dialog (without Applying).

10.2.3.2 Data type

The type of result that can be assigned as a vector display variable for the Field Point Mesh depends on the selected RAYNOISE calculation parameters. Certain results may not be available if they have not been calculated and stored in the database, depending on the options which were selected during the Analysis steps.

The available results are, in general:

- None
- Pressure results
- Energy ratio results (total/direct, ...)
- Noise criteria results (NC, NR)
- Acoustic quality results (Definition, Lateral efficiency, ...)
- Speech intelligibility results (STI, RaSTI, ...)
- Reverberation results (RT60, EDT, EC, ...)

See *Acoustic quantities*, Section 13.1 on Page 13–3, for definitions of the result types.

Where the selected data type has been calculated at multiple frequencies, the frequencies of the results available in the database are listed in a scrolling window, from which one *frequency* can be selected. Where no frequency selection is required (*eg* for broadband results) this window does not appear.

10.2.3.3 Data format

The data formats available for the representation of the *Pressure* data type on the Field Point Mesh are:

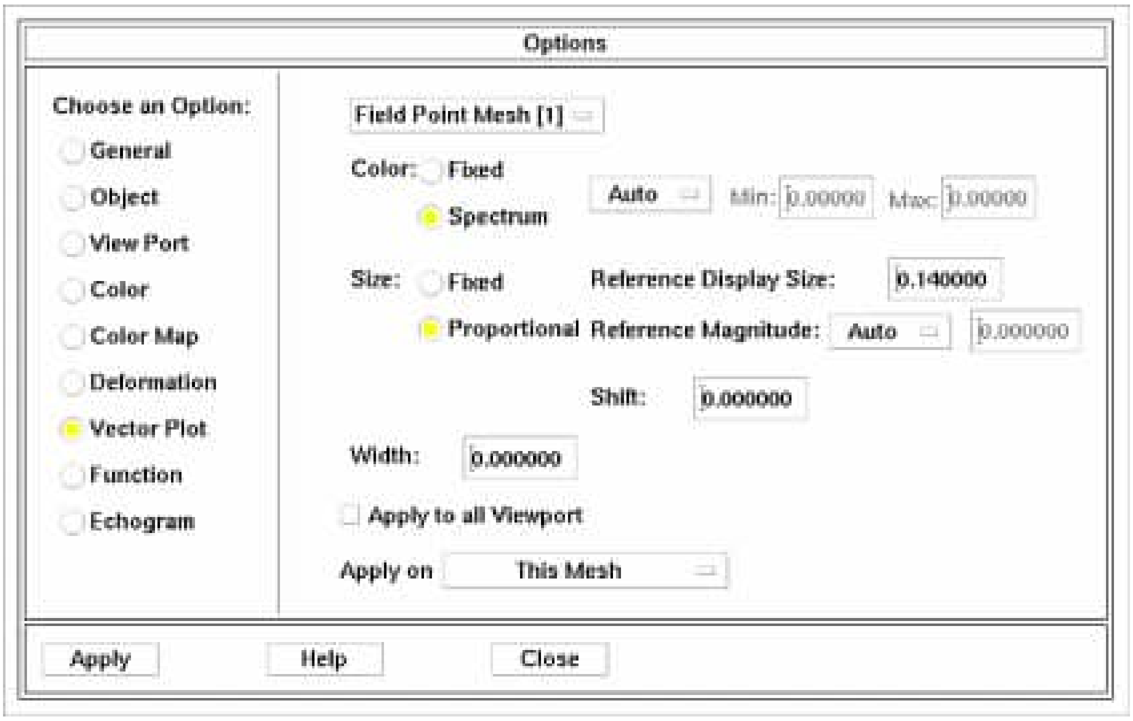
- *Amplitude*
- *Phase*
- *Real Part*
- *Imaginary Part*
- *dB*
- *Given Phase*

Note that, in many cases, the Imaginary part is zero, so Amplitude and Real part are identical and it is meaningless to display the Imaginary part, Phase or results at a Given phase. This will always be the case if all sources are incoherent (*ie*, they have no defined phase relationship and results from multiple sources are added incoherently, as sound energy levels only).

For all other Data Types, the Data Format is implicit in the Data Type (*eg* SPL in dB...) so this option menu does not appear.

10.2.3.4 Arrow controls

The **Options...** button in the *Vector Field Definition* dialog opens up a multi–dialog for setting parameters that control the display of colored arrows:



Options

Choose an Option:
○ General
○ Object
○ View Port
○ Color
○ Color Map
○ Deformation
● Vector Plot
○ Function
○ Echogram

Field Point Mesh [1]

Color: ○ Fixed ● Spectrum Auto Min: 0.00000 Max: 0.00000

Size: ○ Fixed ● Proportional Reference Display Size: 0.140000

Reference Magnitude: Auto 0.000000

Shift: 0.000000

Width: 0.000000

□ Apply to all Viewport

Apply on This Mesh

Figure 10.2–10 Graphical Options dialog with Vector switch

This dialog is also activated through the **Display, Graphical Options...** menu entry with the *Vector Plot* switch setting.

The buttons at the bottom of the Options dialog have the following functions:

- **Apply** passes the selected options, scaling *etc*, to the Vector Field Definition dialog, without closing the Options dialog
- **Close** closes the Options dialog (without Applying).

10.2.3.5 Color values for vectors

The settings in the **Color** area determine the color that will be applied to the arrows. The color use in Vector Field plots is largely analogous to color use in Color Map plots (see Section 10.2.1.4 on Page 10–8):

Fixed applies a fixed blue color to all the arrows

Spectrum applies a color depending on the magnitude of the data value represented by the arrow.

Auto provides an automatic color scale between the maximum and minimum values of the displayed quantity.

Manual allows the user to specify *Minimum* and *Maximum* values for the color scale in the corresponding entry fields.

10.2.3.6 Scale values for vectors

The sizes of the plotted vectors are determined by three parameters:

- **Size**

Fixed applies a fixed relative size to the arrows. This size is determined by the value in the Reference Display Size entry field, which defines the arrow size as a multiple of the maximum size of the acoustic mesh. (Maximum size is indicated on Input of the mesh).

Proportional applies a relative size which depends on the magnitude of the data represented by the arrow. This size is determined by the value in the Reference Display Size entry field. When the Proportional Reference Magnitude button is set to *Auto*, an arrow scaled by the Reference Display Size will be applied to the maximum value of the displayed data, and the arrow sizes for the other data values will be scaled accordingly. When the button is set to *Manual*, the Reference Display Size will be applied to the data value specified in the Proportional Reference Magnitude entry field, and the arrow size for the other data values will be scaled accordingly.

- **Shift**

The *Shift* changes the 'dynamic range' of the vector plot by shifting the data value that corresponds to a 'zero' vector. The effect is analogous to the effect of *Shift* in Deformation Scale Control (see Section 10.2.2.4 on Page 10–14).

Note that the direction of the arrows in a vector plot depends on the orientation of the positive normal to the field–point elements. Take care when defining the field–point mesh to get consistently–oriented elements. (See *Reversing element orientations*, Section 6.2.6 on Page 6–7).

- **Width**

This entry field specifies the thickness of the arrow lines as a real value. The default zero value corresponds to the smallest line thickness which can be displayed on the screen, which depends on the display resolution.

10.2.3.7 Apply to all Viewport

When toggled *On*, this causes the chosen options to be applied to all viewports in the Main Window, if multiple viewports are being used. When *Off* (default) the selected parameters only apply to the active viewport.

10.2.3.8 Apply on

This option menu allows selection of the object(s) to which the settings underneath will be applied. This is only relevant if more than one field point mesh is available for display (*ie* more than one model is open).

The settings for each object can be controlled independently (*This Mesh* menu option) or propagated to all displayed objects (*All Meshes of this Type* or *All Meshes* have the same effect).

10.2.3.9 Clearing a vector field display

In order to clear an arrow plot display and to return to the normal display of the model, select the data type *None* and apply the **Update** button. Alternatively, the 'Hide All' button on the toolbar can be used, which clears all results and also any groups (sets, materials, ...) from the display.

10.2.4 Composite post–processing displays

RAYNOISE Graphical Post−processing contains a special feature: Composite Post−processing displays, obtained by combining different types of post−processing displays into one display. For example, the Color Map display of the STI distribution on a field−point mesh can be displayed on top of the Deformed Mesh Geometry of the acoustic pressures (SPL) across the field.

If a Postprocessing dialog box is already open, it is switched to the selected Post−processing plot type (color map, deformation, vectors) as soon as the menu item (**Postprocessing, Color Map...** *etc*) is picked.

The combination of different post−processing displays into a single, composite, post−processing display is achieved by executing multiple post−processing displays in succession. Each time the **Update** button of the post−processing dialogs is clicked, the present settings of this dialog are applied to the Field Point Mesh, without destroying the previous settings for the other display types.

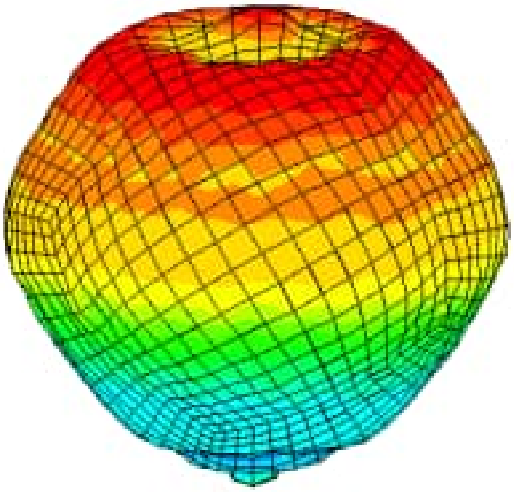


Figure 10.2–11 Composite display: color contours and deformed shape

10.2.5 Multiple model displays

Using the multi–mode database management system, it is possible to Open several databases (*ie* several models) at the same time. (See *Open an existing database file*, Section 5.2.2 on Page 5–5, for more information on databases). The visibility of these models and their related results can be turned on and off independently in the Main Window, with the *View menu* (see Section 8.1 on Page 8–3). This can be done independently, in each of the viewports of the Main Window, if multiple viewports have been opened. See *Changing the number of views*, Section 12.1.12 on Page 12–16, for information on multiple viewports). This enables the viewing of results on multiple models in the same RAYNOISE session – for instance, to compare the results of two possible designs, two variants of the same design with different materials, different source characteristics, *etc*. The following points must be noted when working with multiple models and results:

- Results are only visible if the corresponding model object (*ie* the field point mesh of that model) is made visible. If some display of results is commanded on an invisible object, the display action will be carried out but there will be no apparent change on the screen: only when the object is made visible will the change be seen.

- The visibility of objects is independent of whether the model of which they are a part is the Active model (*ie* **File, Activate...** is irrelevant in the context of results display). However, the title in the upper right of the screen indicates the active model.

- Color maps, deformation and vector plots on different field−point meshes in the same viewport will use the same colors but different scales, unless they are forced to use the same scale by selecting *Manual* ranges in the corresponding Graphical Options dialogs (see for example *Scale values for field point mesh*, Section 10.2.1.5 on Page 10−9, for how to do this). Similar procedures apply for deformation and vector displays. Similarly, scales for all three types of results display can be controlled independently or applied to all viewports.

10.2.6 Results animation

10.2.6.1 General principles

Through the **Animation...** function, Graphical Post−processing gives the user access to a powerful tool for results analysis: real−time animation of calculation results, by the instantaneous and successive display of results for a series of steps (either for successive frequencies or for successive phase angles).

This capability enables, for instance:

- tracking the variation of the acoustic pressure with frequency in a color map representation over the entire acoustic domain.
- tracking the variation of instantaneous acoustic pressure with time, from the results of phase ray−tracing.

In principle, any type of mesh post−processing display can be used for an animated display. In practice, only certain combinations have real physical meaning.

The general principles of RAYNOISE animation are as follows:

Selection of a Mesh Postprocessing type of display in the **Postprocess** menu: **Color Map** or **Deformation** or **Vector Field**. This also involves the selection of the data type and the data format of the acoustic quantity to be represented, and options if desired.

Selection of the **Animation...** function in the Postprocessing dialog, to open a dialog which allows the user to select the series of instantaneous results to be displayed in succession. (See Figure 10.2−12). By selecting the *Phase* step with the Phase/Frequency toggle, results at the frequency which has already been selected will be displayed as the phase angle cycles from 0 to 360 degrees, with the increment specified in the input field. A series of frames for each phase angle is built up and then cycled as a video sequence.

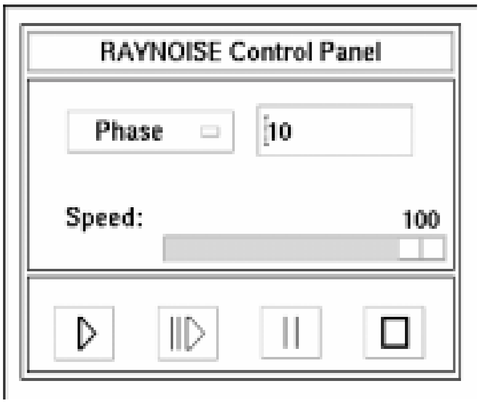


Figure 10.2–12 Animation dialog

By selecting the *Frequency* option, results will be displayed at a series of frequencies, starting with the lowest frequency which has been calculated and ranging up to the highest, then returning to the lowest and repeating the cycle.

10.2.6.2 Control panel buttons

The animation is controlled by several buttons, which resemble those on a tape–recorder. From left to right:

- *Play*: starts the animation; the frame sequence is first built, then cycled.
- *Step*: the animation proceeds one step at a time when prompted.
- *Pause*: the animation is frozen at the current frame
- *Stop*: the animation is halted and returns to the post–processing dialog.

The *speed control slider bar* can be used to adjust the replay speed, from 0 (no animation) to 100 (maximum speed) before or during the animation play.

10.2.6.3 Storage of animation frames

The animation process first builds the sequence of frames, stores them and then replays them, to speed up the animation play. There are two methods for storing the animation frames:

- in memory (default)
- as Compuserve GIF format files

The storage method is controlled by the Environment Variable OUTPUTFRAME. (See Environment variables, Section 11.8 on Page 11–19 for more information).

Note! It may be necessary to allocate substantial memory to RAYNOISE to support animation, possibly much more than would otherwise be required to support any of the calculation steps. This is especially true if the animation frames are all stored in memory for replay. Memory allocation is controlled by the –mX parameter on the command line which starts RAYNOISE (see *Space and memory requirements*, Section 5.4 on Page 5–19).

10.3 Frequency response function curves

10.3.1 Introduction

RAYNOISE Graphical Postprocessing supports different types of curve plotting for the visualization of results through two−dimensional XY−graphs. The following categories of XY−graphs for representing the variation of various acoustic quantities as functions of frequency or time are currently available through different entries in the **Postprocessing** menu:

- The **Point Function...** menu entry opens a dialog to produce graphs representing the variation of acoustic results at field points as a function of frequency.
- The **File Function...** menu entry opens a dialog to produce graphs representing the variation of data or results stored in files, as a function of frequency.
- The **Table Function...** menu entry opens a dialog to produce graphs representing the values in tables.

The menu entries listed above open different variants of the *Response Function* dialog. Each of the dialog variants contains switches, option menus and entry fields that are common to all types of curve plotting. These complements and options are detailed in Section 10.3.2 on Page 10−28, *Common curve concepts*.

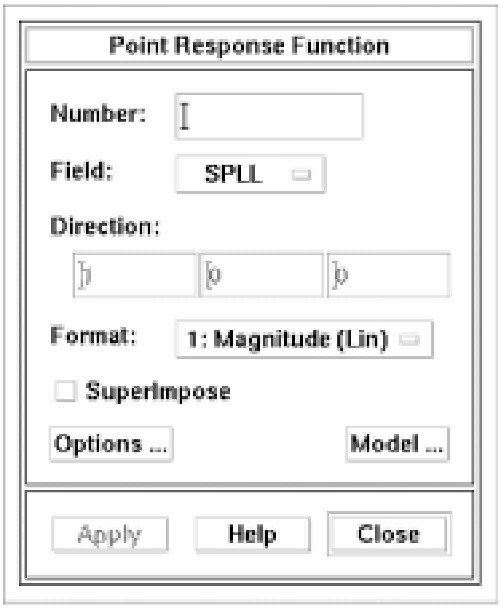
The curve plotting settings that are specific to a curve plot type are detailed in separate sections for each curve plot type. These sections also contain figures illustrating the appearance of the different variants of the *Response Function* dialog.

A second window is opened when the response function dialog is opened, in which the curves themselves will be drawn. This window can be re−positioned and re−sized by the user at any time. Its initial position and size are determined by the Environment Variable CRVWINPOS in the RAYNOISE.STP file.

10.3.2 Common curve concepts

10.3.2.1 Overview of dialogs

This section describes the common elements of the different *Response Function* dialog variants and explains the general concepts used in curve plotting.


Point Response Function

Number:

Field: SPLL

Direction: 1 0 0

Format: 1: Magnitude (Lin)

☐ Superimpose

Figure 10.3–1 General lay–out of the response function dialog

10.3.2.2 Common complements

The general lay−out of the *Response Function* dialog is shown in Figure 10.3−1. All *Response Function* dialogs always contain seven buttons:

- **New Curve** clears the display and plots a new curve without closing the dialog.
- **Format** selects the type of scale on the y−axis:

  1 – *Magnitude (Lin)*: Magnitude with a linear scale

  2 – *Magnitude (Log)*: Magnitude with a logarithmic scale

  3 – *Magnitude (dB)*: Magnitude with a dB−scale. Note that the addition of A−, B−, C− or D−weighting can be selected in the related Options dialog. However, beware of double−weighting the values if the selected data type already includes A−weighting (*ie*, SPLA)

  4 – *Magn/Phase*: Amplitude value (in dB lin) and phase value (in degrees) on two parallel plots

  5 – *Real/Imag*: Real part and imaginary part, on two parallel plots

  6 – *Real*: Real part only

  7 – *Imaginary*: Imaginary part only

- **Superimpose** when the chosen curve parameters are applied, the new curve will be superimposed on any existing curve(s). If this button is *Off* (greyed) a new curve will cause previous curve(s) to be deleted from the curve−plotting window.

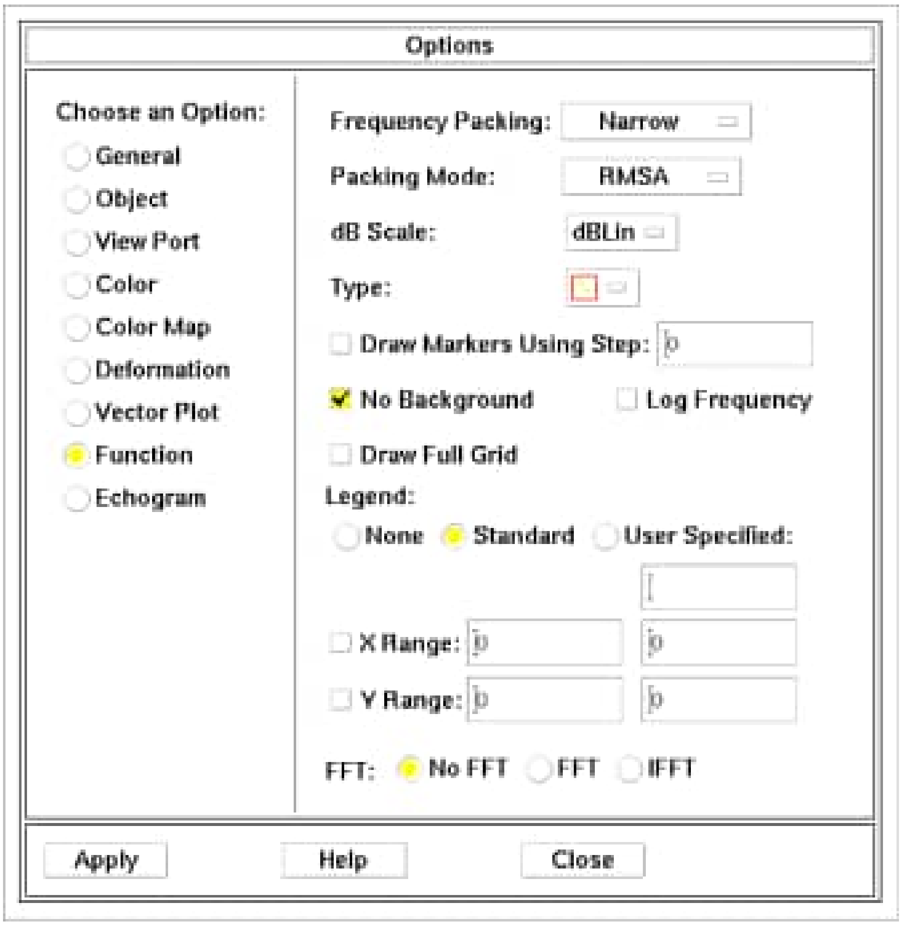
- **Options** opens the sub–dialog in which curve options can be selected.
- **Model** enables curves to be plotted for different models (all of which must be open, but need not explicitly be made Active). A sub–dialog for model selection is opened, from which the required model can be selected and the Apply button used in the sub–dialog. The model thus selected becomes the currently–active model.
- **Apply** actions the chosen curve parameters and draws a curve.
- **Help** opens the related on–line help.
- **Close** closes the curve window and returns control to the main window. The curve window also contains an 'exit' button ➡ which has the same effect.

Note! Do not use the menus and other dialogs of the main window, other than graphical actions such as rotating the model view or picking points, whilst carrying–out actions on the curve window, since the RCL is in 'Curve' mode and commands generated by the menus of the main window will lead to command syntax error messages and possible odd effects.

The Direction fields which may appear are greyed–out and are reserved for future use with data which are spatial vectors.

10.3.3 Options for response functions

The **Options...** button in the *Response Function* dialogs opens a new dialog for setting general XY–graph display parameters (Figure 10.3–2). This dialog is also activated through the **Display, Graphical Options...** menu entry with the *Function* switch setting.



Choose an Option:
○ General
○ Object
○ View Port
○ Color
○ Color Map
○ Deformation
○ Vector Plot
● Function
○ Echogram

Frequency Packing: Narrow
Packing Mode: RMSA
dB Scale: dBLin
Type:
☐ Draw Markers Using Step: 0
☑ No Background ☐ Log Frequency
☐ Draw Full Grid
Legend:
○ None ● Standard ○ User Specified:
☐ X Range: 0 0
☐ Y Range: 0 0
FFT: ● No FFT ○ FFT ○ IFFT

Figure 10.3–2 Graphical Options dialog for response functions

The amended options are first entered with the **Apply** button, then **Close** is used to remove the dialog, or it may be left open if repeated actions are likely.

10.3.3.1 Frequency Packing

This option menu determines the type of frequency packing used to display Frequency Response Functions:

- *Narrow*: displays the FRF in narrow band format (no packing). This is the default and will normally give typical 1/1 or 1/3–octave spectra, if the analysis results contain the corresponding frequencies.
- *Third Octave*: displays the FRF in third–octave band format.
- *Octave*: displays the FRF in octave–band format.

Third Octave or *Octave* will normally only be used if narrow–band spectra have been calculated (see *Frequency response calculation*, Section 9.6 on Page 9–20) and the user wants to convert (integrate) these into banded spectra. Note that the frequency–band integrations are only done during the curve plotting: the results in the database are unaffected

Frequency packing concepts, Section 10.3.4 on Page 10–34 contains more information about the concepts of frequency–packing type.

10.3.3.2 Packing Mode

This option menu determines the frequency packing mode used in the display of Frequency Response Functions:

- RMSA
- Added
- Average
- Continuous

Frequency packing concepts, Section 10.3.4 on Page 10–34 contains more information about the concepts of frequency–packing mode.

10.3.3.3 dB Scale

This push–button enables the application of dB filters to the (linear) dB display format of FRF curves:

- dBLin to apply no filtering (linear dB scale)
- dBA to apply A–filtering
- dBB to apply B–filtering
- dBC to apply C–filtering
- dBD to apply D–filtering

dB filters, Section 10.3.5 on Page 10–38, contains more information about dB correction filters.

> Beware of the risk of double–filtering the results if the selected result type already includes a filter (*ie* SPLA). To display a result with C–weighting, for instance, select the SPL–linear result type and then apply the dBC scale to the plot.

10.3.3.4 Type

This option menu allows selection of the representation type for the curve:

- Type 1 – Continuous (lines connecting each of the data points)
- Type 2 – Discrete (histogram with narrow vertical bars)
- Type 3 – Steps (step function with jumps mid–way between the data points)
- Type 4 – Discrete (histogram with broad vertical bars)

Type 4 is recommended for 1/1 and 1/3–octave band spectra.

Different curve representation types are shown superimposed in Figure 10.3–3.

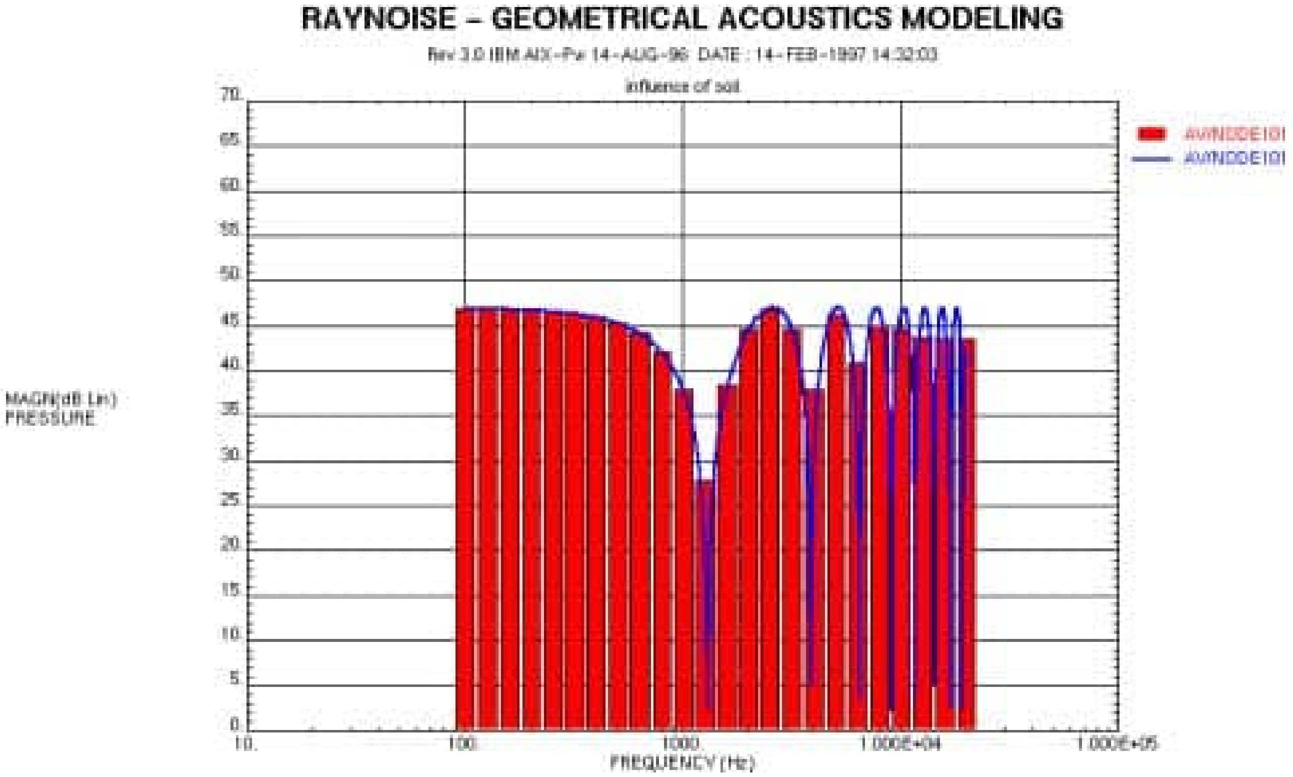


Figure 10.3–3 Curve types for response functions

10.3.3.5 Draw Markers Using Step

When toggled *On*, marker symbols are placed at certain function points on the plotted curve. The interval between successive markers is defined by the value in the **Step** entry field next to the label: a step of *n* means that a marker will be placed every *nth* point (not every *n* Hz).

10.3.3.6 No Background

When toggled *On*, the (default) cyan background within the plotting grid is suppressed.

10.3.3.7 Draw Full Grid

When toggled *On*, grid lines are drawn between the larger axis ticks, producing a reference grid on the plot.

10.3.3.8 Log Frequency

When toggled *On*, the frequency axis is displayed in logarithmic format (recommended for wide−band data such as octave−band spectra).

When toggled *Off*, the frequency axis is displayed in linear format (default).

10.3.3.9 No Legend

When toggled *On*, the legend of the plotted curve is suppressed.

10.3.3.10 XRange

The two entry fields specify lower and upper bounds on the X−axis.

If the first (or second) entry field is empty, no lower (or upper) bound is imposed and the lower (or upper) bound is derived from the minimum (or maximum) value of the data to be plotted on the X−axis.

10.3.3.11 YRange

The two entry fields specify lower and upper bounds on the Y−axis.

If the first (or second) entry field is empty, no lower (or upper) bound is imposed and the lower (or upper) bound is derived from the minimum (or maximum) value of the data to be plotted on the Y−axis.

10.3.4 Frequency packing concepts

10.3.4.1 Continuous and discrete excitation

The FRF files of RAYNOISE contain acoustic data as a function of frequency. More precisely, these data are obtained for a set of discrete calculation frequencies.

There are two ways of interpreting these discrete results, depending on the real nature of the acoustic behavior:

- If the excitation is provided at a set of discrete excitation frequencies, such as from tonal sources or a narrow–band definition of the sources, the RAYNOISE results represent the acoustic response at these discrete frequencies. For each frequency present in the excitation, one set of results is provided.
- If the excitation is defined by a continuous spectrum in a certain frequency range, the RAYNOISE results provide either a sampling of the continuous response spectrum, or an average within each frequency band (such as 1/1 or 1/3 octave). The latter is a common case, with band–average sound power values for sources and corresponding field–point results, *eg* at octave band center frequencies.

10.3.4.2 Packing type: third octave and octave bands

It is customary in acoustics to display frequency distributions in 1/3 or 1/1–octave band format. The definition of the octave and third–octave frequency bands as used by RAYNOISE are listed in the following tables 10.3–1 and 10.3–2.

The **Packing Type** determines whether results are added in some way in each of the 1/1– or 1/3–octave bands. If they are not, they are treated as discrete values.

The packing type *Narrow* means that no packing in frequency bands is applied, which is the default. This is the logical approach if the analysis has been performed at octave– or third–octave–band center frequencies and results are to be displayed as spectral lines at those frequencies.

Octave band number	Center frequency	Lower frequency	Upper frequency
1	16	0	22.4
2	31.5	22.4	45
3	63	45	90
4	125	90	180
5	250	180	355
6	500	355	710
7	1000	710	1400
8	2000	1400	2800
9	4000	2800	5600
10	8000	5600	11200
11	16000	11200	22400
12	31500	22400	infinite

Table 10.3–1 Octave band upper, lower and center frequencies in RAYNOISE

Reference: Pierce, A.D., *Acoustics – An Introduction to Its Physical Principles and Applications* (Acoustical Society of America, 1989).

Third–octave band number	Center frequency	Lower frequency	Upper frequency
1	16	0	18
2	20	18	22.4
3	25	22.4	28
4	31.5	28	35.5
5	40	35.5	45
6	50	45	56
7	63	56	71
8	80	71	90
9	100	90	112
10	125	112	140
11	160	140	180
12	200	180	224
13	250	224	280
14	315	280	355
15	400	355	450
16	500	450	560
17	630	560	710
18	800	710	900
19	1000	900	1120
20	1250	1120	1400
21	1600	1400	1800
22	2000	1800	2240
23	2500	2240	2800
24	3150	2800	3550
25	4000	3550	4500
26	5000	4500	5600
27	6300	5600	7100
28	8000	7100	9000
29	10000	9000	11200
30	12500	11200	14000
31	16000	14000	18000
32	20000	18000	22400
33	25000	22400	28000
34	31500	28000	infinite

Table 10.3–2 Third–octave band upper, lower and center frequencies as used in RAYNOISE

10.3.4.3 Packing Mode

The packing mode determines the algorithm to be applied to the computation of frequency–band data values when frequency packing is applied:

$$RMSA = (root - mean - square - of - amplitudes) \Rightarrow p_{band} = \sqrt{\frac{1}{n}\sum_{\lambda_1}^{\lambda_2} p^2(\lambda)} \qquad \text{Eqn 10.4–1}$$

$$Added = (sum - of - amplitudes) \cdots\cdots\cdots\cdots \Rightarrow p_{band} = \sum_{\lambda_1}^{\lambda_2} p(\lambda) \qquad \text{Eqn 10.4–2}$$

$$Average = (averaged - amplitudes) \cdots\cdots\cdots\cdots \Rightarrow p_{band} = \frac{1}{n}\sum_{\lambda_1}^{\lambda_2} p(\lambda) \qquad \text{Eqn 10.4–3}$$

$$Continuous = (integrated - amplitudes) \cdots\cdots\cdots \Rightarrow p_{band} = \int_{\lambda_1}^{\lambda_2} p(\lambda)d\lambda \qquad \text{Eqn 10.4–4}$$

where λ_1 and λ_2 are the lower and upper frequency of the band.

The packing mode *Added* corresponds to a discrete frequency response spectrum, while the packing mode *Continuous* corresponds to a continuous frequency response spectrum.

The packing mode *Average* is used to average the results in the different bands, and is used when excitations are defined in octave or third–octave bands. However, the values at different frequencies are treated as coherent (added in a complex sense).

The *RMSA* (Root Mean Square Amplitude) packing mode is the same as *Average*, except that the root–mean–square procedure is used for the additions. This is logical for non–coherent spectra and is the default packing mode.

Note! The different packing modes result in very different values on the curves. Make sure you select the one which is appropriate to your application.

10.3.5 dB filters

A, B, C and D filters are the empirical corrections that can be applied to the linear dB level to take into account the subjective response of the human ear at different frequencies. The dB−corrections for the different filters are shown in Figure 10.3−4 for a frequency range from 10 Hz to 20 000 Hz. They conform to the standard values from ANSI S1.42 *Design Response of Weighting Networks for Acoustical Measurements*.

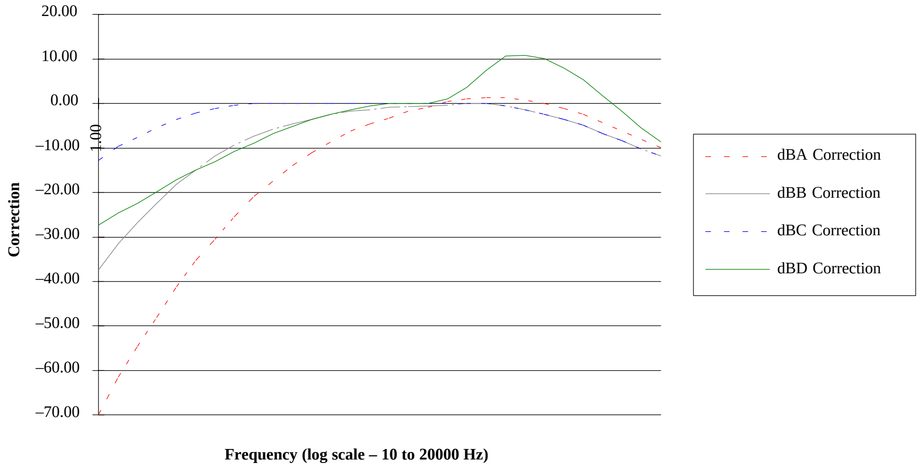


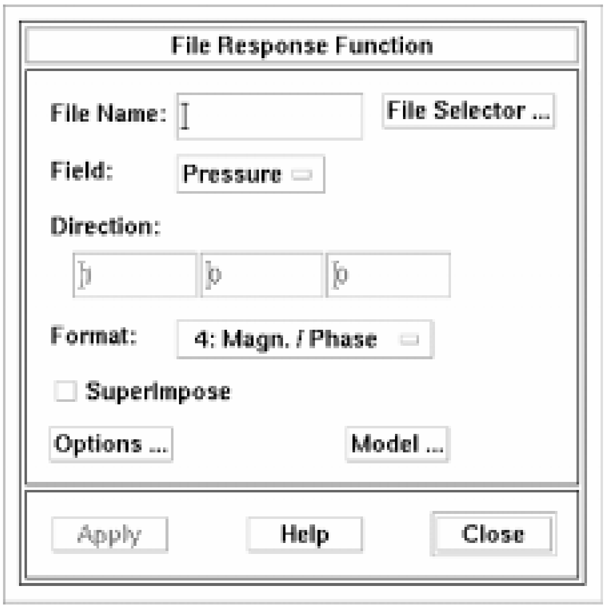
Figure 10.3–4 A, B, C and D dB–filters (corrections in dB).

If the frequency is lower than 10 Hz, RAYNOISE applies the correction at 10 Hz, if the frequency is higher than 20 kHz, RAYNOISE applies the correction at 20 kHz.

Beware of the risk of double−filtering the results if the selected result type already includes a filter (*ie* SPLA). Use the dB linear result and *then* apply the filter.

10.3.6 Plotting data from a file

The **File Function...** entry of the **Postprocess...** menu opens the variant of the *Response Function* dialog which plots the numeric data contained in an external file:



File Response Function

File Name:

Field: Pressure

Direction: 1 0 0

Format: 4: Magn. / Phase

☐ Superimpose

Figure 10.3–5 Response function dialog for data from a File

The **File** entry field allows the user to specify the name of the file for which numeric data should be plotted. **File Selector...** opens the sub–dialog in which the current file directory is listed, and one can select a file or move to other directories.

The data in the file should normally represent the variation of some quantity with frequency. The file should be in the RAYNOISE result–function format (typically, for pressures; typically, given the filename extension .RES: see *Appendix 4: Frequency response function files* for more information). This will often be the output from a Response calculation, or from operations with the **Combine** tool (see Section 11.4 on Page 11–10), using the output format Response Function.

The remaining buttons for **Format**, **Superimpose**, **Options...**, **Model...**, **Apply**, **Help** and **Close** have the same effects as in the **Point Function...** dialog. (See *Common curve concepts*, Section 10.3.2 on Page 10–28).

10.3.7 Plotting a table

The **Table Function...** entry of the **Postprocess...** menu opens the variant of the *Response Function* dialog which plots the numeric data contained in a table stored in RAYNOISE internal memory.

The **Number** entry field allows the user to specify the number of the table for which numeric data should be plotted. **Table Selector...** opens the sub–dialog in which the currently–available tables are listed by name, and one can be selected.

The remaining buttons for **Format**, **Superimpose**, **Options...**, **Model...**, **Apply**, **Help** and **Close** have the same effects as in the **Point Function...** dialog. (See *Common curve concepts*, Section 10.3.2 on Page 10–28).

Figure 10.3−6 shows an example of a curve plot in Real Part format of a table containing transmission loss data.

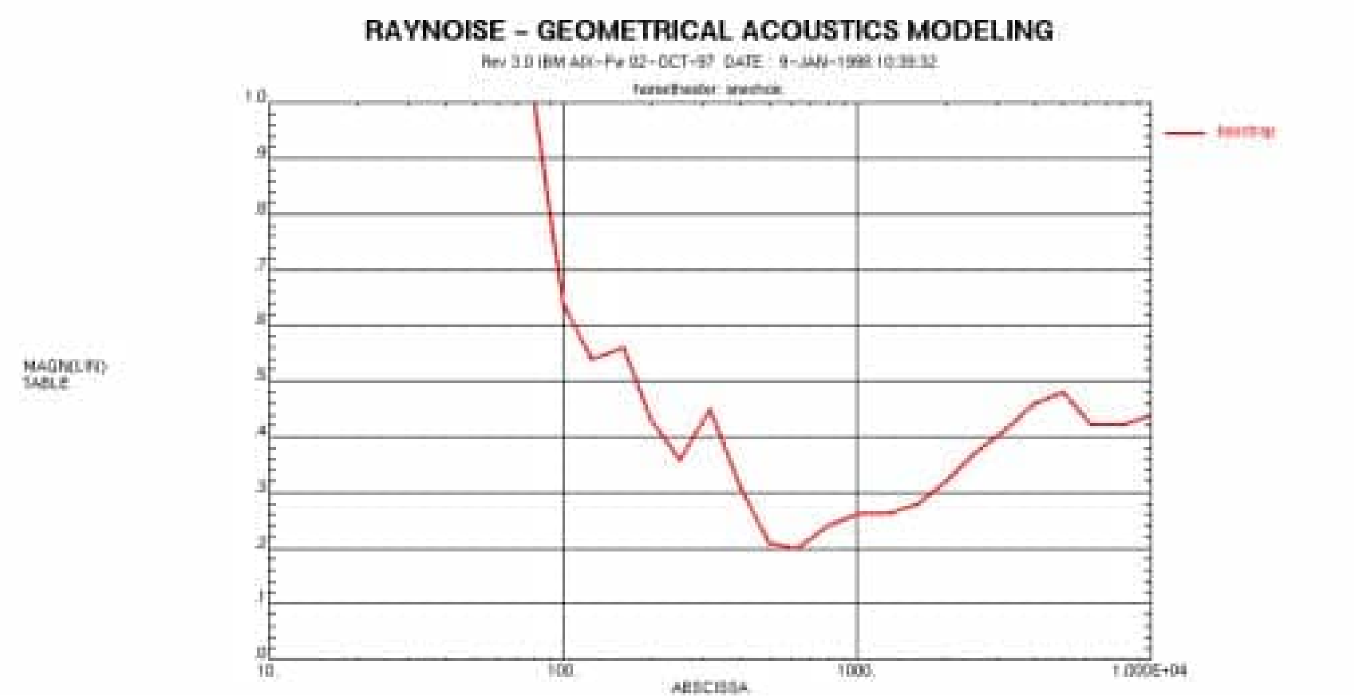


Figure 10.3–6 Example of a table plot with the Table Function command

10.4 Echogram displays

10.4.1 Function

An echogram is a special type of XY–graph, which shows the pressure amplitudes of the successive 'wavefronts' of the impulse response passing through a receiver point, as a function of time. Each reflection line shows the amplitude and arrival time of the intersection of the beam cross–section with the selected receiver (field point).

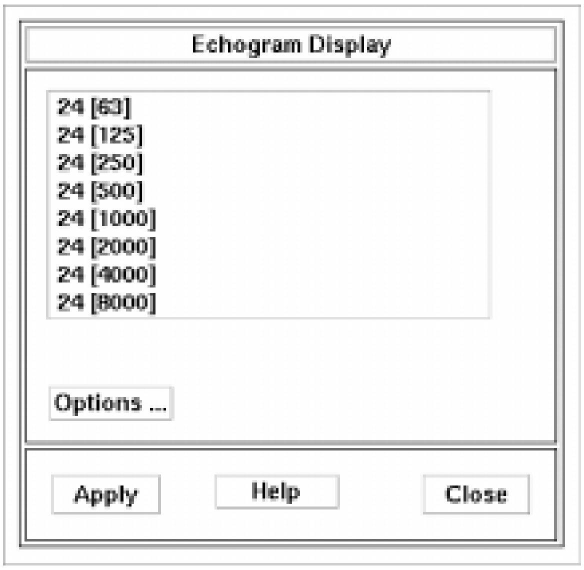
The corresponding ray paths (beam centerlines) are plotted on the model view.

For background information on the method, see the discussion of beam tracing in the section on *Hybrid methods*, Section 3.1.4 on Page 3–6.

Echogram parameters and Histogram parameters affect the storage of the data presented in echogram displays: see *Effect of calculation parameters on data storage*, Section 9.5.7 on Page 9–17.

10.4.2 Echogram display procedure

The **Echogram...** entry of the **Postprocess...** menu opens the curve–plotting dialog adapted to echogram plots:


Echogram Display

24 [63]
24 [125]
24 [250]
24 [500]
24 [1000]
24 [2000]
24 [4000]
24 [8000]

Figure 10.4–1 Echogram display dialog

Any echogram that was saved in the database (using the **Echo Saving** switch during the Mapping calculation, see Section 9.3.2 on Page 9–7) can be selected and displayed as illustrated in Figure 10.4–2. The available echograms are shown in a scrollable list in the form of point–frequency pairs below each other and are identified by an integer indicating the receiver (field point number) followed by the frequency in square brackets [*f*]. To select an echogram, click on its reference in the list and then enter other plotting parameters as desired, before **Apply** to make the plot.

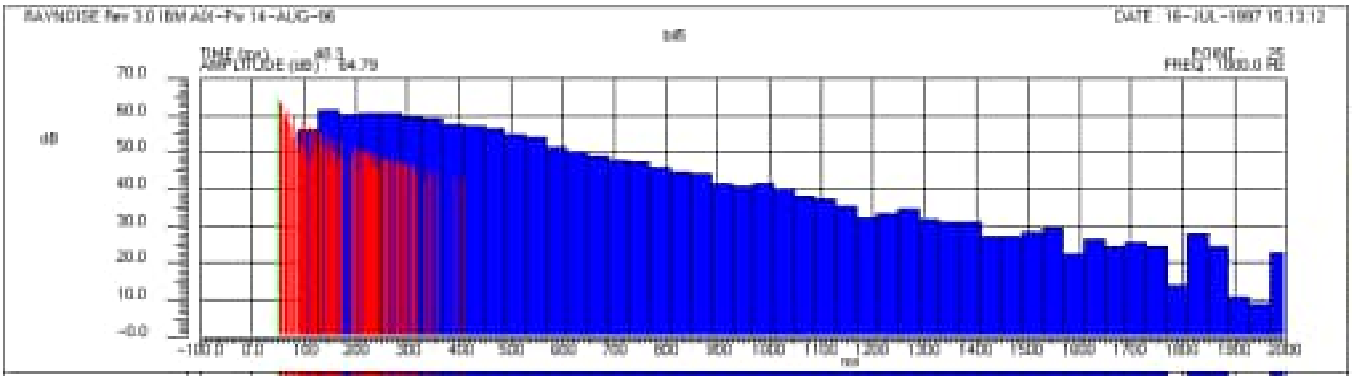


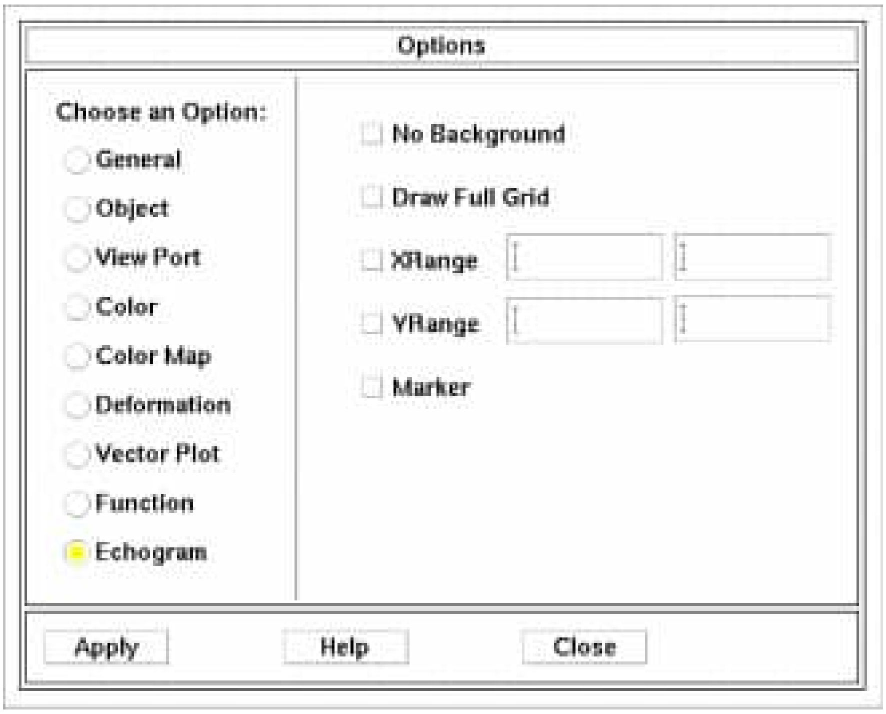
Figure 10.4–2 Example of an echogram window

One reflection is always current and depicted in green. Its values (arrival time and amplitude) are indicated in the header region of the display.

When the echogram window is active (highlighted by clicking with the mouse) it is possible to browse through the various reflections of the echogram using the following keys:

- **Right Arrow Key**: selects the right neighboring (next later) reflection after the current reflection.
- **Left Arrow Key**: selects the left neighboring (next earlier) reflection after the current reflection.
- **Right Shift Key**: moves the current reflection to the right in steps of 50.
- **Left Shift Key**: moves the current reflection to the left in steps of 50.

The **Options** button opens the Graphical Options dialog with the Echogram switch setting, which controls the display attributes of an echogram plot:



Options

Choose an Option:

○ General
○ Object
○ View Port
○ Color
○ Color Map
○ Deformation
○ Vector Plot
○ Function
● Echogram

□ No Background
□ Draw Full Grid
□ XRange
□ YRange
□ Marker

Figure 10.4–3 Graphical Options dialog for echograms

For instance, use of the X– and/or Y–range entries allows zooming of the plot. When the **Marker** toggle button is *On*, markers are added to the Ray–path visualization.

10.4.3 Ray path visualization

For each current reflection, the equivalent ray path is drawn in green on the model view(s) in the active viewport of the Main Window. Ray paths of previously–selected reflections are re–drawn in red, if the echogram is 'browsed'. When the **Marker** toggle button (of the Graphical Options dialog, Echogram section) is *On*, markers are added at the intersections of ray–paths and surfaces. The display of previously–selected ray paths is cleared and only the current selection is displayed, if the model view is re–drawn (*eg*, due to a change of view point) or a different viewport is activated (see *Echogram display procedure*, Section 10.4.2on Page 10–41). An example is shown in Figure 10.4–4 below.

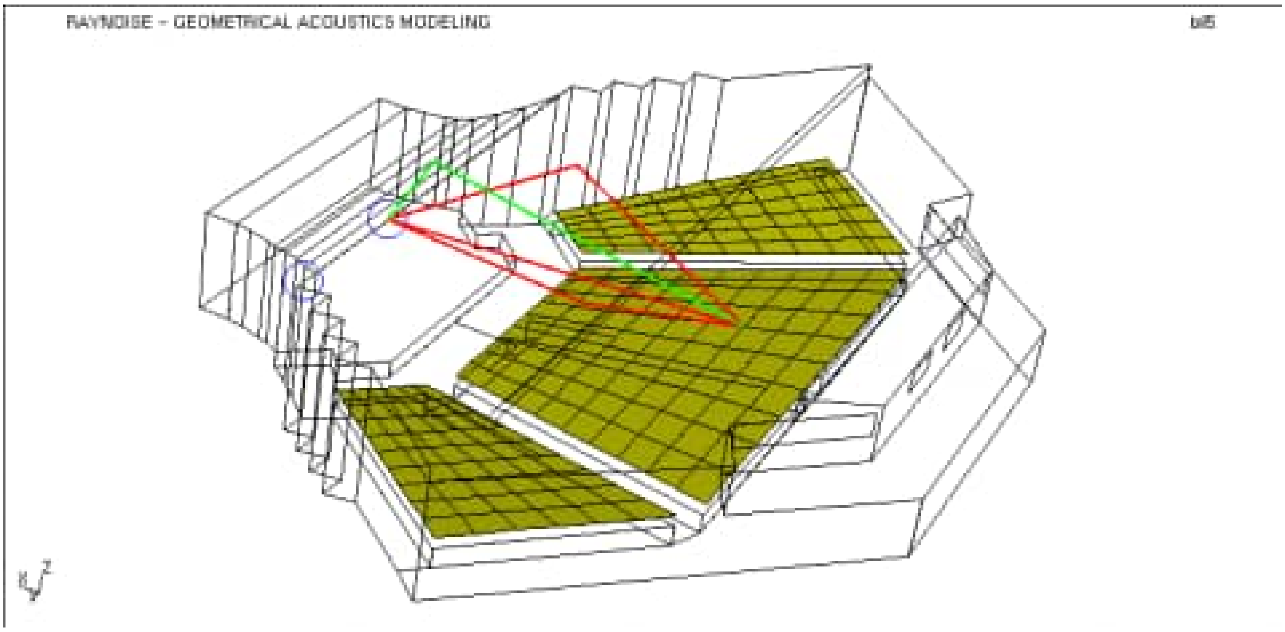


Figure 10.4–4   Ray–path visualization

11

Tools

This chapter describes various utilities and similar functions in RAYNOISE, including:

- *Job options (debug level, test mode, ...)*
- *Job information (occupied space, time used, ...)*
- *The COMBINE tool for processing response functions*
- *Reference values for dB*
- *Environment variables*

11.1 Overview

The **Tools** main menu groups all the commands used to monitor and control the execution of RAYNOISE, together with other utilities.

These tools are available through the following sub menus:

Job Options

- Debug Level
- Echo
- Test
- Macro

Job Info

- CPU Time
- Occupied Space

The following dialogs control general utilities which can be applied irrespective of the nature of the model being processed by RAYNOISE:

- Combine
- Reference
- Open shell
- File editor
- Environment variables
- Journal

In addition, the RAYNOISE Command Language offers some related commands:

- CALCULATOR Command Syntax
- WAIT Command Syntax

11.2 Job options

11.2.1 Summary

The commands of the **Tools, Job Options...** menu control the way in which a RAYNOISE session is executed. The following entries are available:

- **Debug level** controls the amount of information sent by RAYNOISE to the output unit (Echo window or Logfile).
- **Echo** turns the echoing of commands in the RAYNOISE output unit on or off.
- **Test** controls the execution of RAYNOISE calculations for data–checking
- **Macro** controls the recording and playback of a RAYNOISE interactive session.

11.2.2 Debug level

More information on data transfer and manipulation or intermediate results during the calculation process are printed when the **Debug** mode is switched on. This is done by changing the debug level from the default level 0 (No Debugging) to higher levels:

Debug Level	Label	Comment
0	NoDebug	No debug information (default)
1	Small	Summary information on the calculation steps being performed
2	Average	As for level 1 plus space allocation information
3	Big	Detailed information on the calculation steps being performed
4	Huge	Complete debug information

Table 11.2–1 RAYNOISE debug levels

Command example:

```
RAYNOISE> Debug 2 Return
```

11.2.3 Echo

Switching the **Echo** mode on echoes commands to the output unit (screen or file specified as Logfile). This makes the content of log files easier to read. The echoed command is always expressed in RCL syntax, whether it was created directly on the command entry line, read from a file, or created using the GUI menus. The echoed command is preceded by the indicator ‘RCL>’.

By default, Echo is *Off*. When it is *On*, it can be turned off by entering the menu again, which will then present the sequence **Tools, Job Options, NoEcho**.

Command examples:

```
RAYNOISE> Echo Return
RAYNOISE> NoEcho Return
```

11.2.4 Test

It is often interesting to check the command syntax and data consistency before starting a calculation, especially when working with command files.

Switching the **Test** mode *On* prevents the execution of any calculation procedures. All instructions are read, decoded and interpreted but calculation procedures are not started. This allows a thorough verification of all command files without spending unnecessary computing time. The message *** *TESTING MODE* *** *Nothing Done* is displayed.

When Test mode is *On*, it can be turned off by entering the menu again, which will then present the sequence **Tools, Job Options, NoTest**.

Command examples:

```
RAYNOISE> Test Return
RAYNOISE> NoTest Return
```

- See also: *Reading a command file*, Section 4.6.4 on Page 4–14.

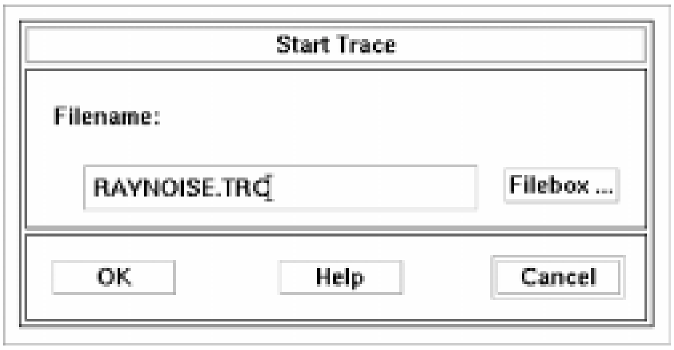
11.2.5 Macro

11.2.5.1 Purpose

A macro is a sequence of GUI events (menu selection, dialog input, model translation, rotation, zoom, choice of results display options). Macro files are sometimes referred to as trace files.

The **Tools, Job Options, Macro** menu is used to:

- start the creation of a macro;
- end the recording of GUI events in the macro;
- replay a macro.



Start Trace

Filename:

RAYNOISE.TRC

Figure 11.2–1 Tools, Macro recorder dialog

Macros are used:

- to avoid repeating common sequences of events;
- to create running demonstrations;
- as a communication tool: macro files are ASCII files that can easily be exchanged with other users or RAYNOISE technical support personnel, to describe the way RAYNOISE was used in a specific application.

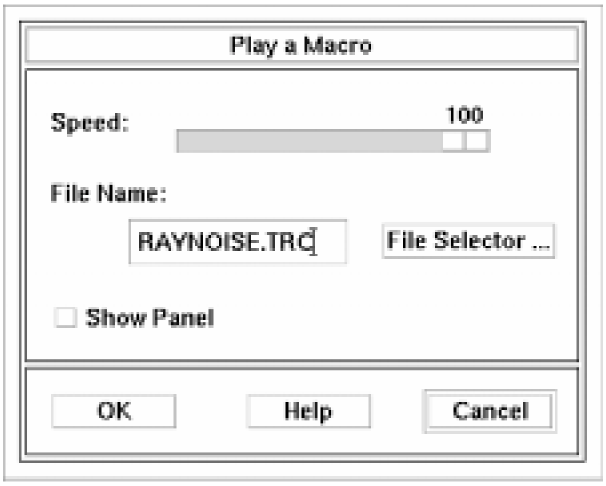
11.2.5.2 To Start/Stop recording a macro

The command **Tools, Job Options, Macro, Start Recording...** presents a dialog in which the name of the macro file can be entered. The default macro file name is RAYNOISE.TRC. All GUI events will be recorded in this file in an ASCII format. Once the mouse is returned to the main window, the recording commences.

The recording of events is stopped by returning to the dialog and picking **Tools, Job Options, Macro, Stop Recording**.

11.2.5.3 Playing a macro file

A macro file can be replayed using **Tools, Job Options, Macro, Play Macro**, which opens a dialog where the macro file name can be entered or the **File selector...** dialog opened.



Play a Macro

Speed: 100

File Name: RAYNOISE.TRC

Figure 11.2–2 Tools, Macro playback dialog

The *Speed* slider controls the initial speed of playback of the GUI events stored in the file. If the **Show Panel** button is activated, a dialog appears during the playback, with buttons similar to a tape recorder (*Play, Pause, Stop*) and an interactive speed–control slider.

Hint: Before initiating the replay of a macro file, it is recommended to take care to put RAYNOISE into the same condition as it was when the recording was started (same open and active models, same view point with the same objects visible, ...) otherwise meaningless behavior can occur, which in the worst case may corrupt some data and at best may give strange graphical presentations. Often, the macro will begin with the opening of a new or existing database file.

11.3 Job information

11.3.1 Description

The **Tools, Job Info** menu gives information about the computer resources used by RAYNOISE. The following entries are available:

- CPU time
- Occupied space

11.3.2 CPU time

The **Tools, Job Info, Time** menu gives an indication about the CPU time used by RAYNOISE:

- total CPU time since beginning the current RAYNOISE session;
- CPU time since the last call of the timer.

Times are given in seconds and in hours, minutes and seconds.

The CPU statistics are sent to the output unit (Echo window or logfile).

In addition, some RAYNOISE commands automatically return time statistics, in particular the calculation procedures.

Command example:

```
RAYNOISE> TIME Return
```

11.3.3 Occupied space

The **Tools, Job Info, Space** menu returns information about the memory usage of the current RAYNOISE session: the total number of words used and the percentage of the total available memory space used are written to the output unit (Echo window or file specified in the Logfile command).

In Debug mode (see Section 11.2.2 on Page 11–4) RAYNOISE also prints the allocated memory space, in words, and the available (free) memory space, as a percentage of the total space.

Command example:

```
RAYNOISE> SPACE  Return
```

- See also *Allocating memory to RAYNOISE*, Section 9.10.2 on Page 9–29.

11.4 Combine

11.4.1 Purpose

The **Tools, Combine** menu enables combination of response functions (stored in the database) using standard operators. It creates new response functions, which can be saved in the database (as Tables) or Exported in files. It can be useful, for example, for computing the FRF of transmission loss (*eg* the ratio between the results at two field points) or other functions of the results. The Combine dialog (and the corresponding RCL sequences) can be thought of as a calculator for response functions or spectra.

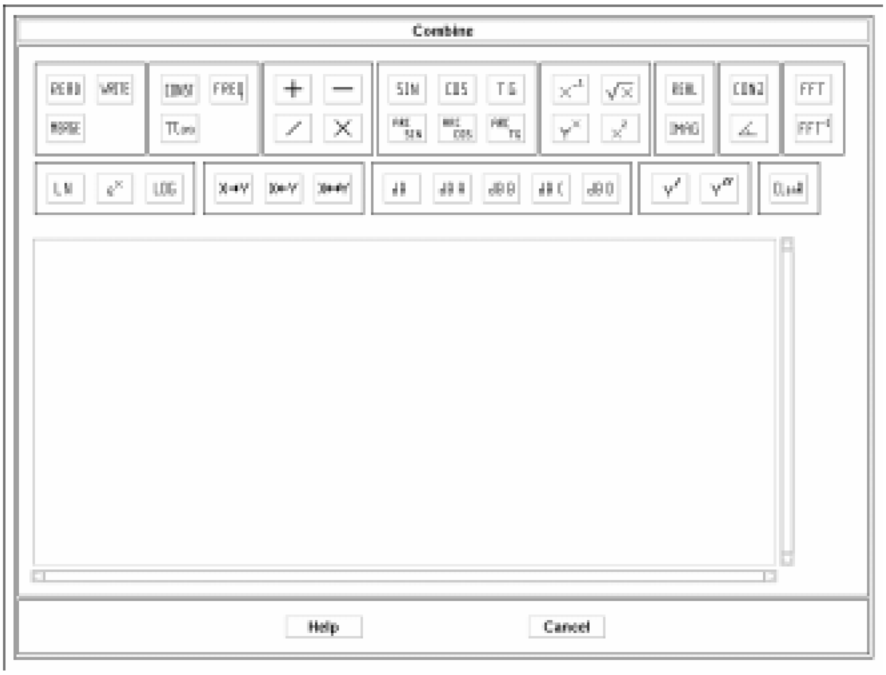


Figure 11.4–1 Tools, Combine dialog

11.4.2 Procedures in Combine

Combine works with four registers named X, Y, Z and T, each containing a complex vector representing one acoustic variable at the different frequencies. Operations can be performed on the registers in much the same way as in an HP−type calculator.

Specific data are read into the top register (X) by the **Read** button, where pressure or other results (depending on the calculation and storage options used) can be selected, as functions of the abscissa (A−register) which is typically frequency. The first set of values read into the X−register defines the values of the abscissa:

any further data to be read must follow these initial abscissa values, or the original abscissae are overwritten. Data with different abscissae (such as different frequency ranges) can be combined with the **Merge** command. (To be logical, they must be data of the same type).

Once data have been loaded into one or more registers, operations can be performed like:

- add, multiply, subtract, divide registers;
- take the square, the square root, the logarithm, the exponential of a register;
- create registers with constant values, with the frequencies or angle values, with A, B, C or D filter corrections;
- create registers containing the third−octave or octave band values.

The operations are applied to the current values of the X− and Y−registers. The results of an operation, such as multiplication, are substituted for the values in the X−register, the values in the Y−register are removed, and any values in the Z− and T−register are moved to the next register above ('shift−to−left' or 'stack' principle). The values are manipulated as complex numbers, using the so−called 'post−fix' or RPN notation, like an HP calculator: first, values are loaded into registers, then an operator is applied, finally the result of the operation is written to a register. The exception to the last step is the Integrate operator (only accessible by the command language) for which the results are written on the output unit (Echo window or logfile).

11.4.3 Reading data into the Combine tool

Results are retrieved directly from the database and entered into the X−register by means of the **Read** dialog, which is activated by the **Read** button. If the X−register already contains data, they are shifted to the Y−register, with the parallel 'shift−to−right' rule applied to the other registers if they also contain data.

The information transferred to the X−register depends on the settings of two pull−down menus. Point results, File results, or Table data can be selected. A point or a table must be selected by number, or the subsidiary selection tools may be used (point picking, table selector). A file is selected by filename (including the full path, if the file is not in the current working directory) or the File Selector may be used.

In the case of field−point data, the data type of the values transferred to the X−register can be selected from the second pull−down, either *Pressure*, *STI* or *RT* data. These choices are not relevant to data from a table or a file, where only one set of data will be present.

The label **Merge** will merge two responses containing different frequency steps (or other abscissae) into one single stack. The frequencies in the A–register and in the merged file or table will be sorted in ascending order and duplicated frequencies will be eliminated, producing a new A–register. The data in the X–register are sorted in the same way. The average value of the two data values is used for any duplicated frequencies.

11.4.4 Mathematical operators in the Combine tool

11.4.4.1 Types of operators

The mathematical operators in the Combine dialog can be divided into two major categories :

- Operators requiring a single argument

These operators act only on the data in the X register and store the result in the X register. Typical examples of single–argument operators are ROOT, SQUARE, INVERT, ...

- Operators requiring two arguments

These operators combine the data in the X–register with the data in the Y–register and store the result in the X–register. After this operation, the data in the other registers is also shifted one register to the left.

11.4.4.2 Order of the arguments

The order of the arguments in a two–argument operation is always : Y–register followed by X–register, unless otherwise noted. For instance, the DIVIDE operation will divide the contents of the Y–register by the contents of the X–register.

11.4.4.3 Available operators

The following groups of mathematical operators are available:

- Standard arithmetic operators: ADD, SUBTRACT, MULTIPLY, DIVIDE
- Power operators: SQUARE, ROOT, INVERT, POWER
- Logarithmic operators: LOG, LN, EXP
- Trigonometric operators: SIN, COS, TAN, ARCSIN, ARCCOS, ARCTAN

- Complex value operators: REAL (returns real part of complex number), IMAG (returns imaginary value of complex number), CONJ (returns conjugate value of complex number)
- Derivative operators: Y' (first derivative) and Y" (second derivative), where the derivation is applied with respect to the abscissa values (A−register).
- Special purpose operators: CONST (to enter constant values), PI (to enter the value 3.1415...), FREQ (returns frequency values as stored in the A−register), ANGLE (returns angle values as stored in the A−register)
- Register copy operators: X−to−Y, Y−to−X, Exchange X and Y
- Clear operator: empties all data from all registers

In addition, the Combine dialog contains some operators which are very useful for calculations related to acoustic quantities :

- Decibel operators : dB (conversion to dBlin values), dBA, dBB, dBC, dBD for respectively A−, B−, C− and D−filter dB correction
- Frequency−Time domain conversion operators: FFT (Fast Fourier Transform: time domain to frequency domain) and FFT^{-1} (Backwards or Inverse Fast Fourier Transform: frequency domain to time domain)

All these functions are clearly identified in the Combine dialog by buttons containing the corresponding function label.

The command INTEGRATE can be used to give integrals of all the values in each of the registers. This command is not available in the Combine dialog, but must be accessed via the RCL:

```
RAYNOISE> COMBINE
COMBINE> READ ...
COMBINE> INTEGRATE  Return
```

11.4.5 Writing results from the Combine tool

11.4.5.1 Output of X−register

The contents of the X−register can be stored in several ways by means of the Write dialog, which is opened by clicking the WRITE button.

11.4.5.2 Combine, Write dialog

The following formats are available

- *Response Function*

The X−register contents are stored in an external response function file in the form of complex pressure data as a function of A−register values. The name of the external file is specified in the File entry field, while the internal name to be used in Graphical Postprocessing plots is specified in the Name entry field.

See *Frequency response function files*, Section 16.1 on Page 16−3, for details of the file format.

- *Sound*

The X−register contents are stored in an external file using a sound format determined by the SOUNDFORMAT environment variable. This file can then be auralized using sound editor programs provided on UNIX workstations or on PCs. Note that this form of auralization is not the same as the standard convolution and auralization method provided in the *Analysis* menu, since it requires a frequency−response function to be produced in some way (by a series of commands in the Combine dialog) and then to be converted to time−domain data by inverse FFT, before writing the external file.

- *Function File*

The X−register contents are stored as an external file, containing the data in a three−column free format: A−register values and the real and imaginary parts of the X−register.

- *Table*

The X−register contents are stored as a table in the model database, whereby the contents of the A−register are used to define the table abscissa values.

Table number and table name are specified in the corresponding entry fields.

11.4.6 Visualization of results from the Combine tool

Results created by the Combine dialog can be visualized in two ways:

- Write the results to an external Response Function file and use the **Post−process, File Function...** capability (see Section 10.3.6 on Page 10−38).

- Write the results to a Table and use the **Postprocess, Table Function...** capability (see Section 10.3.7 on Page 10−39).

11.5 Reference values

11.5.1 dB values

Acoustic quantities are often presented using their dB value. The dB value of a quantity A is calculated by the relation:

$A_{dB} = A_{factor} \log (A_{eff}/A_{ref})$

Where:

A_{dB} is the value of A expressed in dB

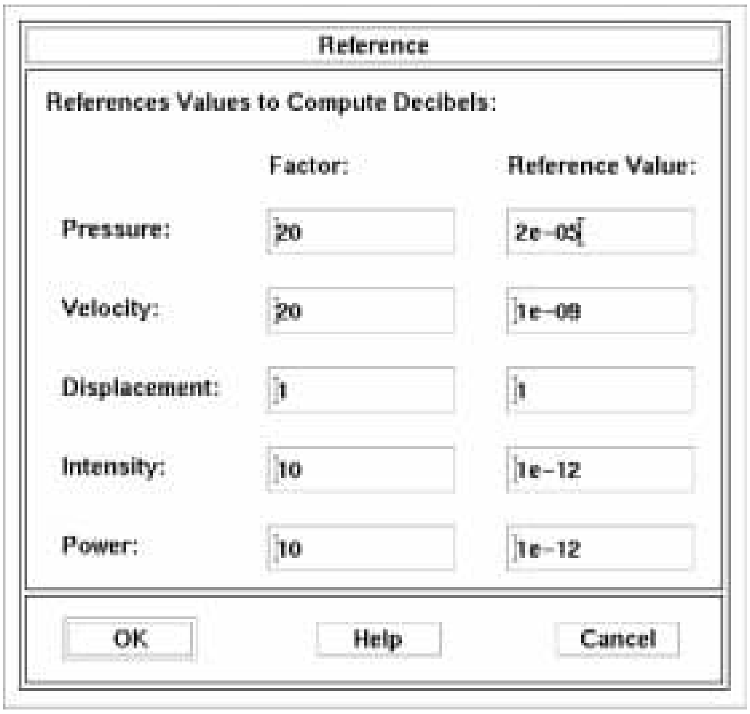
A_{factor} is the multiplier of the logarithmic function

A_{eff} is the ‘rms’ value of A = A/sqrt(2)

A_{ref} is the reference value of A

11.5.2 Changing reference values

A_{factor} and A_{ref} can be defined in the **Tools, Reference** dialog:



Reference

References Values to Compute Decibels:

	Factor:	Reference Value:
Pressure:	20	2e-05
Velocity:	20	1e-08
Displacement:	1	1
Intensity:	10	1e-12
Power:	10	1e-12

Figure 11.5–1 Tools, Reference dialog

Although only pressure reference values are relevant to the presentation of results calculated in RAYNOISE itself, the other values are also available for the sake of completeness (for instance, for cases where frequency–response results are loaded from other programs such as SYSNOISE).

Default values of A_{factor} and A_{ref} are defined in the RAYNOISE set–up file by the following environment variables:

• EREFERENCE	*edbfactor*	for power
• PREFERENCE	*pdbfactor*	for pressure

For more information, see *Units in RAYNOISE*, Section 7.8 on Page 7–35.

11.6 Open shell

This menu opens a new Unix window. The Unix command used to perform the operation is contained in the Environment Variable SHELLCMD (see Section 11.8.3.68 on Page 11−34). The precise operation may be system−dependent.

11.7 File editor

This menu launches a file editor in a new Unix window. The Unix command used to perform the operation is contained in the Environment Variable EDIT (see Section 11.8.3.24 on Page 11−25). The precise operation may be system−dependent.

11.8 Environment variables

11.8.1 Function

Environment variables are used to customize RAYNOISE and fix user preferences. Environment variables are defined at four different levels:

- in the source code (absolute defaults)
- in the *Set–up file* RAYNOISE.STP read at start–up. (See Section 11.8.4 on Page 11–39).
- in the *Profile file* RAYNOISE.PRF file read at start–up after the set–up file. These values override those of the set–up file. (See Section 11.8.5 on Page 11–40).
- within RAYNOISE, environment variables can be modified in the **Tools, Environment Var.** dialog:

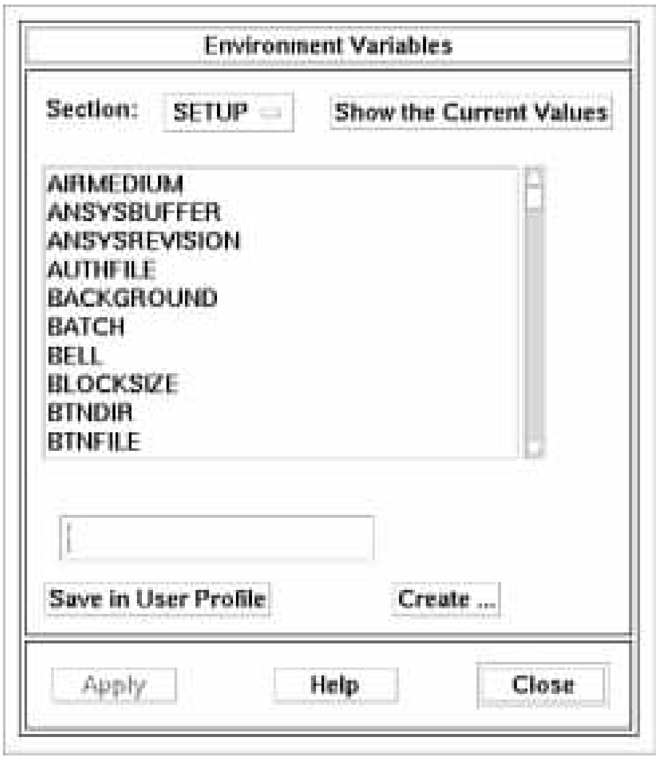

Section: SETUP

Figure 11.8–1 Tools, Environment Variables dialog

11.8.2 Changing environment variables

To modify an environment variable:

- open the **Tools, Environment Var.** dialog
- select the environment variable in the list
- change its value, using the entry field in the dialog; in some cases, a toggle button appears, *True* (button in/colored) or *False* (button out/grey)
- confirm the change by clicking on the **Apply** button.

The changes remain active for the rest of the current RAYNOISE session, unless changed again. To save the changes permanently (in the user's *Profile file*) use the **Save In User Profile** button (see Section 11.8.5 on Page 11–40).

Note that many environment variables, especially those related to graphical display options, colors, etc, are also modified by the use of Toolbar buttons (face–paint/wireframe toggle, rotation, ...) or Graphical options dialogs.

11.8.3 List of user–controlled environment variables

All relevant environment variables are listed below in alphabetical order. Some additional environment variables may be present in the list: these are used by other LMS programs, not RAYNOISE. Default values are values defined in the RAYNOISE.STP file for a typical installation.

11.8.3.1 AIRMEDIUM

Type: Character

Definition: If True, the fluid through which the rays pass has the properties of air.

Default value: False

11.8.3.2 AUTHFILE

Type: Character

Definition: Full path name of the RAYNOISE.LIC authorization file

Default value: installation_path/RAYNOISE.LIC

11.8.3.3 BACKGROUND

Type: Integer

Definition: Color number for background color:

0	–	Black
1	–	White
2	–	Red
3	–	Green
4	–	Dark Blue
5	–	Light Blue
6	–	Purple
7	–	Brown

Higher color numbers are available but are installation–dependent.

Default value: 1

11.8.3.4 BATCH

Type: Character

Definition: Run mode (VMS only)

Default value: FALSE

11.8.3.5 BELL

Type: Character

Definition: If set ON, important error messages produced by RAYNOISE are accompanied by an audible signal.

Default value: ON

11.8.3.6 BTNDIR

Type: Character

Definition: directory path of the active button file

Default value: installation dependent

11.8.3.7 BTNFILE

Type: Character

Definition: location of the active button file

Default value: installation dependent

11.8.3.8 BTNWINPOS

Type: 2 Integers

Definition: Defines the initial position of the button window on the screen.

Default value: 1200, 30

11.8.3.9 BUFFERSIZE

Type: Character

Definition: Size of buffer (in words) used by RAYNOISE during read/write operations

Default value: 80000

11.8.3.10 CRVWINPOS

Type: 4 Integers

Definition: Defines the initial position and size of the curve (spectrum) window on the screen.

Default value: 100, 300, 200, 300

11.8.3.11 DATADIFFUSIONFILE

Type: Character

Definition: Full path name of the file containing diffusion parameters.

Default value: installation_path/RAYNOISE.DIF

11.8.3.12 DATASABINEFILE

Type: Character

Definition: Full path name of the file containing absorption data (Sabine coefficients).

Default value: installation_path/RAYNOISE.SAB

11.8.3.13 DATASOURCEPOWERFILE

Type: Character

Definition: Full path name of the file containing the powers of sources.

Default value: installation_path/RAYNOISE.SPW

11.8.3.14 DATATRANSMISSIONFILE

Type: Character

Definition: Full path name of the file containing transmission loss parameters.

Default value: installation_path/RAYNOISE.TL

11.8.3.15 DIALOGPOS

Type: 2 Integers

Definition: Defines the initial position of a dialog when first placed on the screen.

Default value: 100, 130

11.8.3.16 DIFFRACT

Type: Character

Definition: If set ON, any diffraction edges defined in the model will be taken into account in the ray–tracing.

Default value: OFF

11.8.3.17 DIFFUSE

Type: Character

Definition: If set ON, any defined diffusion will be taken into account in the ray–tracing.

Default value: OFF

11.8.3.18 DXFREVISION

Type: Integer

Definition: Defines the minimum level of AutoCAD release, supported by the DXF interface.

Default value: 12

11.8.3.19 DYNAMICMOTION

Type: Character

Definition: Decides if model is fully plotted (Full) or represented by its features only (Features) during dynamic rotation, translation and zoom

Default value: Features

11.8.3.20 DYNAMICRANGE

Type: Real

Definition: The fall in SPL on a beam cross–section at which the tracing of the beam is abandoned

Default value: 90.0

11.8.3.21 ECHOSTORE

Type: Integer

Definition: The maximum order of reflection of any echo stored in the detailed echogram data requested for storage in a mapping calculation. (Note: can have a major influence on data storage requirements).

Default value: 10

11.8.3.22 ECHOWINPOS

Type: 4 Integers

Definition: Fixes the position and size of the Echo Window of the RAYNOISE GUI. Four numbers must be given for the position of the top left corner (two numbers), the width and the height in pixels.

Default value: 10, 595, 1000, 165

11.8.3.23 EDBFACTOR

Type: Real

Definition: Factor multiplying the logarithm in the expression giving dB values of power

Default value: 10.0

11.8.3.24 EDIT

Type: Character

Definition: Enables or disables the use of a text editor on the Journal file, during the RAYNOISE session.

Default value: installation dependent

11.8.3.25 EREFERENCE

Type: Real

Definition: Reference value used in the calculation of dB value of power.

Default value: 1.0e−12

11.8.3.26 ETHERNETDEVICE

Type: Character

Definition: Name of the Ethernet device

Default value: machine dependent

11.8.3.27 FEATUREANGLE

Type: Real

Definition: During dynamic rotation, translation and zooming, the model is represented only by its feature lines (see DYNAMICMOTION variable, Section 11.8.3.19 on Page 11−24). A feature line is defined as an element edge joining two elements making an angle larger than FEATUREANGLE. This variable controls the degree of refinement of the model displayed during dynamic motion.

Default value: 15.0

11.8.3.28 FOREGROUND

Type: Integer

Definition: Color number for foreground color:

0	–	Black
1	–	White
2	–	Red
3	–	Green
4	–	Dark Blue
5	–	Light Blue
6	–	Purple
7	–	Brown

Higher color numbers are available but are installation–dependent.

Default value: 0

11.8.3.29 FREEFORMAT

Type: Character

Definition: Controls compatibility of a Free format mesh file for transfer to LMS SYSNOISE or re–use with RAYNOISE. If set to 'SYSNOISE' the mesh elements of types PLG3 and PLG4 are converted to TRI3 and QUA4 respectively, and all other polygons (with more than four vertices) are not exported; if set to 'RAYNOISE', all polygons are exported, using their RAYNOISE element type labels.

Default value: 'RAYNOISE'

11.8.3.30 FREEREVISION

Type: Real

Definition: The level of RAYNOISE which created files readable in Free format. If set to 2.1, it enables upwards compatibility of data from previous major version of RAYNOISE. (Free format files are compatible between versions 3.0 and 3.1).

Default value: 3.0

11.8.3.31 GIFDIR

Type: Character

Definition: Directory containing GIF files used by RAYNOISE.

Default value: installation–dependent

11.8.3.32 GRAPHICALCOMMAND

Type: Character

Definition: Decides if graphical commands are written to the RCL History Window.

Default value: FALSE

11.8.3.33 GRAPHICUPDATE

Type: Character

Definition: Decides if the graphical window is updated at every change in model data.

Default value: TRUE

11.8.3.34 HELPDIRECTORY

Type: Character

Definition: Location of RAYNOISE Help files and utilities

Default value: installation–dependent

11.8.3.35 HIGHACCURACY

Type: Character

Definition: If TRUE, causes checks to be carried out on the intersection of beams and the edges of surfaces, to ensure that the ray path for an arrival at a receiver is valid and not due to 'edge leakage'. This improves the quality of results in some cases, at the cost of increased CPU time. If FALSE, these checks are not carried out, but CPU time will be less. See *Approximate beam–tracing*, Section 3.1.5 on Page 3–9, for a description.

Default value: FALSE

11.8.3.36 HINTERVAL

Type: Real

Definition: The duration in milliseconds of the time window used for each ‘bin’ of the echo histogram in a mapping calculation.

Default value: 10.0

11.8.3.37 HLENGTH

Type: Integer

Definition: The number of echo histogram ‘bins’ used in a mapping calculation. The total histogram duration is given by HINTERVAL*HLENGTH (*eg* 40 bins each of 10 ms gives a total duration of 400 ms).

Default value: 40

11.8.3.38 HPKEY

Type: Character

Definition: Specifies if the security code in the Licence File should be checked against the Ethernet address or the machine id. (Only applicable to HP computers).

Default value: Systemid

11.8.3.39 IDEASREVISION

Type: Character

Definition: I–DEAS Revision used. Values supported are *V* (valid for I–DEAS 4, V and VI) and *MA* (for Master Series).

Default value: ’MASTERSERIES’

11.8.3.40 INCLUDEALLFACES

Type: Character

Definition: If TRUE, all edges of all elements are included in the Envelope set. It is then possible to pick any edge graphically, for use as a diffraction edge. (For instance, the edge on the corner between two surfaces, such as the ridge at the top of a roof). If FALSE, only completely–free edges are included in the Envelope set. The variable only has effect when Importing a mesh (*ie*, it must be set *before* Import). See *Diffraction edges*, Section 7.6 on Page 7–25, for more details.

Default value: TRUE

11.8.3.41 INTERFORMAT

Type: Character

Definition: Name of the default format for Import and Export menus.

Default value: Installation−dependent

11.8.3.42 JNLFILE

Type: Character

Definition: Name of the JOURNAL file

Default value: RAYNOISE.JNL

11.8.3.43 LASTMODELFILE

Type: Character

Definition: Retains the name (without path) of the most−recently−opened RAYNOISE database file, to be offered as default in dialogs where this filename is called for.

Default value: None

11.8.3.44 MAINWINPOS

Type: 4 Integers

Definition: Fixes the position and size of the Main Window of the RAYNOISE GUI. Four numbers must be given for the position of the top left corner (two numbers), the width and the height in pixels.

Default value: 256, 0, 940, 705

11.8.3.45 MOUSEBUFFERSIZE

Type: Integer

Definition: Mouse buffer size

Default value: 500

11.8.3.46 NORMIRF

Type: Integer

Definition: The normalization factor used to scale the dynamic range of the sound file produced in the Auralization procedure. Files have a 16–bit dynamic range: the default procedure makes maximum use of this range, depending on the basic impulse response and the source signal in the Convolution. NORMIRF allows the user to control this range scaling.

Default value: 0 (= null, automatic ranging)

11.8.3.47 OUTPUTFRAME

Type: Character

Definition: If true, animation frames are saved as GIF files.

Default value: FALSE

11.8.3.48 OVERWRITE

Type: Character

Definition: If TRUE existing files can be overwritten without confirmation

Default value: TRUE

11.8.3.49 PAINTERALGO

Type: Character

Definition: If TRUE, an improved hidden–line algorithm is used in face–painted display of the mesh. If FALSE, the simpler algorithm (as in Rev 3.1) is used.

Default value: TRUE

11.8.3.50 PATHSTORE

Type: Integer

Definition: The maximum order of reflection of any echo for which ray path information is stored in the detailed echogram data requested for storage in a mapping calculation. Must be equal to or less than ECHOSTORE. (Note: can have a major influence on data storage requirements).

Default value: 5

11.8.3.51 PDBFACTOR

Type: Real

Definition: Factor multiplying the logarithm in the expression giving dB values of pressure.

Default value: 20.0

11.8.3.52 PLDEVICE

Type: Character

Definition: Plot device

Default value: DEFAULT (installation−dependent)

11.8.3.53 PREFERENCE

Type: Real

Definition: Reference value used in the calculation of dB value of pressure.

Default value: 2.0e−5

11.8.3.54 PROMPT

Type: Character

Definition: Prompt

Default value: 'RAYNOISE> '

11.8.3.55 RAYMETHOD

Type: Character

Definition: Selects the use of triangular or conical beams.

Default value: TRIANGULAR

11.8.3.56 RAYS

Type: Integer

Definition: Sets the calculation parameter for the number of rays per source.

Default value: 2000

11.8.3.57 RCLHWINPOS

Type: 4 Integers

Definition: Fixes the position and size of the Echo Window of the RAYNOISE GUI. Four numbers must be given for the position of the top left corner (two numbers), the width and the height in pixels.

Default value: 2, 50, 238, 696

11.8.3.58 RECORDLENGTH

Type: Integer

Definition: Related to RECSIZE. Must be set to 4 if RECSIZE is expressed in words and to 1 if RECSIZE is in bytes.

Default value: 1 (except on SGI hardware, default = 4)

11.8.3.59 RECSIZE

Type: Integer

Definition: Size of direct access file records (in words OR in bytes – see RECORDLENGTH)

Default value: 4096

11.8.3.60 REFLECTIONORDER

Type: Integer

Definition: Sets the calculation parameter for the maximum number of reflections to be calculated for each ray.

Default value: 10

11.8.3.61 RELHUMIDITY

Type: Real

Definition: Relative humidity, if the fluid is air.

Default value: 50.0

11.8.3.62 RFBUFFERSIZE

Type: Real

Definition: Sets the maximum number of samples of a FRF, table or file to be displayed with the CURVE routines.

Default value: 10000

11.8.3.63 RHO

Type: Real

Definition: Acoustic fluid density

Default value: 1.225

11.8.3.64 SAMPLEFREQUENCY

Type: Real

Definition: Sampling frequency for auralization files

Default value: 44100

11.8.3.65 SHADING

Type: 3 Real

Definition: RGB components (between 0. and 1) of the color used for the shaded images

Default value: 0.9, 0.9, 0

11.8.3.66 SHADINGDEPTH

Type: Real

Definition: Intensity (between 0. and 100%) of the shading contrast.

Default value: 50

11.8.3.67 SHADINGRANGE

Type: Integer

Definition: Determines the number of colors used to create the shading effect.

Default value: 40

11.8.3.68 SHELLCMD

Type: Character

Definition: System command used to open a new shell.

Default value: TRUE

11.8.3.69 SHOWSTARTUPDIALOG

Type: Character

Definition: Decides if the initial dialog (New Model, Open Existing Model, *etc*) is opened automatically when RAYNOISE is started

Default value: TRUE

11.8.3.70 SHOWTIPSID

Type: Character

Definition: If TRUE, GUI buttons ID will be displayed instead of tips. This is useful for building the tips file. If FALSE, tips are displayed.

Default value: FALSE

11.8.3.71 SNAPANGLE

Type: Real

Definition: The final model rotation, after dynamic motion, is a multiple of SNAPANGLE (in degrees).

Default value: 10.0

11.8.3.72 SOUND

Type: Real

Definition: Sound Speed

Default value: 340.0

11.8.3.73 SOUNDFORMAT

Type: Character

Definition: Format of sound files produced by the Combine dialog. Possible formats are: AU (Sun, HP), AIFF (SGI), WAV (PC)

Default value: AIFF

11.8.3.74 SPECTRUMRANGE

Type: Integer

Definition: Number of colors used in Graphical Postprocessing contour plots and arrow plots.

Default value: 16.

11.8.3.75 STDEVICE

Type: Character

Definition: Standard graphical output device

Default value: DEFAULT

11.8.3.76 STORELEVEL

Type: Integer

Definition: Sets the volume of results data to be stored in the database. Also installation– and license–dependent: see *Histogram parameters*, Section 9.5.4 on Page 9–15 .

Default value: 1

11.8.3.77 SUBRAYS

Type: Real

Definition: Not currently used by RAYNOISE

Default value: 4

11.8.3.78 SYSDIR

Type: Character

Definition: Full path of system directory

Default value: installation–dependent

11.8.3.79 TABFILE

Type: Character

Definition: Name of file used to save RAYNOISE tables.

Default value: RAYNOISE.TBL

11.8.3.80 TAILCORRECTION

Type: Integer

Definition: Turns automatic tail correction on or off and selects the correction method. (The values 0/1/2 correspond to None/Continuous/Statistical correction). See *General parameters*, section 9.5.5 Page 9–16, for more details.

Default value: 0 (or installation–dependent)

11.8.3.81 TEMPERATURE

Type: Real

Definition: Temperature, when the acoustic medium is air.

Default value: installation−dependent

11.8.3.82 TIMEWINDOW

Type: Real

Definition: Duration (in milliseconds) of the period for which echoes are calculated.

Default value: 2000.0

11.8.3.83 TIPFILE

Type: Character

Definition: File containing menu tips definition

Default value: installation−dependent

11.8.3.84 TMPDIR

Type: Character

Definition: Directory where temporary files will be created

Default value: ' ' (null − *ie* the local directory from which RAYNOISE is started)

11.8.3.85 TOLERANCE

Type: Real

Definition: Defines the relative geometrical tolerance used (for instance) when deciding whether two nodes are coincident. The absolute tolerance is obtained by multiplying TOLERANCE by the maximum mesh dimension.

Default value: 1.0e−3

11.8.3.86 TRACE

Type: Character

Definition: If TRUE, a trace−bar shows calculation progress. If FALSE, it is turned off.

Default value: TRUE

11.8.3.87 USEGUI

Type: Character

Definition: If TRUE, enables use of the Graphic User Interface. If FALSE, an alphanumeric–only mode is invoked (necessary for batch–type execution).

Default value: TRUE

11.8.3.88 USRDIR

Type: Character

Definition: User directory

Default value: ' ' (null – *ie* local directory from which RAYNOISE is started)

11.8.3.89 VIEWPOINT

Type: 3 Real

Definition: Viewing angles for the default Mesh Viewer viewpoint

Default value: –135 , –60 , 0

11.8.3.90 WATCHPAINTING

Type: Character

Definition: If TRUE, the face–painting process can be followed on the screen. If FALSE, the result is displayed at one time as soon as any hidden view is calculated. See also *Watch Paining Progress*, Section 12.1.3.2 on Page 12–4.

Default value: FALSE

11.8.3.91 WATCHROTATION

Type: Character

Definition: If TRUE, the model is rotated dynamically in real time. If FALSE, the result is displayed at one time as soon as any hidden view is calculated. See also *Watch Rotation Progress*, Section 12.1.3.3 on Page 12–5.

Default value: FALSE

11.8.3.92 WKSFILE

Type: Character

Definition: Location of the workstation file containing a description of the plotting devices.

Default value: installation–dependent

11.8.3.93 WORLDVIEW

Type: Character

Definition: Filename of the viewer for the online help.

Default value: installation–dependent

11.8.3.94 XFONT1

Type: Character

Definition: Font for GUI

Default value: '–*–helvetica–medium–r–*–*–10–*–*–*–*–*–*–*'

11.8.3.95 XFONT2

Type: Character

Definition: Font for GUI

Default value: '–*–helvetica–bold–r–*–*–18–*–*–*–*–*–*–*'

11.8.4 Set–up file

The RAYNOISE.STP file contains a definition of most environment variables. These values override those coded in the RAYNOISE executable.

The file is structured in sections, divided by comment lines starting with '{'. Each content line has the form:

ENVIRONMENT variable_name = parameter(s)

All environment variables can be listed in the **Tools, Environment Var...** dialog.

11.8.5 Profile file

For customizing the RAYNOISE working environment, user−specific values for certain environment variables can be placed in the RAYNOISE profile file called RAYNOISE.PRF. This file, which is normally located in the user's home or login directory, is read after the RAYNOISE.STP file and its content over−rides the default values of the .STP file, allowing each user to create his or her personal RAYNOISE environment.

If the RAYNOISE.PRF file is not present, the values defined in the RAYNOISE.STP file will apply.

At any moment during a RAYNOISE session, the current settings of the environment variables can be saved in the .PRF file with the **Save in User Profile** button of the **Tools, Environment Var...** dialog.

The RAYNOISE.PRF file is structured in sections, with additional comment lines starting with '{'. Each section starts with a line of the form:

ENVIRONMENT SECTION section_name

All environment variables defined below this line are attached to the section section_name. Different sections can then be listed selectively in the **Tools, Environment Var...** dialog.

At installation, RAYNOISE defines two sections:

- READR: internal environment variables related to the RCL parser
- SETUP: general purpose environment variables

Each content line has the form:

ENVIRONMENT variable_name = parameter(s)

The profile file can also be created manually. The last line must be the END command, for example:

```
{
{  My own profile file
{
ENVIRONMENT   SOUND=1500
ENVIRONMENT   RHO=1000
END
```

See also *Changing environment variables* (Section 11.8.2 on Page 11–19).

11.8.6 User–defined environment variables

User–defined environment variables can be defined using names not yet assigned to existing variables. They are created using the RCL ENVIRONMENT command and can be defined:

- in the RAYNOISE.STP file
- in the RAYNOISE.PRF file
- during a RAYNOISE run.

Two examples:

```
RAYNOISE> Environment MESH_DIR=/data/projets/meshes Return
RAYNOISE> Input Mesh Format Free File $MESH_DIR/mymesh.fre
```

```
RAYNOISE> Environment INPUT=´Input Mesh Format Free File´ Return
RAYNOISE> $INPUT  myfile.fre
```

The name of the environment variable in the ENVIRONMENT command should NOT be preceded by a $ sign. The $ sign is used subsequently, in the command line, and means: 'the value of'.

11.9 Journal

All RCL commands listed in the RCL History Window are automatically copied to the journal file RAYNOISE.JNL or, if earlier versions of this file exist, to RAYNOISE.JNL.1, RAYNOISE.JNL.2, etc.

The **Journal** menu:

- closes the journal file;
- starts the editor (see Environment Variable *Edit*, Section 11.8.3.24 on Page 11–25). It is then possible to modify this file in order to use it as a command file. After exiting the editor, the file is renamed RNSJNL.OLD (unless otherwise specified using the commands of the editor).
- opens a new RAYNOISE.JNL file

See also *Command syntax*, Section 4.6.3 on Page 4–13.

11.10 Graphics parameters file

The file RAYNOISE.WKS stores the graphics parameters which control the drivers used when making plots and other graphical output. The file is usually kept in the RAYNOISE system (binary) directory. (The path name is installation–dependent and is set by the environment variable WKSFILE, see Section 11.8.3.92 on Page 11–39). The RAYNOISE.WKS file is readable and may be edited (normally only on first installation of the software, or to add new devices). The necessary parameters depend on the device characteristics, but typically include the plot width and height, character sizes and related flags, black–and–white/color switch, and RGB factors (or greyscale levels) for the standard colors such as are used to label groups (sets, materials, ...). These may be adjusted, *eg* to improve the color balance on a particular plotter. Refer to the installation instructions for more information.

12

General display options

This chapter describes the general options which can be applied to displays, to control features such as:

○ *Visibility of objects*

○ *Face painting/wireframe representation*

○ *Numbers of views*

○ *Scaling and zoom*

○ *Numbering and other markers*

○ *Color schemes*

○ *Scales for results presentations*

12.1 Display menu

12.1.1 Purpose

The **Display** menu in the Menubar contains all the commands and dialogs that control the visual appearance of all entities in the graphics window as well as the graphics window layout:

- Color and appearance of objects and groups (materials, sets)
- Color and appearance of Graphical Post–processing plots
- Graphics window layout
- External plot file format

12.1.2 Graphical Options

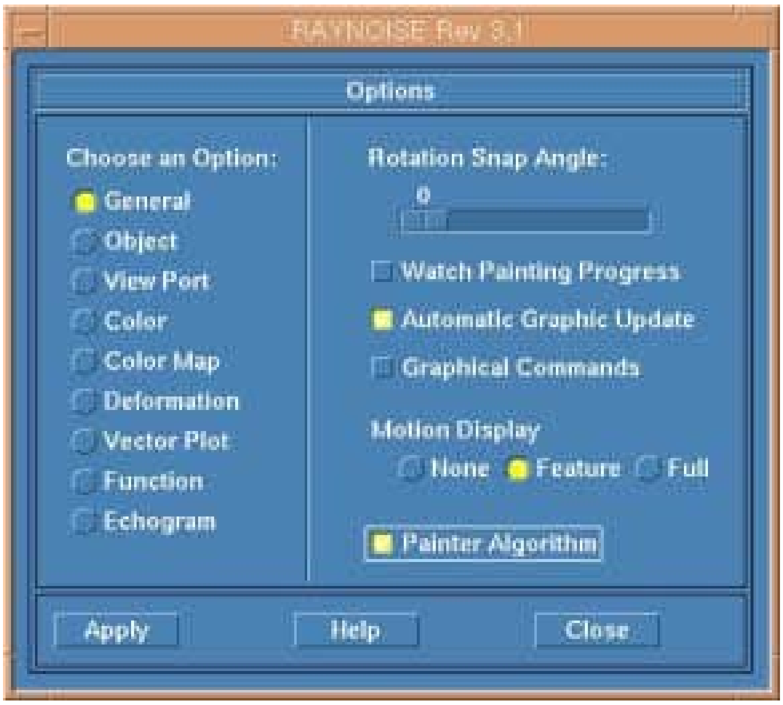
12.1.2.1 Dialog layout

The **Graphical Options...** menu entry opens up a dialog that controls the visual appearance of all RAYNOISE entities either in the entire graphics window or in the active Viewport (see *Changing the number of views*, Section 12.1.12 on Page 12–16), for both the Model Viewer and Graphical Postprocessing.

The *Graphical Options* dialog is a multiple dialog, in which the content of one part of the dialog can depend on the setting of switches in another part of the dialog. The *Graphical Options* dialog consists of two parts:

- The left–hand side of the dialog consists of a radio box containing switches to select the class of graphical display parameters which are to be set or modified
- The right–hand side of the dialog consists of the actual buttons, entry fields and switches for setting the graphical display parameters of the selected class. Its contents depend on the switch settings in the left–hand side of the dialog.

Figure 12.1–1, for instance, shows the appearance of the *Graphical Options* dialog for the switch setting **General**.



Choose an Option:

● General
○ Object
○ View Port
○ Color
○ Color Map
○ Deformation
○ Vector Plot
○ Function
○ Echogram

Rotation Snap Angle:
0

☐ Watch Painting Progress
☑ Automatic Graphic Update
☐ Graphical Commands

Motion Display
○ None ● Feature ○ Full

☑ Painter Algorithm

Figure 12.1–1 Graphical Options dialog for the General switch

The *Graphical Options* dialog contains three buttons at the bottom of the dialog:

- ***OK*** applies the settings to the display and closes the dialog
- ***Update*** applies the settings to the display while the dialog remains opened
- ***Done*** closes the dialog without applying the settings to the display

12.1.3 General switch setting

The switch setting *General* opens up the dialog for display parameter settings that affect the entire graphics window (Figure 12.1–1):

12.1.3.1 Rotation Snap Angle

The slide bar sets the snap angle which governs the increments with which model rotation is carried out: when rotating the model with the mouse (Toolbar button) the final rotation angle will be an exact multiple of the snap angle (expressed in degrees). This precision is not taken into account if the *Watch Rotation Progress* toggle is *On*.

12.1.3.2 Watch Painting Progress

When toggled *On*, the build–up of a Face Painting Rendering (shaded) image can be followed interactively face by face.

When toggled *Off*, the Face Painting Rendering image is displayed only when it has been fully completed.

Note! Turning on *Watch Painting Progress* slows down the display of entities in the graphics window.

12.1.3.3 Watch Rotation Progress

When toggled *On*, **dynamic model rotation** is activated and the rotation of the model proceeds in a smooth, continuous way and the *Rotation Snap Angle* is not taken into account.

When toggled *Off*, dynamic model rotation is de–activated and only the initial and final positions of the model are displayed.

12.1.3.4 Automatic Graphic Update

When toggled *On*, the automatic updating of the graphics window is enabled: the effect of any RAYNOISE command (defining a source, modifying a node coordinate, ...) is immediately reflected in the graphics window.

When toggled *Off*, the automatic updating of the graphics window is disabled.

Hint: It may be desirable to turn OFF automatic graphic updating in some cases, when reading a command file or creating a large number of sets (eg, from AutoCAD layers using the DXF interface). In other cases it is usually useful to have immediate graphic feedback of the modeling actions.

12.1.3.5 Graphical commands

This button controls the inclusion of graphical commands in the Command History List.

When toggled *On*, the execution of any graphical operation simultaneously generates an equivalent RCL command that is added to the Command History List.

When toggled *Off*, graphical operations are executed directly without generating an equivalent RCL command.

12.1.3.6 Painter algorithm

This button controls the quality of the 'hidden line' algorithm which is used in Face Painting.

When toggled *On*, a painter algorithm is used which (in most cases) gives a complete hidden–line view of the model. This can be beneficial when picking items, such as edges to be used for diffraction.

When toggled *Off*, the hidden line approach is simpler (standard algorithm of RAYNOISE Rev 3.0) and may have some inconsistencies, particularly when some large elements or multi–vertex polygons are present in the mesh. However, the refresh speed of the model view is usually faster, which can be beneficial (for example) if rotating a large model.

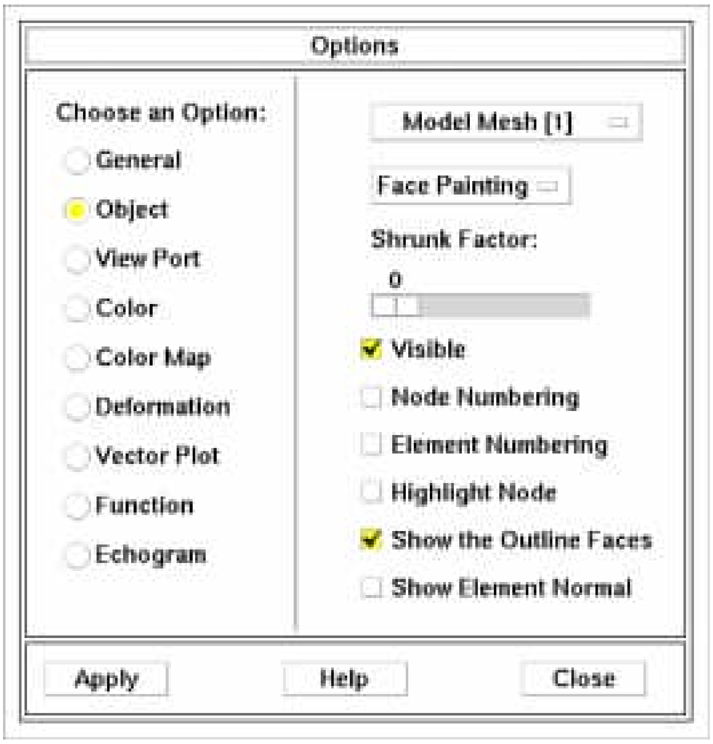
Note! The switches *Rotation Snap Angle, Watch Painting Progress, Watch Rotation Progress, Automatic Graphic Update* and *Painter Algorithm* also affect the local values (within the current RAYNOISE session) of the corresponding graphical environment variables SNAPANGLE, WATCHPAINTING, WATCHROTATION, GRAPHICUPDATE and PAINTERALGO. The initial value of these variables is defined in the RAYNOISE.STP or .PRF file (see *Set–up file*, Section 11.8.4 on Page 11–39, and *Profile file*, Section 11.8.5 on Page 11–40).

12.1.4 Object switch setting

12.1.4.1 Opening the dialog

The option setting **Object** opens up the dialog for display parameter settings that affect the appearance of graphical *Objects* in the currently active Viewport (Figure 12.1–2).

The dialog contains a list of currently available *Objects* (Acoustic Mesh, Field Point Mesh, Sources,...) for which display attributes can be set. The object is selected by highlighting the corresponding entry in the list.



Options

Choose an Option:

○ General
● Object
○ View Port
○ Color
○ Color Map
○ Deformation
○ Vector Plot
○ Function
○ Echogram

Model Mesh [1]

Face Painting

Shrunk Factor: 0

☑ Visible
□ Node Numbering
□ Element Numbering
□ Highlight Node
☑ Show the Outline Faces
□ Show Element Normal

Figure 12.1–2   Graphical Options dialog for the Object switch

Depending on the type of object selected, some of the display parameters are not applicable and the corresponding buttons or switches are made insensitive or ghosted.

12.1.4.2 Corresponding Rendering Method

This controls the choice between two ways of representing an object:

- *Face Painting Rendering* (for shaded image display)
- *Wireframe Rendering* (for line display)

Note that the face–painted/wireframe toggle button on the Toolbar has the same effect, except that it operates simultaneously on all objects in the active viewport.

Hint: To view the field point mesh inside an acoustic mesh, select the wireframe mode for the acoustic mesh and the face–painted mode (or wireframe mode) for the field point mesh.

12.1.4.3 Shrunk Factor

The slide bar controls the element ‘shrink’ factor: depending on the shrink percentage, the edges of the elements will be moved towards the element center of gravity, to give a smaller element appearance.

Shrunk percentage 0 means full–size elements (no shrink). At shrunk percentage 100, the elements will collapse to their centers of gravity and become invisible.

12.1.4.4 Visible

This On/Off toggle switch controls the visibility of the selected object in the graphics window. It corresponds to the de/selection of the object in the **View, Objects...** dialog.

12.1.4.5 Node Numbering

This On/Off toggle switch controls the display of node labels on top of the mesh and is equivalent to the Toolbar button n

12.1.4.6 Element Numbering

This On/Off toggle switch controls the display of element labels on top of the mesh and is equivalent to the Toolbar button e

12.1.4.7 Field–point Numbering

This On/Off toggle switch controls the display of field point numbers on top of the mesh and is equivalent to the Toolbar button p

12.1.4.8 Highlight Node

This On/Off toggle switch controls the display of the position of the nodes by means of small diamond symbols displayed on top of the mesh line intersections.

12.1.4.9 Show the Outline Faces

This On/Off toggle switch controls the display of the element borders. When switched On, all element boundaries are highlighted using a thin line.

12.1.4.10 Show Element Normal

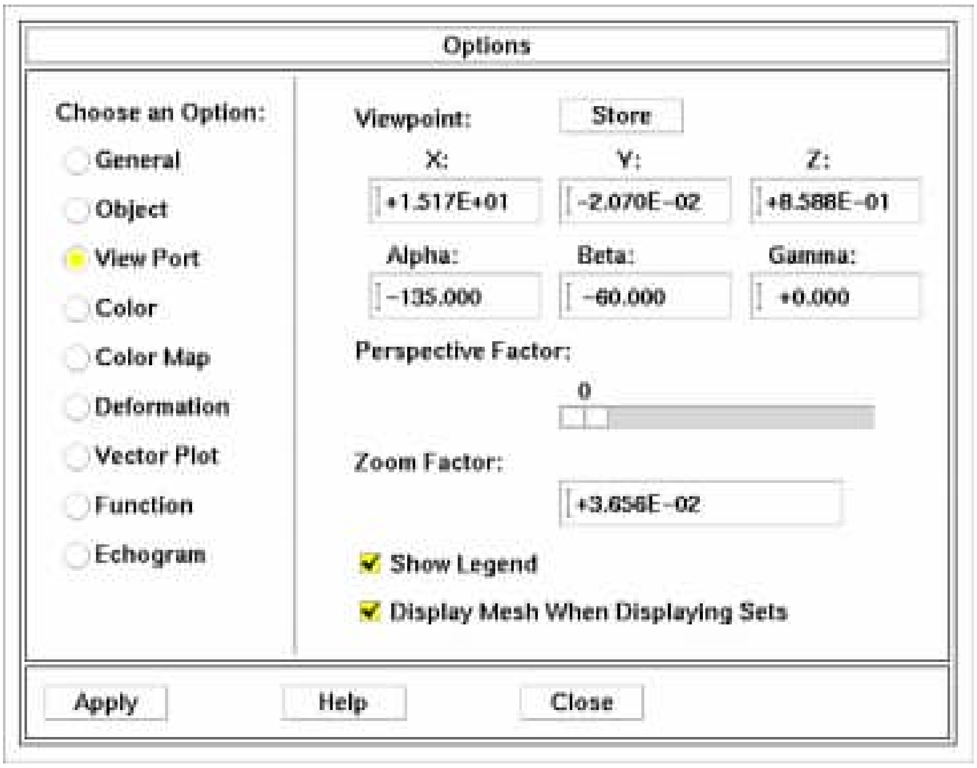
This On/Off toggle switch controls the display of surface element normals. When switched On, all surface element normals are visualized by means of blue arrows at the element center of gravity. See *Reversing element orientations*, Section 6.2.6 on Page 6–7, for information on reversing element normals. Element normal direction may be of relevance if panel sources or transmission properties are being used. The switch is equivalent to the Toolbar button ↑

12.1.5 Viewport switch setting

12.1.5.1 Opening the dialog

The menu switch **Viewport** opens up the dialog for inspecting and setting the current display parameters of the active Viewport (Figure 12.1−3).

The Viewport can be thought of as a ‘window’ through which the user observes the model in the graphics window. There can be up to four independent Viewports (see *Changing the number of views*, Section 12.1.12 on Page 12−16).



○ General
○ Object
● View Port
○ Color
○ Color Map
○ Deformation
○ Vector Plot
○ Function
○ Echogram

Viewpoint:

X: +1.517E+01 Y: -2.070E-02 Z: +8.588E-01

Alpha: -135.000 Beta: -60.000 Gamma: +0.000

Perspective Factor: 0

Zoom Factor: +3.656E-02

☑ Show Legend

☑ Display Mesh When Displaying Sets

Figure 12.1–3 Graphical Options dialog with Viewport switch

12.1.5.2 Viewpoint Angles

The *Alpha, Beta* and *Gamma* fields show the successive Euler angles of rotation of the global X,Y,Z coordinate system of the model with respect to the fixed coordinate system of the Viewport (horizontal X−axis, vertical Y−axis and Z−axis perpendicular to the screen):

- *Alpha* (*Precession angle):* rotation around the Z−axis
- *Beta* (*Nutation angle):* rotation around the new position of the X−axis

- *Gamma (Rotation angle):* rotation around the new position of the Z–axis

Clicking the **Store As User Default** button stores the present Viewport settings for recall through the **Display, View Point, User Default** menu selection. This user default viewpoint remains active for the current session of RAYNOISE, unless a new viewpoint is stored. It also becomes the user's own default User viewpoint, in his Profile file RAYNOISE.PRF, if Environment Variables are saved in the user profile after Storing a specific viewpoint as user default. (See Section 11.8.5 on Page 11–40). Note that no viewpoint information is stored in the model database.

12.1.5.3 Position

The *X, Y, Z* entry fields display the present coordinates of the center of the Viewport with respect to the global X,Y,Z coordinate system of the model.

12.1.5.4 Perspective Factor

The slide bar controls the perspective factor in the Viewport, ranging from 0 (no perspective) to 100 (maximum perspective).

12.1.5.5 Zoom Factor

The entry field displays the present zoom factor of the Viewport: increasing the value is equivalent to zooming in, decreasing the value is equivalent to zooming out. This corresponds with the dynamic zoom button on the Toolbar.

12.1.5.6 Show Group Legend

When toggled *On*, the color legend of all the groups visible in the graphics window is displayed in the upper left corner of the Viewport(s)

12.1.5.7 Display Mesh When Displaying Sets

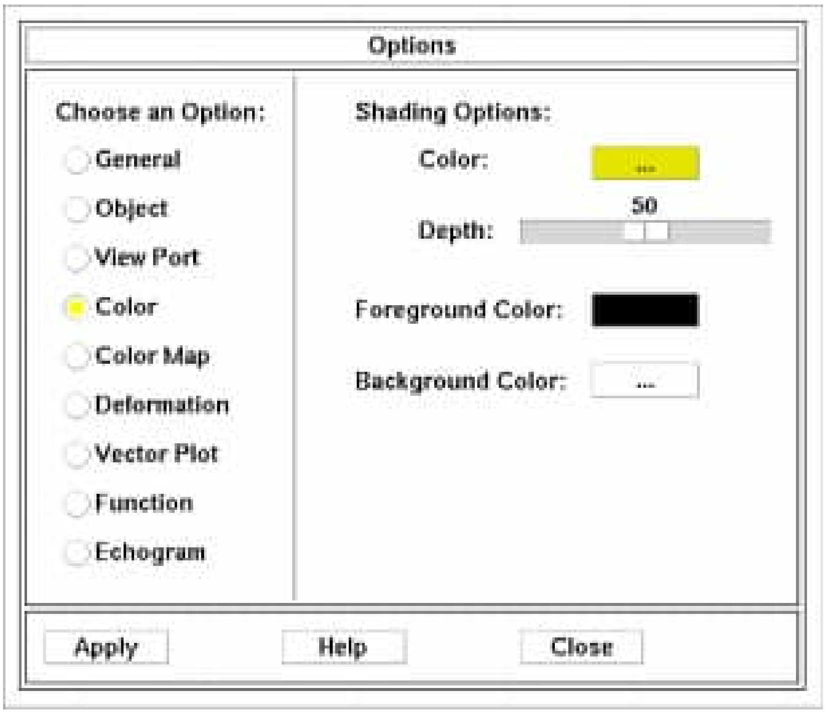
When toggled *On*, Sets are displayed on top of the mesh.

When toggled *Off*, the mesh becomes invisible whenever Sets are displayed. This can be useful, to make partial displays of large or complex models. When no set is selected, the whole mesh is displayed.

12.1.6 Color switch setting

12.1.6.1 Opening the dialog

The switch setting **Color** shows the settings of the general color parameters of the graphics window (Figure 12.1–4).



Options

Choose an Option:

○ General
○ Object
○ View Port
● Color
○ Color Map
○ Deformation
○ Vector Plot
○ Function
○ Echogram

Shading Options:

Color: ...

Depth: 50

Foreground Color: ■

Background Color: ...

Figure 12.1–4   Graphical Options dialog with Color switch

12.1.6.2 Shading Options Color

Defines the color of the mesh for the Face Painting Rendering scheme. Clicking the button with the three points opens a new dialog containing a slide bar tool for combining the basic colors **R**ed, **G**reen and **B**lue. By varying the values of the slider between 0 and 100% for each constituent, a new mesh color is defined. The color is displayed on a palette for immediate visual feedback.

12.1.6.3 Shading Options Depth

The slide bar controls the amount of shading applied to the mesh in the Face Painting Rendering scheme from 0% (no shading – all faces are illuminated equally) to 100% (full shading – face illumination depends on face inclination with respect to the Viewport: parallel to Viewport = complete illumination, perpendicular to Viewport = no illumination)

12.1.6.4 Foreground Color

This button controls the color of text and line drawings displayed in the graphics window. Clicking on the button opens a *Color Selector* dialog (Figure 12.1–5) containing 8 pre–defined colors: radio buttons select the color that will be applied to all text strings and line drawings in the graphics window.

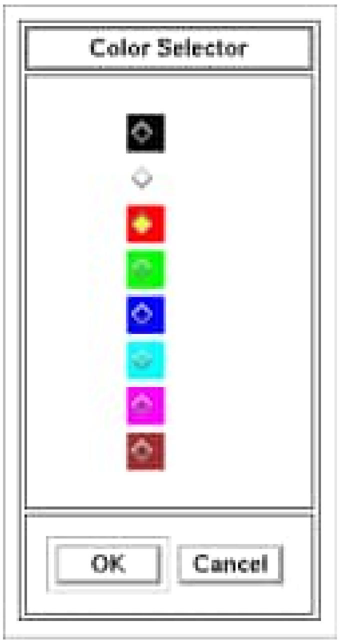


Figure 12.1–5 Color Selector dialog

12.1.6.5 Background Color

This button controls the background color of the graphics window. Clicking on the button opens a *Color Selector* dialog (Figure 12.1–5) containing 8 pre–defined colors: radio buttons select the color that will be applied to the entire background of the graphics window screen.

The 8 pre–defined colors in the Color Selector dialog are controlled by the color entries in the RAYNOISE.WKS file (see Section 11.10 on Page 11–43).

12.1.7 Color Map switch setting

The dialog for the *Color Map* switch is described in detail in *Color representation control*, Section 10.2.1.4 on Page 10–8.

12.1.8 Curve switch setting

The dialog for the *Function* switch is described in detail in *Options for response functions*, Section 10.3.3 on Page 10–30.

12.1.9 Plotting driver

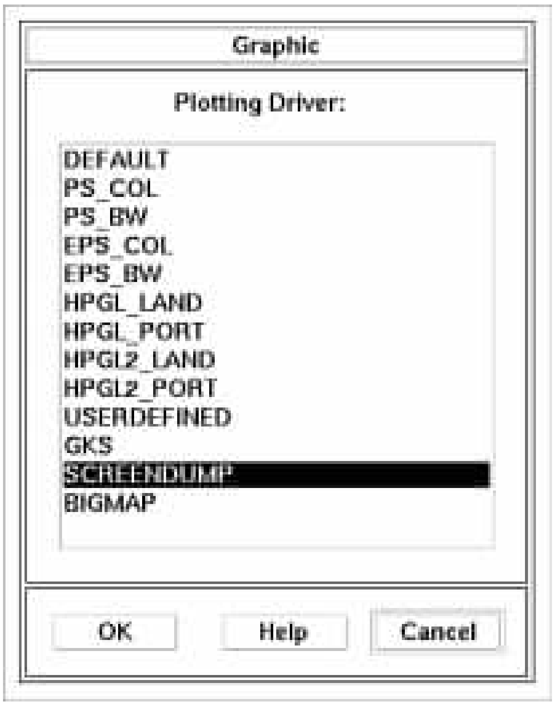


Figure 12.1–6 Plotting driver selection dialog

The **Plotting Driver...** menu entry opens a dialog containing a list of available graphical formats for the creation of external plot files.

When clicking the **Print** button on the Toolbar, the contents of the graphics window will be copied to a plot file in a graphical format corresponding to the highlighted entry in the list above.

By default, the following graphical formats are supported by RAYNOISE:

- Color and B/W PostScript formats (PS_COL, PS_BW)
- Color and B/W Encapsulated PostScript formats (EPS_COL, EPS_BW)
- HPGL Landscape and Portrait formats (HPGL_LAND, HPGL_PORT*)*
- HPGL2 Landscape and Portrait formats (HPGL2_LAND, HPGL2_PORT)
- User–defined format (USERDEFINED)
- GKS format (GKS)

Other formats (*eg* Windows Print Manager on PC) may also be available and are installation–specific.

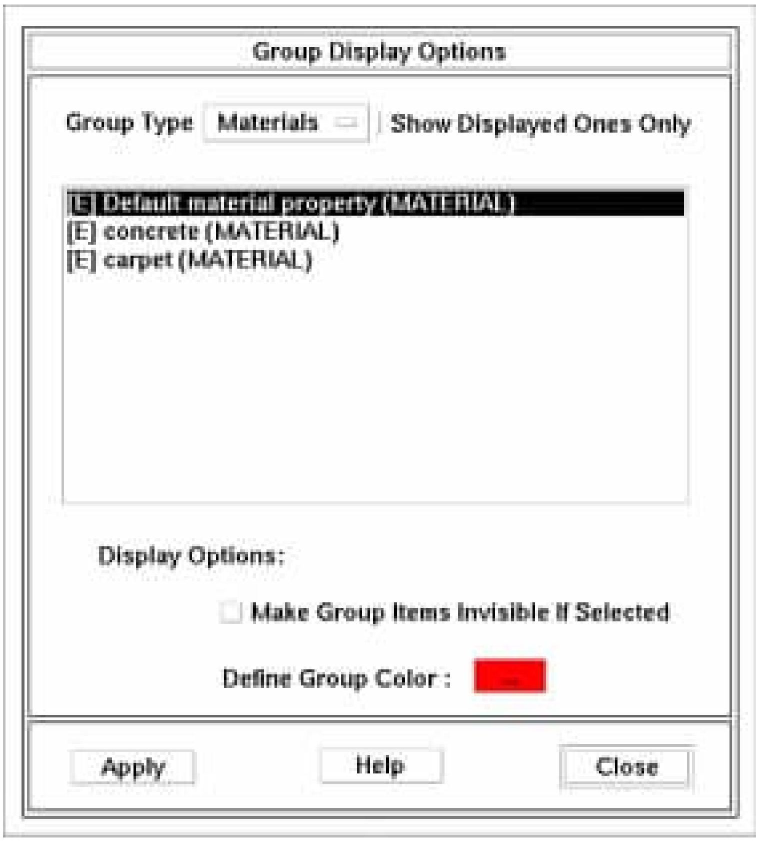
Hint: The Encapsulated PostScript formats are especially useful for the creation of image files for insertion into word–processing documents.

The available graphic drivers are defined in the graphics parameters file RAYNOISE.WKS (see Section 11.10 on Page 11–43).
The initial plotting driver used is defined by the PLDEVICE environment variable in the Set–up file RAYNOISE.STP or Profile file RAYNOISE.PRF (See Section 11.8.5 on Page 11–40).

12.1.10 Group Colors

12.1.10.1 Opening the dialog

The **Group Colors...** menu entry activates a dialog where display colors can be selected for each of the currently–defined groups (Figure 12.1–7).


Group Display Options

Group Type: Materials | Show Displayed Ones Only

[E] Default material property (MATERIAL)
[E] concrete (MATERIAL)
[E] carpet (MATERIAL)

Display Options:

□ Make Group Items Invisible If Selected

Define Group Color : 🟥

Figure 12.1–7 Display, Group Colors... dialog

12.1.10.2 Group Type

This button selects the type of group to be listed in the area below the button: *Sets, Layers* or *Materials.*

12.1.10.3 Show Displayed Ones Only

When toggled *Off*, all the existing groups of the selected type are listed.

When toggled *On*, only those groups of the selected type that are currently selected for display in the *View* menu are listed.

12.1.10.4 Define Group Color

This button controls the assignment of a display color to a group. Select the desired group by highlighting its entry in the list, click on the button, which opens a *Color Selector* sub–dialog containing 8 pre–defined colors, and select a color by clicking the corresponding radio button (Figure 12.1–8).

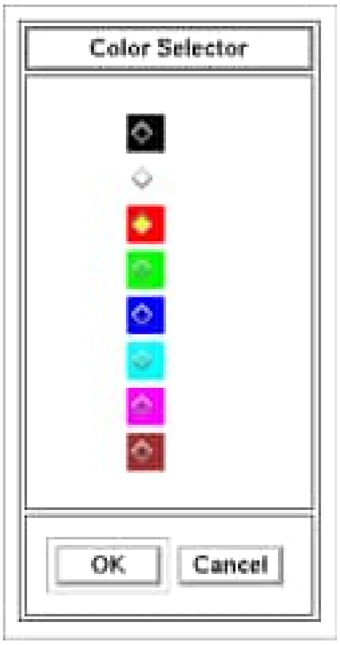


Figure 12.1–8 Color Selector dialog

12.1.10.5 Make Group Items Invisible If Selected

When toggled *On*, the elements contained in the highlighted group entry in the listing area will be removed from the display in the graphics window. This can be useful, for instance, to look inside a closed mesh by removing one side.

12.1.11 View Point

The user can immediately apply several pre–defined settings to the currently active Viewport through the **Display, View Point** menu. The menu provides the following types of pre–defined views with respect to the global X,Y,Z axes of the model:

- 3 Coordinate plane views: *XY, XZ* or *YZ*
- 6 Isometric views: *X on Top, Y on Top, Z on Top* (*Front* and *Rear* positions)
- *User Default:* user−defined view, stored with the *Store As User Default* button in the *Graphical Options, Viewport* dialog (Section 12.1.5 Page 12−9).

The User Default view is also defined by the environment variable VIEWPOINT in the Set−up file RAYNOISE.STP or Profile file RAYNOISE.PRF. (See Section 11.8.5 on Page 11−40).

Hint: Store As User Default only stores the current viewpoint for the current RAYNOISE session, so it is lost on Exit. It is not associated with the model or stored in the database. If you want to store it for use in future sessions, it must be stored in the RAYNOISE.PRF file, via **Tools, Environment Var...**

12.1.12 Changing the number of views

The user can divide the graphics window area into one, two, three or four independent Viewports with the **Display, Change Number of Views...** menu. The lay−out of the multiple Viewports for the different entries is shown in Figure 12.1−9.

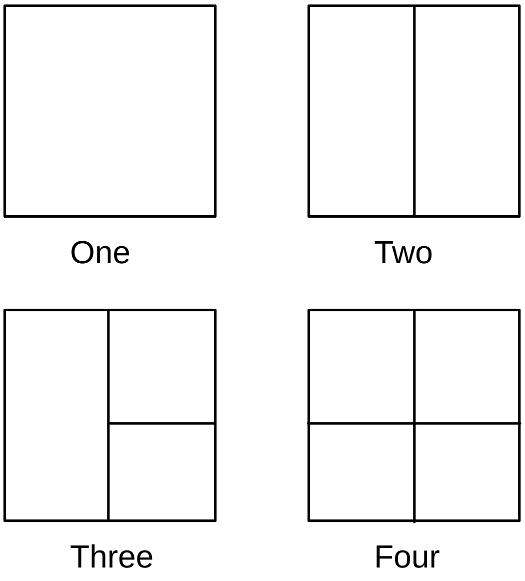


Figure 12.1–9 Multiple Viewports lay–out

Note! At any time, only one of the multiple Viewports is active. Only this Viewport will be affected by GUI commands. Thus, one can display different objects or different open models, or use different visualization methods, in different viewports. Some graphical commands, especially animation, are exceptions to this rule and affect all viewports simultaneously.
The active viewport is identified by a red border.
A Viewport is made active by clicking in its area with the mouse.

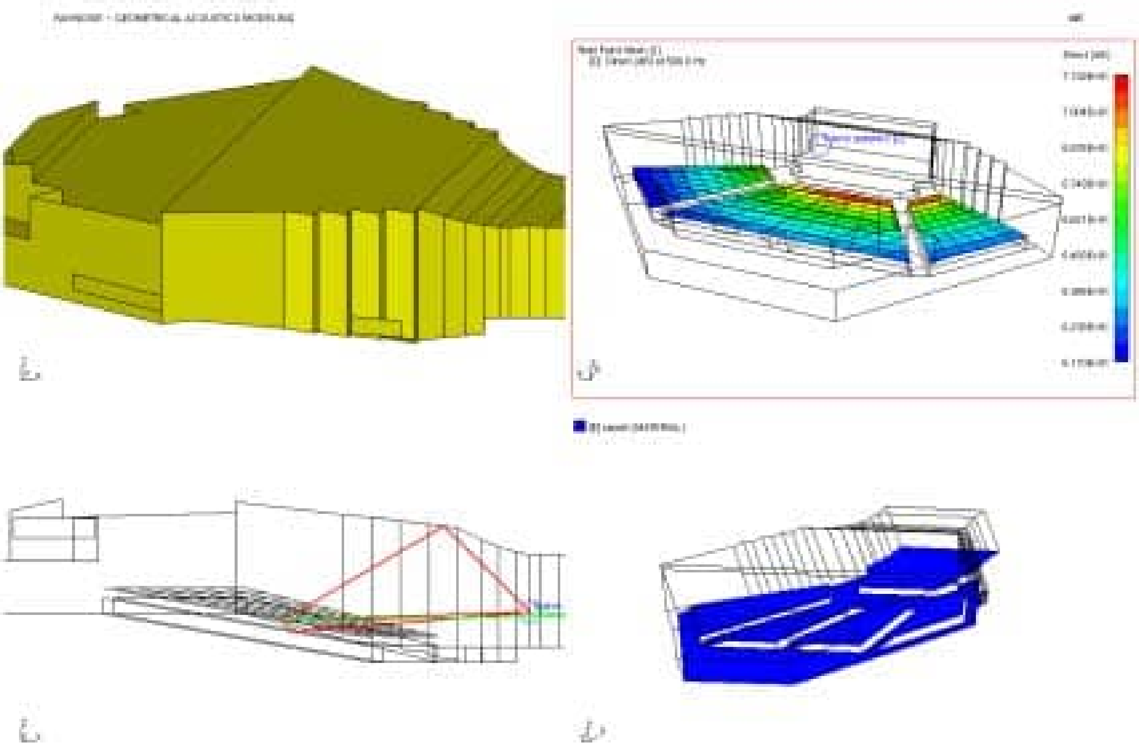


Figure 12.1–10 Example of multiple viewports (top right active)

13

Appendix 1: Acoustic quantities

13.1 Overview of acoustic results

13.1.1 Purpose

This chapter contains the definitions of all the acoustic parameters (results) as they are implemented RAYNOISE.

The parameters fall into five groups, related to different types of applications. (See Types of parameters, Section 13.1.2 on Page 13–3). The definitions are then listed below in alphabetical order.

Not all parameters may be available in a particular RAYNOISE model, depending on the program modules which have been licensed and the data storage level selected during the analysis processes. (See Available parameters, Section 13.1.3 on Page 13–3).

13.1.2 Types of parameters

The **Pressure** group contains the results for pressures as physical quantities (*eg* in MKS units) and SPL in dB (Linear).

The **SPL** group contains the results for stationary sound: Sound Pressure Level (SPL), Noise Criterion (NC), Noise Rating (NR), Direct energy and Total–to–Direct energy ratio.

The **Speech** group contains the results for Speech Transmission Index (STI), Definition, Echo Criterion (EC) for Speech and Early Reflection Ratio (ERR).

The **Music** group contains the results for Clarity, Lateral Efficiency (LE), Central Gravity Time (TCG) and Echo Criterion (EC) for Music.

The **Reverberation** group contains the results for Reverberation Times based on different decay ratios (RT_{60}, RT_{30}, RT_{20}) and Early Decay Time (EDT).

13.1.3 Available parameters

13.1.3.1 Effect of licensed modules

If only the Stationary Sound module of RAYNOISE is installed, the available results are limited to the acoustic parameters of the SPL group (see *Types of parameters*, Section 13.1.2 on Page 13–3) together with the position–independent

Reverberation Time. If the Transient module is also installed, all results can be available, provided they have been calculated and stored. The Auralization module has no influence on the available acoustic results.

13.1.3.2 Effect of storage level

Whatever the modules which are installed, the Environment Variable STORELEVEL controls the amount of results storage, by limiting the parameters which are saved.

STORELEVEL = 0 stores only the Stationary Sound results (*ie* the SPL group and the position–independent Reverberation Times, see *Types of parameters*, Section 13.1.2 on Page 13–3).

STORELEVEL = 1 stores all groups of results. STI data are approximated.

STORELEVEL = 2 also stores all groups of results. STI data are 'exact' and use detailed data from Modulation Transfer Functions.

The choice of STORELEVEL affects the computer resources used, especially the database file size.

- See *Types of parameters*, Section 13.1.2 on Page 13–3, for more details.

- See *Histogram parameters*, Section 9.5.4 on Page 9–15, for details on STORELEVEL.

- See *Extracting information, Inquire menu*, Section 4.8.1 on Page 4–17, for how to list results.

13.2 Definitions of acoustic quantities

13.2.1 Central Gravity Time

Central gravity time (TCG) is another criterion which is based on the ratio between 'early'– and 'late'–arriving energy, but using a 'smooth' function. It is the first moment of the area under the decay curve, defined as:

$$TCG = \frac{\int_0^\infty t.p^2(t)dt}{\int_0^\infty p^2(t)dt} \qquad (ms)$$

Eqn 13.2–1

References

- Kürer, R., *Acustica* 21 (1969) p.370.

13.2.2 Clarity

The clarity index uses a value of 80ms for the limit of perceptibility for music and is defined as:

$$C = 10\log\left(\frac{\int_0^{80ms} p^2dt}{\int_{80ms}^\infty p^2dt}\right) \qquad (dB)$$

Eqn 13.2–2

References

- Reichardt,W, *et al*, *Acustica* 32 (1975) p.126.

13.2.3 Definition

The Definition criterion is a measure for speech intelligibility. It compares the 'useful' sound with the total sound. 'Useful' sound reflections are those which arrive at the listener's position within 50 ms after the direct sound:

$$D = \frac{\int_0^{50ms} p^2 dt}{\int_0^{\infty} p^2 dt} \qquad (x100\%) \qquad \text{Eqn 13.2-3}$$

References

- Thiele,R., *Acustica* 3 (1953) p.291.

13.2.4 Direct Energy

The direct energy (SPL_{direct}) is the Sound Pressure Level (see Section 13.2.11 on Page 13–8) without taking into account any reflections.

13.2.5 Early Reflection Ratio

This ratio (ERR) is given by the early sound energy (up to 50ms) divided by the (theoretical) direct sound energy. With direct sound alone the ratio equals one. With direct sound and an equally–strong early reflection, the ratio is two.

References

- Barron M., *Auditorium acoustics and architectural design* E & FN Spon (1993) p. 236.

13.2.6 Echo Criterion of Dietsch and Kraak

The Echo Criterion of Dietsch and Kraak is based on the ratio:

$$t_s(\tau) = \frac{\int_0^{\tau} t.p^n(t)dt}{\int_0^{\tau} p^n(t)dt} \qquad (ms) \qquad \text{Eqn 13.2-4}$$

The quantity used for rating the strength of an echo is based upon the running difference quotient of $t_s(\tau)$:

$$EC = \max \frac{\Delta t_s}{\Delta \tau}$$ *Eqn 13.2–5*

A critical value EC_{crit} must not be exceeded, to ensure that not more than 50% of the listeners will hear an echo. Suitable values for the exponent *n*, Δτ and EC_{crit} depend on the application, *ie* speech or music:

	n	Δτ	EC_{crit}
Speech	2/3	9	1.0
Music	1	14	1.8

References

- Dietsch, L. & Kraak, W., *Acustica* 60 (1986) p. 205.

13.2.7 Lateral Efficiency

Lateral Efficiency is a measure for the impression of spaciousness, for instance in a concert hall. It is based on the ratio of laterally–arriving reflections to total reflections:

$$LE = \frac{\int_{25ms}^{80ms} p^2 \cos^2(\theta) dt}{\int_{80ms}^{\infty} p^2 dt} \qquad (x100\%)$$ *Eqn 13.2–6*

The angle θ is the angle between the arriving reflections and the "ear–to–ear" axis of a horizontal head at the receiver point, looking towards the source.

References

- Jordan,V., *Acoustical Design of Concert Halls and Theaters*, ASP Ltd., London, 1980, pp.159 and 191.

13.2.8 Noise Criterion (NC)

Beranek developed the Noise Criterion (NC) to deal with commercial buildings. The criterion consists of a family of curves which relates the noise spectrum to the disturbance of speech communication. The NC number of a noise spectrum can be determined by superimposing the noise spectrum in octave band levels on the NC curves. The lowest curve which is not exceeded in any of the octave bands, is the NC rating of the particular noise.

References

- Beranek, 1957

13.2.9 Noise Rating (NR)

The Noise Rating method was developed from the NC curves for wider application. NR numbers are widely−used in Europe.

References

- Maekawa Z. and Lord P., *Environmental and architectural acoustics* E & FN Spon, p. 47−48.

13.2.10 Reverberation Times

Reverberation Time definitions come in many variants. RAYNOISE calculates RT_{60}, RT_{30}, RT_{20} and RT_{10} (or EDT), where RTx is the time needed to obtain a reduction in SPL of xdB, after source cut off, multiplied by $60/x$. RT_{10} is the same as EDT (Early Decay Time). All these decay rates are taken from the Schroeder integration curve, defined as:

$$s^2(t) = \int_t^\infty p^2(\tau)d\tau \qquad \text{Eqn 13.2−7}$$

13.2.11 Sound Pressure Level

13.2.11.1 General

The Sound Pressure Level (*SPL*) is defined as:

$$SPL = 10\log_{10}\frac{p_{eff}^2}{p_0^2} \qquad \text{Eqn 13.2−8}$$

where: p_{eff} is the effective pressure and p_0 the reference pressure.

$p_0 = 2x10^{-5}$ (N/m2)

The effective pressure is the result of the superposition of all reflections at the point of interest.

See *Reference values*, Section 11.5 on Page 11–15, for details on reference pressure and how to change it.

13.2.11.2 Coherent and incoherent sources

Depending on whether the source is incoherent or coherent, we obtain the effective pressure by:

$$p_{eff}^2 = \sum_i p_i^2 \qquad (incoherent) \qquad \text{Eqn 13.2–9}$$

$$p_{eff}^2 = \left\| \sum_i p_i^c(\theta_i) \right\| \qquad (coherent) \qquad \text{Eqn 13.2–10}$$

and:

$$p_i^2 = \frac{W.\rho c}{4\pi.r_i^2} \sum_j (1 - a_j) \qquad \text{Eqn 13.2–11}$$

where: W = Sound Power Level (watts)

r_i = distance traveled by the wavefront i (metre)

a_j = absorption coefficient of surface j

${p_i}^c$ = complex pressure due to wavefront i

θ_i = phase of wave front i.

13.2.12 Speech Transmission Index

The Speech Transmission Index (STI) is an objective criterion used to characterize speech intelligibility. The loss of intelligibility of a speech signal is mainly related to reverberation, interference and background noise. Speech is an auditive flux of which the spectrum varies continuously in time. The individual sinusoidal components of the envelope of the speech signal have to be preserved as clearly as possible. The Modulation Transfer Function (MTF) is a response curve, expressing the attenuation of this envelope by the room (or the environment in general). The reverberant transmission from the source and from other coherent sources (such as linked loudspeakers) and the degradation due to background noise are all taken into account.

On the assumption that the reverberation process follows a pure exponential decay, the MTF is given by:

$$MTF(F) = \left[1 + \left(2\pi F \frac{T}{13.8}\right)^2\right]^{-1/2} . \left[1 + 10^{-SNR/10}\right]^{-1/2} \qquad \textit{Eqn 13.2-12}$$

where T = reverberation time (sec)

SNR = background noise (dB)

F = modulation frequency (Hz)

In its most general form, the MTF is the normalized Fourier Transform of the impulse response in the frequency range 0.25 to 32 Hz:

$$MTF(F) = \frac{\left| \int_0^\infty e^{2\pi jFt} . p^2(t)dt \right|}{\int_0^\infty p^2(t)dt} . \left[1 + 10^{-SNR/10}\right]^{-1/2} \qquad \textit{Eqn 13.2-13}$$

The STI is an averaged and normalized value of the MTF.

With the calculation parameter (Environment Variable) STORELEVEL = 1, RAYNOISE uses the approximate formula (Eqn 13.2–12) to calculate the STI. When STORELEVEL = 2, the Fourier Transform approach (Eqn 13.2–13) is taken. See *Histogram parameters*, Section 9.5.4 on Page 9–15, for more details of how to apply this.

RaSTI (which may relate to more–easily measured data) is the average of the STI–values at 500Hz and 2000Hz.

References

- Houtgast, T. *et al*, Predicting Speech Intelligibility in Rooms from the Modulation Transfer Function. Part I, *Acustica* 16(1980) p. 60.

- Plomp, R. *et al*, Predicting Speech Intelligibility in Rooms from the Modulation Transfer Function. Part II, *Acustica* 46(1980) p. 75.

13.2.13 Total/Direct Energy Ratio

The Total/Direct Energy Ratio equals the Sound Pressure Level (see Section 13.2.11 on Page 13–8) minus the Direct energy (see Section 13.2.4 on Page 13–6), in dB:

$$SPL_{total} - SPL_{direct}$$

14

Appendix 2: DXF interface

This chapter explains the operation of the DXF interface and the entities supported by it:

- ○ *Drawing eXchange File (DXF) format*
- ○ *Supported AutoCAD drawing entities*
- ○ *Scanning of the DXF file*
- ○ *Supported DXF file entities and sections in the file*
- ○ *Importing DXF entities into RAYNOISE*
- ○ *Exporting RAYNOISE geometry in DXF format*
- ○ *Exporting RAYNOISE sets in DXF format*

14.1 Introduction

RAYNOISE comes with a two–way interface which works with any existing computer–aided design package that supports the DXF (Drawing eXchange File) format (see Section 14.2 on Page 14–4). Importing models created in a CAD–program into RAYNOISE can save a lot of time and effort, especially when complex models or a set of variants are needed.

The AutoCAD program from Autodesk Inc. is widely used in the architectural design world. All the AutoCAD Drawing Entities which are supported in RAYNOISE are described in *Supported AutoCAD drawing entities*, Section 14.2.2 on Page 14–4.

AutoCAD is just one example of many CAD–programs that can be used to create models for transfer to RAYNOISE. In principle, all CAD–packages that are able to export data in DXF format are suitable.

14.2 Drawing eXchange File (DXF) format

14.2.1 General description of DXF files

In order to assist in interchanging drawings between AutoCAD and other programs, Autodesk has defined a Drawing Exchange File (DXF) format. A full description of the DXF format can be found, for instance, in the AutoCAD Customization Manual (Chapter 11) of release 13. In AutoCAD, a DXF file can be produced with the command **DXFOUT**, while **DXFIN** imports DXF files. A DXF file is an ASCII formatted file and has generally the extension *.dxf*.

A DXF file always includes the following sections : HEADER, TABLES, BLOCKS, ENTITIES and END OF FILE. Each section is composed of many groups, each of which occupies two lines. The first line of a group is the group code, which is a positive non–zero integer. The second line of the group is the group value.

The ENTITIES section contains drawing entities, such as LINE, POINT, CIRCLE,... . These entities are described by a group that introduces the item, giving its type and/or name, followed by multiple groups that supply the values associated with the item.

Not all entities are supported by RAYNOISE. Only those that can be interpreted as part of a RAYNOISE model geometry are considered. For instance CIRCLE entities are skipped when reading the DXF file, but 3DFACE entities are retained. The following sections describe in detail all supported AutoCAD Drawing Entities, as well as their equivalent DXF Entities, in detail.

14.2.2 Supported AutoCAD drawing entities

14.2.2.1 3D Faces

3D Faces are quadrilateral or triangular 3D objects. In *AutoCAD* they are created by the command **3DFACE**. If only three points are specified, the third and fourth points will be the same. 3D faces can be combined to model complex three–dimensional surfaces.

Example:

```
Command : 3dface
First point :     7.6,      -12.6,          0.2
Second point :    14.0,     -19.1,          0.2
Third point :     23.4,     -9.7,           -2.2
Fourth point :    20.8,     -7.0,           -2.2
```

14.2.2.2 Solids

Solids are quadrilateral or triangular 2D objects. In AutoCAD they are created by the command **SOLID**. If only three points are specified, the third and fourth points will be the same.

14.2.2.3 Polyline

A Polyline is a general, two– or three–dimensional polyline consisting of straight–line segments with zero width. The user must supply 2D (x,y) or 3D (x,y,z) coordinates for each vertex. Polylines can be open or closed. Arc segments are not supported. In AutoCAD, Polylines can be created with the following commands:

- **3DPOLY** creates a general 3D Polyline
- **POLYGON** creates a regular 2D Polygon with 3 to 1024 sides
- **PLINE** creates a general 2D Polyline.

14.2.2.4 3D Polygon mesh

A 3D Polygon mesh is a mesh defined in terms of a matrix of *M* x *N* vertices. 3D Polygon meshes can be open or closed. In AutoCAD, 3D Polygon meshes can be created by various commands:

- **3DMESH** creates a topologically–rectangular polygon mesh. The user has to specify its size in terms of M and N, as well as the location of each vertex in the mesh.
- **RULESURF** creates a polygon mesh representing the ruled surface between two curves (not recommended for models to be used in RAYNOISE).
- **TABSURF** creates a polygon mesh representing a general tabulated surface defined by a path curve and a direction vector.
- **REVSURF** creates a surface of revolution by rotating a path curve around a selected axis.
- **EDGESURF** creates a Coons surface patch from four adjoining edges (not recommended for models to be used in RAYNOISE).

Example:

```
Command : 3dmesh
Mesh M size : 4
Mesh N size : 3
Vertex (0,0) : 50,40,3
Vertex (0,1) : 50,45,5
Vertex (0,2) : 50,50,3
Vertex (1,0) : 55,40,3
Vertex (1,1) : 55,45,5
Vertex (1,2) : 55,50,3
Vertex (2,0) : 60,40,0
Vertex (2,1) : 60,45,-1
Vertex (2,2) : 60,50,0
Vertex (3,0) : 65,40,0
Vertex (3,1) : 65,45,0
Vertex (3,2) : 65,50,0
```

14.2.2.5 Polyface mesh

A Polyface mesh is a general polygon mesh, defined by vertices and faces composed of those vertices. The location of each vertex in the mesh is specified as well as the vertices for each face. The concept of polyface meshes resembles very much the way geometries are modeled in RAYNOISE.

Example:

```
Command : pface
Vertex 1 : 1.5,  6.5
Vertex 2 : 1.0,  2.5
Vertex 3 : 3.0,  2.5
Vertex 4 : 3.0,  5.0
Vertex 5 : 7.5,  5.5
Vertex 6 : 7.5,  5.5,  3.5
Vertex 7 : 4.5,  7.0,  3.0
Vertex 8 : 2.5,  8.5,  2.0
Vertex 9 : return
Face 1, vertex 1 : 5
Face 1, vertex 2 : 6
Face 1, vertex 3 : 7
Face 1, vertex 4 : 8
Face 1, vertex 6 : return
Face 2, vertex 1 : 5
Face 2, vertex 2 : 4
Face 2, vertex 3 : 3
Face 2, vertex 4 : 2
Face 2, vertex 5 : 1
Face 2, vertex 6 : return
Face 3, vertex 1 : return
```

14.2.2.6 3D Object

The *3d.lsp* AutoLISP application creates various 3D objects, including a box, cone, dome/dish, pyramid, torus, wedge, and a simple 3D mesh. When using the **3D** command, each object is created as a polygon mesh.

14.2.3 Scanning of the DXF file

14.2.3.1 DXF sections read by RAYNOISE

RAYNOISE only reads the TABLES section and the ENTITIES section of the DXF file. The TABLES section contains the layer definitions. Each layer will be interpreted as a set of elements. (See Importing sets, Section on).

14.2.3.2 Coordinate systems

If *AutoCAD* is in Paper Space when you dump the model to a DXF file, all entities will be read by RAYNOISE as if they were defined in model space. Note also that RAYNOISE ignores Viewport information since it is trying to transfer the model, rather than the correct ”drawing”.

14.2.3.3 Blocks

The BLOCKS section of the DXF file is skipped. If you have blocked data, you should use the EXPLODE command in AutoCAD, prior to writing the DXF file for RAYNOISE. You may have to use EXPLODE several times if you have nested BLOCKS.

14.2.4 Supported DXF file entities

14.2.4.1 DXF file section

The following entities in the ENTITIES section of a DXF file are supported when importing a model generated in a CAD program into RAYNOISE:

All older DXF entities are backwards−compatible with previous releases of AutoCAD.

14.2.4.2 3DFACE

Group codes:

10,20,30	
11,21,31	
12,22,32	
13,23,33	Corner Point Coordinates
8	Layer name

14.2.4.3 SOLID

Group codes:

10,20,30	
11,21,31	
12,22,32	
13,23,33	Corner Point Coordinates
8	Layer name

14.2.4.4 POLYLINE

Group codes:

66	Vertices–follow flag
70	Polyline flag
71,72	Polygon mesh M and N vertex counts
8	Layer name

Depending on the group value of the Polyline Flag group, this entity can have several interpretations. The Polyline Flag is a bit–coded field. Supported bits are listed in Table 14.2–1 below.

Flag bit value	Meaning
1	closed polyline or polygon mesh closed in M direction
8	3D polyline
16	3D polygon mesh
32	polygon mesh closed in N direction
64	polyface mesh

Table 14.2–1 Polyline flags

Although the number of vertices per face cannot exceed 4, one can represent more complex polygons by decomposing them into triangular wedges. Their edges should be made invisible to prevent visible artifacts of this subdivision from being drawn. The PFACE command in AutoCAD performs this subdivision automatically.

RAYNOISE recognizes these subdivisions and composes them back to the original polygon, as long as the number of vertices does not exceed 20. When RAYNOISE creates a DXF file, a polyface mesh is created whose faces are decomposed into partly visible triangular wedges, in the same way that the command PFACE performs the subdivision.

14.2.4.5 POLYLINE VERTICES

Group codes :

10,20,30	Point coordinates
70	Vertex flag
71,72,73,74	Vertex Indices for Polyface mesh12,22,32
8	Layer name

Depending on the group value of the Vertex Flag group, this entity can have several interpretations. The Vertex Flag is a bit–coded field. The supported bits are listed in the table below.

Flag bit value	Meaning
32	3D polyline vertex
64	3D polygon mesh vertex
128	polyface mesh vertex

Table 14.2–2 Polyline vertex flags

The vertex indices defining the mesh are given by code 71, 72, 73, and 74 groups, whose values are integers specifying one of the previously–defined vertices by index. If the index is negative, the edge that begins with that vertex is invisible.

14.2.5 Importing DXF entities into RAYNOISE

14.2.5.1 Dialog or command

Importing a DXF file into RAYNOISE is done with the **File, Import** dialog or the command INPUT (see *Import and Export data*, *Mesh*, Section 5.3.5 on Page 5–11). The interface format must be selected in the dialog or specified with the command complement FORMAT DXF.

For example, the command sentence:

```
RAYNOISE> INPUT MESH FORMAT DXF FILE MODEL.dxf
```

reads a file called MODEL.dxf, containing a full model description in DXF format.

14.2.5.2 Interpretation of data

A **3D Face** is imported into RAYNOISE as 1 additional *element* and 4 additional *nodes*.

A **Solid** is imported into RAYNOISE as 1 additional *element* and 4 additional *nodes*.

A **Polyline** (closed or open) is imported into RAYNOISE as 1 additional *element* and a number of additional *nodes*, equal to the number of vertices in the Polyline, provided that the number of vertices does not exceed 20.

A **Polygon mesh** is imported into RAYNOISE as MxN additional *elements* and (M+1)x(N+1) additional *nodes*.

A **Polyface mesh** is imported into RAYNOISE as a number of additional *elements* equal to the number of faces in the mesh and a number of additional *nodes* equal to the number of vertices in the mesh.

14.2.5.3 Mesh verification

Once all the entities have been imported into RAYNOISE, the list of nodes is checked for coincidence. All coincident nodes are merged into one single node. A tolerance equal to the TOLERANCE value times the maximal model dimension is used. Isolated nodes, *ie* those not occurring in any of the element connectivity lists, are detected.

Elements with a connectivity list containing a sequence with the same node are degenerated. For instance: 23 : 21,5,5,42,44 is transformed into 23 : 21,5,42,44.

Once the mesh is cleaned up, RAYNOISE checks the coplanarity of the nodes of each element. This is done by comparing the normal vectors of the mathematical planes through every sequence of 3 nodes on the element. If any such 3−node sequence is colinear, RAYNOISE will bypass the middle node and continue.

For each violation of the coplanarity rule a warning message is given, specifying the element and its nodal coordinates.

14.2.5.4 Material assignment using layers

It is very advantageous to use **layers** in AutoCAD. Every surface in the AutoCAD model can be put into a layer named after its associated material.

The layer name of each entity can be imported into RAYNOISE and interpreted as a set of elements. This makes the assignment of the material types to appropriate elements much easier.

The import of layers from a DXF file should be done after importing the corresponding mesh geometry from the same (or compatible) file. In a second step, use the **File, Import, Sets** dialog or the corresponding INPUT command (see *Importing sets of elements from AutoCAD layers*, Section 5.3.7.2 on Page 5−14).

Command example:

```
RAYNOISE> INPUT MESH FORMAT DXF FILE LAYERS.dxf
RAYNOISE> INPUT SET FORMAT DXF FILE LAYERS.dxf
RAYNOISE> EXTRACT SET

 SET INFORMATION
-----------------------------------------------------------------
 NUMBER    SET_ID    NBR ITEMS    TYPE     NAME
-----------------------------------------------------------------
  1  1             6           ELEMENTS        roof
  2  2             16          ELEMENTS        sidewalls
  3  3             12          ELEMENTS        marble floor
  4  4             14          ELEMENTS        wooden floor
  5  5             4           ELEMENTS        door
-----------------------------------------------------------------
 NUMBER OF DEFINED SETS :          5
-----------------------------------------------------------------
```

```
RAYNOISE> MATERIAL 1 SELECT 20 RETURN
RAYNOISE> MATERIAL 2 SELECT 21 RETURN
RAYNOISE> MATERIAL 3 SELECT 22 RETURN
RAYNOISE> MATERIAL 4 SELECT 23 RETURN
RAYNOISE> MATERIAL 5 SELECT 24 RETURN

RAYNOISE> ASSIGN MATERIAL 1 SET 1 RETURN
RAYNOISE> ASSIGN MATERIAL 2 SET 2 RETURN
RAYNOISE> ASSIGN MATERIAL 3 SET 3 RETURN
RAYNOISE> ASSIGN MATERIAL 5 SET 5 RETURN
RAYNOISE>
```

14.2.6 Exporting RAYNOISE geometry in DXF format

14.2.6.1 Dialog and command

RAYNOISE geometry is exported to a DXF−formatted file with the **File, Export** dialog (see *Import and Export of a mesh*, Section 5.3.5 on Page 5−11) or the command OUTPUT. The interface format is specified by selection in the dialog or with the command complement FORMAT DXF.

For example, the command sentence:

```
RAYNOISE> OUTPUT MESH FORMAT DXF FILE MODEL.dxf
```

writes a file called MODEL.dxf, containing the full model geometry in DXF format.

14.2.6.2 Interpretation/translation of data

The RAYNOISE geometry description (nodes, elements) is transformed into a Polyface mesh. Every element consisting of more than 4 nodes is decomposed into partly visible triangular wedges. A number of layers are created equal to the number of material types in RAYNOISE and named according to the material number.

14.2.7 Exporting RAYNOISE sets in DXF format

14.2.7.1 Dialog and command

Sets can be exported to a DXF file, using the **File, Export** dialog (see *Import and Export of sets*, Section 5.3.7 on Page 5–14) or the command OUTPUT. For example, the command sentence:

```
RAYNOISE> OUTPUT SET 7 FORMAT DXF FILE SET7.DXF
```

writes a file SET7.DXF containing the geometry data of set 7.

14.2.7.2 Interpretation/translation of data

If the set to be exported contains Nodes (or Field Points) the resulting DXF file contains just point information.

If the set to be exported contains Elements (or Faces) the RAYNOISE geometry of the set is transformed into a Polyface mesh. Every element consisting of more than 4 nodes is decomposed into partly visible triangular wedges.

15

Appendix 3: Free format files

This chapter describes the RAYNOISE Free format for ASCII file data exchange, for:

○ *Mesh type data*

○ *Results data*

○ *Tables*

○ *Complete data (not documented here)*

15.1 The RAYNOISE Free format interface

15.1.1 General principles

RAYNOISE 3.1 provides the user with a complete two−way (INPUT and OUTPUT) interface with formatted data or results files, known as the RAYNOISE Free format. The files can be used to exchange data with other user programs, to communicate information to other computer systems or simply to have a readable copy of some files.

This interface concerns several kinds of files:

- DATA files
- MESH files
- RESULTS files
- TABLE files.

Each of these files is described in the following sections. In order to obtain more information about the external use of these files, the user should consult LMS or his RAYNOISE local Distributor.

A RAYNOISE Free format file is created by a command of the form:

```
RAYNOISE> INPUT kind_of_file FILE filename FORMAT FREE
```

A RAYNOISE Free format file is read by a command of the form:

```
RAYNOISE> OUTPUT kind_of_file FILE filename FORMAT FREE
```

Note! The reader should not confuse the RAYNOISE Free format file with the FORTRAN free format feature, which corresponds to FORTRAN statements such as:

```
WRITE(LU, * ) OUTPUTLIST
```

or

```
READ (LU, * ) INPUTLIST
```

15.1.2 Free format data files

It is possible to dump the complete data for the model into a Free format file. The definition of this file is very extensive: please contact LMS (*via* your local support representative, if applicable) if you want to request this information. For data transfers between computers (*eg*, PC to Unix...) it is normally recommended to use the **File, DatabaseTo/From...** commands. (See Section 5.3.3 on Page 5−10).

15.1.3 Free format mesh files

15.1.3.1 File contents

MESH information consists of:

- node numbers and Cartesian coordinates
- element numbers, types and connectivity tables.

The mesh data are known after the Import of a mesh into RAYNOISE from an external file (see *Import and Export of a mesh*, Section 5.3.5 on Page 5–11) or the creation of mesh data using *Mesh editing* techniques (see Section 6.2 on Page 6–5). Sets can also be exported as meshes, if they contain elements (or faces). The Field Point Mesh can also be exported as a mesh.

The data are read/written in a RAYNOISE Free format MESH file in three sections.

15.1.3.2 First section: file header

The file header section contains 5 records:

Record #	Format	Content
1:	(A80)	FTYPE
2:	(A80)	REVIS
3:	(A80)	TITLE
4:	(A80)	CDATE
5:	(2I10)	NPOIN,NELEM,NNOEL

with:

FTYPE . = file type (CHARACTER*80; FTYPE='SYSNOISE MESH FILE');

REVIS . = software revision number and release date (CHARACTER*80);

TITLE . = model title defined by the command TITLE (CHARACTER*80);

CDATE . = file creation date and time (CHARACTER*80);

NPOIN . = the number of nodes in the mesh (INTEGER*4);

NELEM = the number of elements in the mesh (INTEGER*4)

NNOEL = the maximum number of nodes per element (INTEGER*4).

15.1.3.3 Second section: Node information

NPOIN records, each describing one node:

Format	Content
(2I10,3E20.8)	IPOIN,IPUSR,(COORD(IDIME),IDIME=1,3)

with :

IPOIN . . = the internal (sequential) node number (INTEGER*4);

IPUSR . = the external (user) node number (INTEGER*4);

IDIME . = a dimension index, varying from 1 to 3 (INTEGER*4);

COORD = the Cartesian coordinates vector (REAL*8) giving the position of node IPUSR.

15.1.3.4 Third section: Element information

NELEM records, each describing one element. A record can consist of more than one line.

Format	Content
(8I10)	IELEM,IEUSR,ITYPE,NNODE, (LNODE(INODE),INODE=1,NNODE)

with :

IELEM . = the internal (sequential) element number (INTEGER*4);

IEUSR . = the external (user) element number (INTEGER*4);

ITYPE . = the RAYNOISE type of element IEUSR (INTEGER*4);

NNODE = the number of nodes connected to element IEUSR (INTEGER*4);

INODE . = a local node index, varying from 1 to NNODE (INTEGER*4);

LNODS = the element connectivity vector (INTEGER*4) containing the external node numbers connected to element IEUSR.

The RAYNOISE element types (`ITYPE`) are defined as follows :

Element type	ITYPE	NNODE
LINE3	2	2
LINE4	3	3
TRIA3	4	3
TRIA6	5	6
QUAD4	6	4
QUAD8	7	8
TETR4	8	4
TET10	9	10

HEXA8	10	8
HEX20	11	20
PENT6	12	6
PEN15	13	15
PLG1	21	1
PLG2	22	2
...	...	...
PLG19	39	19
PLG20	40	20

15.1.4 Free format results files

15.1.4.1 File contents

RESULTS information consists of:

- Pressure values
- SPL values
- Speech values
- Music values
- Reverberation values
- Wideband values.

The results are known at the field points after **Analysis, Postprocess** (command POSTPROCESS, see).

Results are read/written in a RAYNOISE Free format RESULTS file in 7 sections.

15.1.4.2 First section: file header

The file header section contains 6 records:

Record #	Format	Content
1	(A80)	FTYPE
2	(A80)	REVIS
3	(A80)	TITLE
4	(A80)	CDATE
5	(A80)	LABEL
6	(E20.8)	FREQU

with :

FTYPE . = file type (CHARACTER*80; FTYPE='RAYNOISE RESULTS FILE');

REVIS . = software revision number and release date (CHARACTER*80);

TITLE . = model title defined by the command TITLE (CHARACTER*80);

CDATE . = file creation date and time (CHARACTER*80);

LABEL . = label (CHARACTER*80, LABEL='FREQUENCY');

FREQU = the frequency in Hz corresponding to the RESULTS (REAL*8).

15.1.4.3 Second section: Pressure values

NPOIN+1 records defining the pressure values at the field points:

Record #	Format	Content
1	(A80)	RETIT
2+	(2I10,3E20.8,/20X,3E20.8)	IPOIN,IPUSR,PREAL, PIMAG, PHASE, DECIB, PRRMS

(the last record is repeated NPOIN times)

with:

RETIT . = section title (CHARACTER*80; RETIT='PRESSURE VALUES);

IPOIN . . = the internal (sequential) node number (INTEGER*4);

IPUSR . = the external (user) point number (INTEGER*4);

PREAL . = the real part of the pressure value at point IPUSR (COMPLEX*16);

PIMAG . = the imaginary part of the pressure value at point IPUSR (COMPLEX*16);

MAGNI = the magnitude of the pressure at point IPUSR (REAL*8);

PHASE . = the phase (in degrees) of the pressure at point IPUSR (REAL*8);

DECIB . = the pressure in dB at point IPUSR (REAL*8);

PRRMS = the RMS value of the pressure at point IPUSR (REAL*8);

15.1.4.4 Third section: SPL values

NPOIN+1 records defining the SPL values.

Record #	Format	Content
1	(A80)	RETIT
2+	(2I10,3E20.8,/20X,3E20.8)	IPOIN,IPUSR, SPL,SPLA,DIRE,TDSR,NC,NR

(the last record is repeated NPOIN times)

with:

RETIT . = section title (CHARACTER*80; RETIT='SPL VALUES');

IPOIN . . = the internal (sequential) point number (INTEGER*4);

IPUSR . = the external (user) point number (INTEGER*4);

SPL = Sound Pressure Level (dB) at point IPUSR (REAL*8);

SPLA . . = Sound Pressure Level (dB/A) at point IPUSR (REAL*8);

DIRE . . = Direct Sound Pressure Level (dB) at point IPUSR (REAL*8);

TDSR . . = Total−to−Direct−Sound−Ratio (dB) at point IPUSR (REAL*8);

NC . = NC Value (dB) at point IPUSR (REAL*8);

NR . = NR Value (dB) at point IPUSR (REAL*8);

15.1.4.5 Fourth section: Speech values

NPOIN+1 records defining the SPEECH values.

Record #	Format	Content
1 :	(A80)	RETIT
2+:	(2I10,3E20.8,/20X,3E20.8)	IPOIN,IPUSR, DEF50,STI,ECS,ERR

(the last record is repeated NPOIN times)

with :

RETIT . = section title (CHARACTER*80; RETIT='SPEECH VALUES');

IPOIN . . = the internal (sequential) point number (INTEGER*4);

IPUSR . = the external (user) point number (INTEGER*4);

DEF50 . = Definition (%) at point IPUSR (REAL*8);

STI = Sound Transmission Index (–) at point IPUSR (REAL*8);

ECS = Echo Criterion Speech at point IPUSR (REAL*8);

ERR= Early Reflection Ratio (–) at point IPUSR (REAL*8);

15.1.4.6 Fifth section: Music values

NPOIN+1 records defining the MUSIC values.

Record #	Format	Content
1 :	(A80)	RETIT
2+:	(2I10,3E20.8,/20X,3E20.8)	IPOIN,IPUSR, CLAR,LE,TCG,ECM

(the last record is repeated NPOIN times)

with :

RETIT . = section title (CHARACTER*80; RETIT='MUSIC VALUES');

IPOIN . . = the internal (sequential) point number (INTEGER*4);

IPUSR . = the external (user) point number (INTEGER*4);

CLAR . . = Clarity (dB) at point IPUSR (REAL*8);

LE . = Lateral Efficiency (%) at point IPUSR (REAL*8);

TCG = Time Central Gravity (ms) at point IPUSR (REAL*8);

ECM= Echo Criterion Music (–) at point IPUSR (REAL*8);

15.1.4.7 Sixth section: Reverberation values

NPOIN+1 records defining the REVERBERATION values:

Record #	Format	Content
1 :	(A80)	RETIT
2+:	(2I10,3E20.8,/20X,3E20.8)	IPOIN,IPUSR, RT60,RT30,RT20,EDT

(the last record is repeated NPOIN times)

with :

RETIT . = section title (CHARACTER*80; RETIT='REVERBERATION VALUES');

IPOIN . . = the internal (sequential) point number (INTEGER*4);

IPUSR . = the external (user) point number (INTEGER*4);

RT60= Reverberation Time base 60 (s) at point IPUSR (REAL*8);

RT30= Reverberation Time base 30 (s) at point IPUSR (REAL*8);

RT20= Reverberation Time base 20 (s) at point IPUSR (REAL*8);

EDT = Early Decay Time (s) at point IPUSR (REAL*8);

15.1.4.8 Seventh section: Wideband values

NPOIN+1 records defining the wide−band values:

Record #	Format	Content
1	(A80)	RETIT
2+	(2I10,3E20.8,/20X,3E20.8)	IPOIN,IPUSR,SPL,SPLA, NC,NR,RZERO,RASTI

(the last record is repeated NPOIN times)

with :

RETIT . = section title (CHARACTER*80; RETIT='WIDEBAND VALUES');

IPOIN . . = the internal (sequential) point number (INTEGER*4);

IPUSR . = the external (user) point number (INTEGER*4);

SPL = Sound Pressure Level (dB/Lin) at point IPUSR (REAL*8);

SPLA . . = Sound Pressure Level (dB/A/Lin) at point IPUSR (REAL*8);

NC . = Noise Control (dB) at point IPUSR (REAL*8);

NR . = Noise Rating (dB) at point IPUSR (REAL*8);

RZERO = 0.0D0;

RASTI . = Rapid Speech Transmission Index (−) at point IPUSR (REAL*8);

15.1.5 Free format table files

15.1.5.1 Input and Output of tables

Many RAYNOISE data can be defined as frequency−dependent values, using tables. Tables can be read/written by RAYNOISE in a Free format file by commands such as:

```
RAYNOISE> INPUT TABLE FORMAT FREE FILE oldtable.fre
{   or
RAYNOISE> OUTPUT TABLE FORMAT FREE FILE newtable.fre
```

See *Tables*, Section 7.7 on 7.7, for more details on the use of tables.

The structure of the Free format table file is described below.

15.1.5.2 Main section

Record #	Format	Content
1	(4I10,A32)	ITABL,NOTAB,NPAIR,ITATY,TABNM
2+	(1P,3E12.5)	FREQU,CVALU
3	(1P,3E12.5)	SEVAR,CZERO

The main section is repeated for each table, if there are multiple tables. The second record is repeated as many times as their are entries table entries.

ITABL . . = internal table number (INTEGER*4);

NOTAB = external table number (INTEGER*4);

TABNM = table name (CHARACTER*16);

FREQU = frequency of the table entry(REAL*8);

CVALU = complex value of the table entry(REAL*8);

SEVAR . = separator variable (= -1.0) (REAL*8);

CZERO = (0,0) (COMPLEX*16);

16

Appendix 4: Frequency response function files

This chapter details the contents and format of results files produced by Frequency Response calculations:

16.1 Frequency response function files

16.1.1 General description of .RES files

This chapter documents the structure of the frequency response function file, created by the command RESPONSE (**Analysis, Response** dialog, see *Frequency response calculation*, Section 9.6 on Page 9–20).

The command COMBINE (**Tools, Combine** dialog, see *Writing results from the Combine tool*, Section 11.4.5 on Page 11–13) creates files with the same structure.

The files are formatted ASCII files and can be edited.

16.1.2 Content of frequency response files

The file structure is:

Line 1	Title,Date	(A60,A20)
Line 2	Nfreq,Npoin,Ncomp	(3I10)
Line 3	Separator	(80('–'))
Line 4+	Frequ, Press, CZERO, CZERO, CZERO, CZERO, CZERO, CZERO, CZERO, CZERO, CZERO	(E13.5/(6E13.5))

with:

Title:	Problem Title (CHARACTER*60)
Date:	Creation date (CHARACTER*20)
Nfreq:	Number of frequencies (INTEGER*4)
Npoin:	Point number (INTEGER*4)
Ncomp:	Number of results components (20)
Frequ:	Frequency (in Hertz; REAL*8)
Press :	Pressure (COMPLEX*16)
CZERO:	(0.0D0,0.0D0) (COMPLEX*16)

(The zero fields are to maintain compatibility with SYSNOISE and other programs).

17

Appendix 5: Converting data from Revision 2.1

This chapter describes procedures to convert models and results produced with RAYNOISE Revision 2.1, into the new data structure of Revision 3.0 and 3.1, including:

17.1 General principles of conversion

The data structures of RAYNOISE Revision 3.0/3.1 and RAYNOISE Revision 2.1A and previous versions are very different, both as regards the internal data handling during program execution and the files used for saving data and results. However, it has been made possible to load ‘old’ models used with Revision 2.1 into Revision 3.0/3.1, and similarly to load into Revision 3.0/3.1 results which were output from Revision 2.1.

The transfers of model data and results are in general done by reading the ASCII file formats which can be created by RAYNOISE Revision 2.1.

There is no compatibility of any binary files between the two Revisions.

This chapter describes the transfer procedures for model data and results, and how to control the re–running of calculations.

Refer to the installation instructions for your system, for details on upgrading the software from Revision 2.1 to 3.0 or 3.1.

17.2 To transfer a model from Rev 2.1 to 3.x

17.2.1 Recreating a model by reading command files

A model which was set up for RAYNOISE Revision 2.1 can be transferred to Revision 3.x, by reading the command file(s) used to create it, after some limited alterations which are detailed in the following sections. See *Reading a command file*, Section 4.6.4 on Page 4–14, for more information on using a command file in Revision 3.0 and 3.1.

It is not possible to load a RAYNOISE Revision 2.1 data file (*eg* produced by the SAVE DATA command) into Revision 3.x. To transfer a model, it must be regenerated using a command file.

17.2.2 Required alterations to modeling commands

17.2.2.1 Commands to be removed

The following commands are not applicable to Revision 3.x and should be completely removed from the command file, or commented–out:

OPTION

- This is because the Mapping and Image options are merged in Revision 3.x, and the choice of Conical or Triangular beam method is a calculation parameter. (Add a PARAMETER command if desired: see *Required alterations to calculation commands*, Section 17.2.3 on Page 17–7).

PLOT

- The view of the model is created and updated automatically in the Main Window, and explicit Plot commands are irrelevant and not interpreted.

17.2.2.2 Commands to be edited

The following commands have been changed in Revision 3.x and must be edited:

INPUT

- The *filename* is now case sensitive. This also applies to other file–reading and –writing commands. For example:

```
RAYNOISE> INPUT MESH FORMAT DXF FILE model1.dxf
RAYNOISE> {  reads a different file than
RAYNOISE> INPUT MESH FORMAT DXF FILE MODEL1.DXF
```

SET

- If elements which are later referenced as Receiver Surfaces are placed into Sets by the Revision 2.1 command file, this may cause problems when the RECEIVERSURFACE commands are decoded and the elements deleted from the mesh in Revision 3.x (see *Obsolete commands having special effect* below). Therefore, any such Set commands should be edited, removed or commented–out. Check for any references to these sets which may occur later in the Revision 2.1 command file, and alter them if necessary.

SOURCE

- The complement LEVEL should be changed to POWER

- The complement POSITION must be given in full (the normal rule that only the first four characters are required does not apply). The reason is, to distinguish Position from the complement Positive (used in Revision 3.x, to identify the side of an element on which a panel source lies).

- If there are non–default values for ORIENTATION, these must be changed to the new logic of Revision 3.x using a direction–cosines vector rather than a sequence of rotation angles. It may be easiest to do this by deleting the orientation information from the Revision 2.1 command file and later editing the source interactively in Revision 3.x.

- If there are non–default values for Directivity data (3DBHCUTTING and 3DBVCUTTING in Revision 2.1 command format) these must be changed to the new logic of Revision 3.x, by defining Directivity Table(s) and then applying these to the source(s). It may be easiest to do this by deleting the directivity information from the Revision 2.1 command file and later creating the tables and editing the source interactively in Revision 3.x.

- The complement PLANE should be changed to AREA

- For multi–point sources (Line, Area) the definition of Power has changed, so the values have to be adjusted. (See *Sound power level of sources*, Section 7.4.3.1 on Page 7–13, for the definitions of source power data). Alternatively, the power data for a source of this type can be edited after regenerating the model.

- The rest of the command syntax remains the same

SAVE

- The command SAVE DATA FILE *filename* is no longer effective

- Optionally, it can be replaced by database saving (command SAVE RETURN) or else deleted.
- Additional/alternative database commands and database saving on Exit from RAYNOISE may be used in Revision 3.x (see *Working with database files*, Section 5.2 on Page 5–4). This may be added to the command file, or executed via the dialogs (**File, Save...** *etc*) after command file input has finished.

Refer to the installation instructions for your system, for details on upgrading the software from Revision 2.1 to 3.0 or 3.1.

17.2.2.3 Obsolete commands having special effect

The following commands are not applicable to Revision 3.x, but have a special effect to enable upwards compatibility:

RECEIVERSURFACE

The command:

```
RAYNOISE> RECEIVERSURFACE ELEMENT e DIVISION i j RETURN
```

was used to create a receiver surface from a given element. By contrast, receiver surfaces are defined in Revision 3.x by the separate Field Point Mesh, and do not relate to elements of the imported geometry.

When read by Revision 3.x, the RECEIVERSURFACE command causes the element *e* to be removed from the acoustic model mesh. A field point mesh Plane is generated, using the vertices of the element *e* and with the divisions *i,j* along its sides, and added to any existing field point mesh. (*Creating a plane field point mesh*, Section 6.3.5 on Page 6–14, details the field point mesh plane generation tool).

Note that, after eliminating elements used for receiver surfaces, the acoustic model mesh is not the same as the mesh input to the model in Revision 2.1: it is reduced by the elimination of the elements used to generate the receiver surfaces.
To save the reduced mesh for future use in Revision 3.x, write it out using **File, Export, Mesh...** (see *Import and Export of a Mesh*, Section 5.3.5 on Page 5–11).

17.2.2.4 Commands which may be added

The following commands are new in Revision 3.x and may be added (optional):

NEW

The **File, New...** (or **Open...**) dialogs are unique to Revision 3.x, for the opening of new or existing database files. If these commands are not present, RAYNOISE Revision 3.x opens a new database file with the default name *default.rdb* (with a temporary copy named *_default.rdb*). The command:

```
RAYNOISE> New Name `modelname´ Model n File filename
```

can be added to the command file, to give a specific model name and database filename, or the **File, New...** dialog can be used before reading the command file.

- See *Working with database files*, Section 5.2 on Page 5–4, for details.

SAVE

- See *Commands to be edited*, Section 17.2.2.2 on Page 17–4, for details.

17.2.3 Required alterations to calculation commands

17.2.3.1 Commands to be edited

The following commands related to calculations are altered, due to the changes in procedures or terminology in Revision 3.x:

PARAMETERS

- The calculation parameter ORDERREFLECTION has been renamed REFLECTIONORDER and must be changed.

- Since the OPTION command is ineffective (see *Commands to be removed*, Section 17.2.2.1 on Page 17–4) a command PARAMETER RAYMETHOD *n* RETURN can be added to select the Conical (*n*=1) or Triangular (*n*=2) beam–tracing method. The default method is usually Triangular (installation–dependent, set by the Environment Variable RAYMETHOD). See *General parameters*, Section 9.5.5 on Page 9–16.

MAPPING

- The limitation of a Mapping calculation to a specified selection of receiver surfaces has been removed in Revision 3.x, therefore the complement RECEIVER <receiver–selection> should be removed.

- The Revision 2.1 Mapping command always launched a calculation at the eight octave–band center frequencies and no frequency selection was possible. In Revision 3.x, the frequency selection is arbitrary, but if no frequency selection is made (as is the case for a Mapping command from a Revision 2.1 command file) the eight octave–band center frequencies are chosen by default. Hence, the same frequencies will be calculated as in Revision 2.1.

POSTPROCESS

- The limitation of a Postprocess calculation to a specified selection of receiver surfaces has been removed in Revision 3.x, therefore the complement RECEIVER <receiver–selection> should be removed.
- The frequency–selection will default to the eight octave–band center frequencies, as in the case of MAPPING.

17.2.3.2 Obsolete commands having special effect

OUTPUT RESULTS

- It is recommended to remove this command from the command file:

The output of results to an ASCII file is much less common from Revision 3.x than was the case with Revision 2.1, due to the database file system in Revision 3.x. If a command of the form:

```
RAYNOISE> OUTPUT RESULTS FORMAT FREE FILE filename
```

remains in the command file, it is decoded by Revision 3.x. However, only results for the last–calculated frequency, plus wide–band results, are output.

17.3 To transfer results from Rev. 2.1 to 3.x

17.3.1 Loading results from a Free–format file

It is only possible to load results from Revision 2.1 onto the compatible model in Revision 3.x, using an ASCII Free format results file, *ie* what is written from Revision 2.1 by:

```
RAYNOISE> OUTPUT RESULTS FORMAT FREE FILE filename
```

In other words, a set of spatial results for a particular combination of sources and receivers can be transferred.

It is not possible to load the data from Revision 2.1 Mapping or Echo files or binary results files (command SAVE RESULTS...).

17.3.2 Results components which are transferred

Several results parameters which are now calculated in RAYNOISE Revision 3.x are not present in Revision 2.1 result files. The spatial data (receiver point values) will be transferred to Revision 3.x, provided that the field points in the Revision 3.x model are compatible, *ie* they have been generated with the old RECEIVERSURFACE command from Revision 2.1. (See *Obsolete commands having special effect*, Section 17.2.2.3 on Page 17–6).

After the transfer, the Revision 3.x model database contains the values calculated in Revision 2.1, for frequencies from 63Hz to 8kHz in octaves. Results that were not supported in Rev 2.1 are put to zero, or the relevant results section of the database is empty (as if the relevant results, such as echograms, were not requested during a calculation in Revision 3.x).

17.3.3 Procedure to load results from a Rev. 2.1 file

Before reading the results file, the Environment Variable FREEREVISION should be set to 2.1, so that the results file will be read in accordance with RAYNOISE Revision 2.1 formats. The default value of FREEREVISION is 3, *ie* an ASCII file in the format used by Revision 3.x.

The command to read the results is:

```
RAYNOISE> INPUT RESULTS FORMAT FREE FILE filename
```

There is no GUI dialog equivalent.

Example:

```
RAYNOISE> ENVIRONMENT FREEREVISION 2.1 RETURN
RAYNOISE> INPUT RESULT FORM FREE FILE CUBE.RES
```