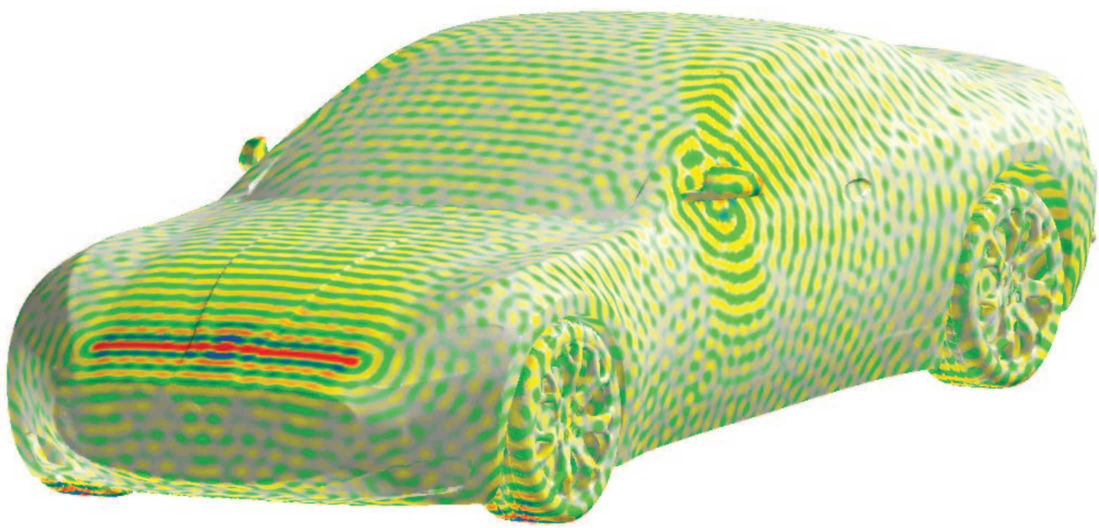
Coustyx User's Manual



Advanced Numerical Solutions
Hilliard OH

November 4, 2009

Contents

List of Figures

List of Tables

Preface

Coustyx computer program has been under development for many years, and is finally available for use by the noise and vibration community. This software is developed based on work supported by the National Science Foundation under Grant No. 0548629. We would like to thank the National Science Foundation for the support. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF.

Vijaya Kumar Ambarisha
Hilliard, OH
May 2009

Chapter 1

Introduction

Coustyx is a software program developed by Advanced Numerical Solutions (ANSOL), for the computation of steady state sound fields. Coustyx integrates the Fast Multipole Method (FMM) with Boundary Element (BEM) formulations to obtain rapid solutions to acoustic field problems. The small memory footprint of Coustyx, coupled with fast solvers allows simulation of very large problems in acoustics.

The Boundary Element Method (BEM) is widely used to predict sound radiation from vibrating mechanical components. BEM codes are popular since they involve only surface discretization and solve exterior (infinite domain) problems naturally. However, conventional BE methods suffer from a major drawback - the BEM coefficient matrices are *fully populated and frequency dependent*. This severely limits the size of models that can be built. The largest models that can be analyzed presently are limited to about $10,000$ unknowns. The dimensions of the elements/panels are related to the frequency range of interest, and thus a limitation on model size restricts the frequency range over which the BE model is useful. Presently, the usefulness of BEM is restricted to *coarse models of small objects* such as engine blocks, in the *low frequency regime*. In spite of its elegance and power, conventional BEM cannot be applied to aircraft interior, submarine exterior or architectural acoustics problems.

The Fast Multipole Method (FMM) is a recent breakthrough, which makes it possible to build intricate BEM models of real life systems and perform acoustic simulations efficiently. ANSOL has developed new BE formulations which are used in conjunction with iterative solvers from the Krylov family, and a new Multilevel Fast Multipole Method (MLFMM) that facilitates extremely fast matrix-vector product computations. Thus, *Coustyx* implements this approach to overcome the limitations of traditional BEM and allows to perform fast NVH analysis of large problems in the mid-to-high frequency regime.

The main developments in this software are:

1. Implementation of a new and improved FMM, in conjunction with the BE formulations, dramatically reduces the memory requirements of the problem, allowing it to handle large problems (up to 1 Million unknowns) with faster performance.

2. A novel BE formulation robust enough to deal with several problems associated with BE formulations, such as Thin Shape Breakdown (TSB), Irregular frequencies (or non-uniqueness issue), ability to model single sided surfaces such as Möbius strip, handle ex-

tremely complex junctions and inconsistent mesh normal orientations. Indepedent interpolation schemes for coordinates and acoustic variables provide greater modeling flexibility.

3. The formulation allows *Coustyx* to deal with wide variety of boundary conditions (BCs). Single sided BCs, Double sided BCs, Transfer BCs, Perforated BCs are all implemented.

4. Interpreter support to help users define custom analysis sequences. Coustyx provides the ability to execute scripts in a special programming language. For example, scripts can be written to define a complicated boundary condition, or to compute transmission loss for mufflers.

5. Parallel implementation of the code on shared memory multi-CPU computers allows efficient usage of system resources.

6. Numerous pre-processing tools provided in *Coustyx* cut down the time taken to setup an acoustic problem. The user can directly build *Coustyx* BE model from FE structure mesh, instead of using a third party software for converting FE meshes to BE meshes.

A very user-friendly Graphical User Interface (GUI) is developed to assist in setting up the acoustic problem by creating and manipulating boundary element models, building the model and running the analysis. It incorporates visualization tools and dialog boxes to query and modify *any parameter* associated with a model. Availability of wide variety of pre-processing tools in *Coustyx* make the model setup simple and easy. Some of the important pre-processing features are listed below.

- Import options: Ability to import FEA mesh and data from any of the three FEA programs: NASTRAN, ABAQUS and ANSYS, and Universal file format.

- Skinning: Ability to skin a finite element mesh to obtain surface mesh for *Coustyx* BE model.

- Fill holes: Ability to fill holes based on delaunay triangulation method.

- Stitch seams: Ability to seam stitch gaps between meshes.

- Create new elements or delete existing ones.

- Change element orientations.

- Automatic generation of junction constraint equations, jump conditions, duplicate acoustic nodes while skinning.

The main objective of this manual is to provide a comprehensive user's guide on how to run *Coustyx* for acoustic analysis. It describes how to import data files from different programs, setup a model, define boundary conditions, carry out an analysis and retrieve analysis results.

In Chapter 2, instructions on how to install *Coustyx* on a windows machine are provided.

In Chapter 3, we present the unit conventions followed in *Coustyx*.

In Chapter 4, we discuss the structure of the *Coustyx* User Interface and brief descriptions of all the steps required to setup an acoustic model for successfully carrying out an analysis.

In Chapter 5, we present more detailed discussion on how to generate a *Coustyx* BE mesh from a FE mesh.

In Chapter 6, a detailed discussion is provided on how to define and apply boundary conditions to a *MultiDomain* or an *Indirect Coustyx* model.

In Chapter 7, we provide details on how to set options required to run an analysis and how to retrieve analysis results for post-processing.

Chapter 8 provides the language syntax for writing scripts in *Coustyx* interpretive language.

Lastly Chapter 9 provides a Tutorial which outlines detailed steps to build *Coustyx MultiDomain*, and *Indirect* models from Finite Element meshes. It also explains how to perform acoustic analysis on these *Coustyx* models.

Chapter 2

Installing Coustyx

For installation on windows, you will need the windows installer file "Coustyx.msi" on your local computer. If you have not received *Coustyx* on a CD, please visit: `http://www.coustyx.com` and follow the downloading instructions provided on the web site. *Coustyx* is compatible with Windows XP, 2000, and Vista. For installation on linux machines contact us at sales@ansol.com

2.1. Software Installation

To start installation of Coustyx, double-click on "Coustyx.msi" or "Setup.exe". The installer will guide you through the steps required for the installation. *Important:* You will need Admin privileges to install *Coustyx*. Log into an account with administrator privileges.

2.1.1. License Agreement

Before proceeding any further please read the *Coustyx* user's license agreement carefully. If you agree with all the terms and conditions specified in the agreement, you can select "I Agree" to proceed further. If you do not agree, cancel the installation.

2.1.2. Select Installation Folder

By default, the folder selected to install *Coustyx* is:

```
C:\Program Files\Ansol\Coustyx\
```

You can change the installation folder by browsing through the folders using "Browse" button. The "Disk Cost" button lists all the available drives on your computer to which *Coustyx* can be installed, along with each drive's available and required disk space.(Figure 2.1)

If you see the error message shown in Figure 2.2 this means that you need to do a Windows Update.

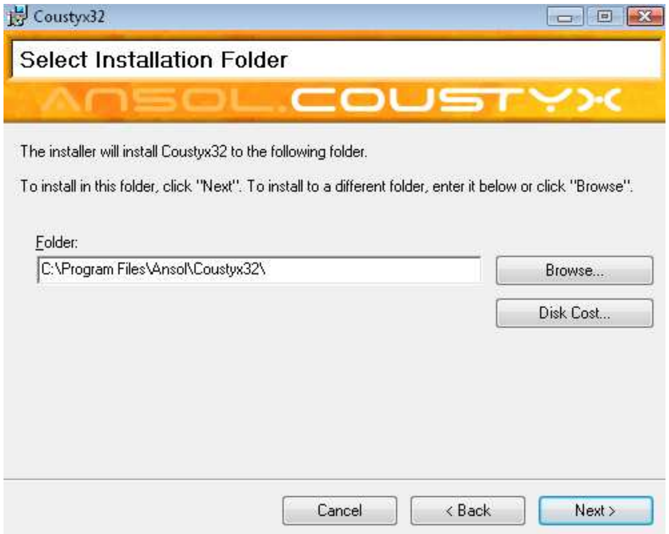


Figure 2.1: Select path window.

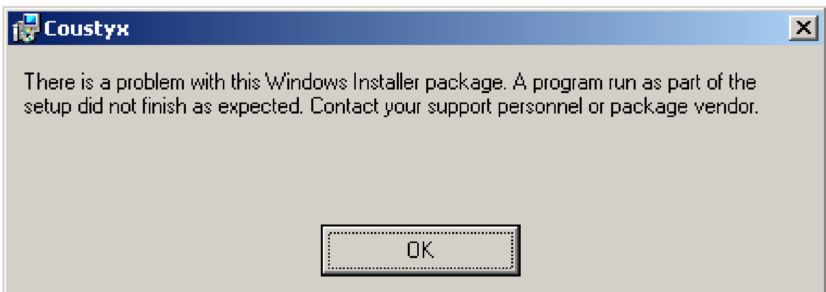


Figure 2.2: Windows installer error message.

2.1.3. Confirm Installation

After the installation folder is selected you will need to confirm to proceed with the installation. You can go back to the previous page to change the installation folder using *Back* button. Once the installation is finished close the window by clicking on *Close* button.

2.2. Dongle Device Driver Installation

If you have been provided a USB dongle, then the dongle device driver needs to be installed first. You will need Admin privileges for this step.

- To install the dongle device driver, open the *Coustyx* file folder and double click on the icon labeled SentinelDongleDeviceDriver.exe.
- Please read the user agreement and if you agree to all terms and conditions, select "I Agree" to proceed with the installation.
- After you have clicked on "I Agree", the driver will be installed.
- Click "Finish" to complete the installation of the dongle device driver.

2.3. License Key Installation

A valid License Key has to be installed before using the software. Open Install License Key window by clicking on the icon found in the "Start menu": Start—All Programs—Coustyx32 or Coustyx64—InstallLicenseKey. Figure 2.3 shows the Install License Key window. You can use any of the two licensing schemes listed in the window. Select the option **Use Native License** when you have a valid license key provided by ANSOL. Select **Use Altair GridWorks License** when you want to use Altair GridWorks License management system.

2.3.1. Use Native License

Choose this option if you want to use the license key provided by ANSOL.

2.3.1.1. If you have a local dongle

If you are using a ANSOL dongle attached to your local computer, the Install License Key window displays the dongle ID as shown in Figure 2.4. The license key will have already been sent to you with the dongle or by email. Copy and paste the license key into the License Key box. You can verify the features licensed under this key from the tabbed window "License Features" (Figure 2.6). If the license key is invalid or expired, appropriate information is highlighted (Figure 2.7). Contact us to renew your expired license. Click on the "Copy License Information" button to copy the information if you need to send us your license details. Press "OK" to install the license key and exit the window, or press "Cancel" to discard changes before exiting the window (Figure 2.5).

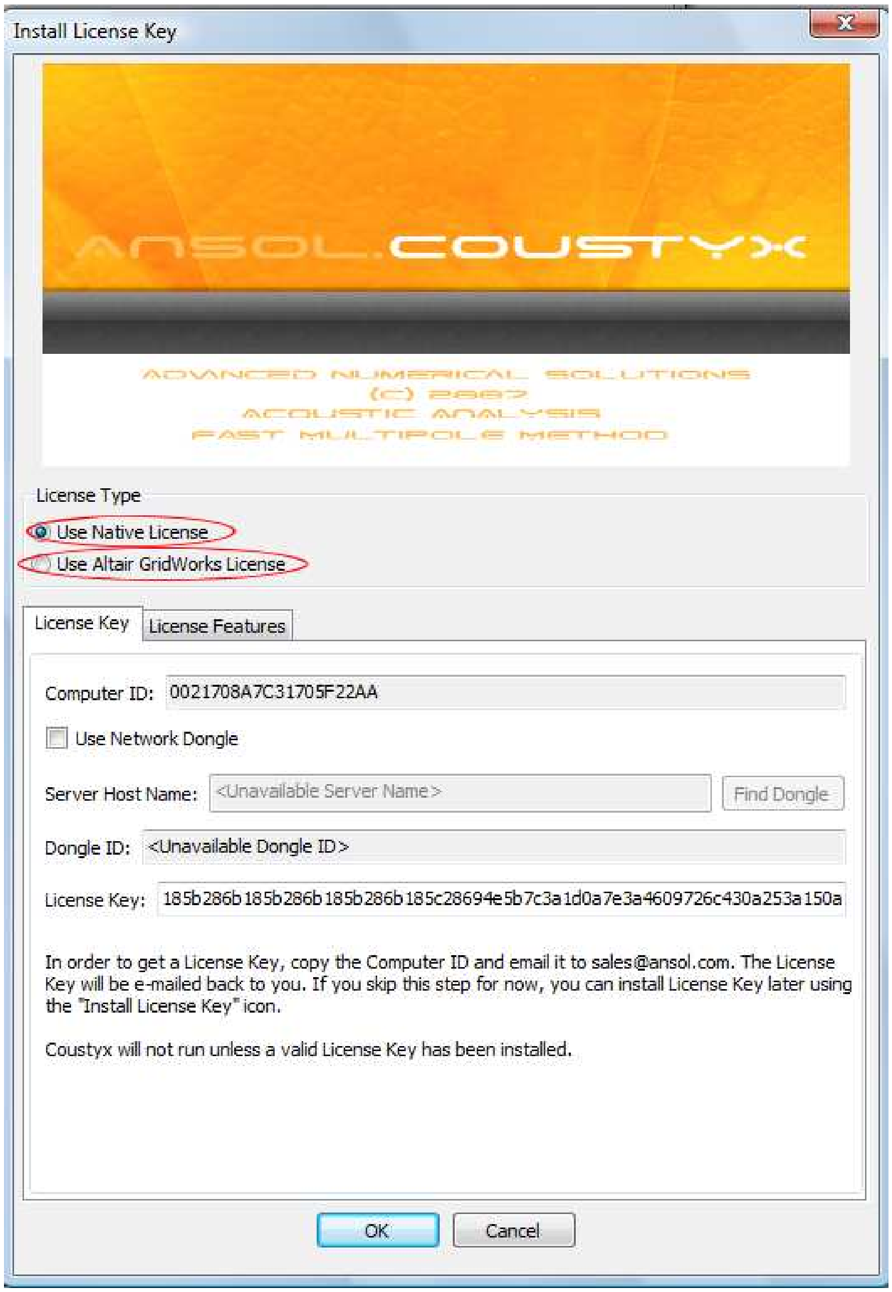


Figure 2.3: Install License Key window.

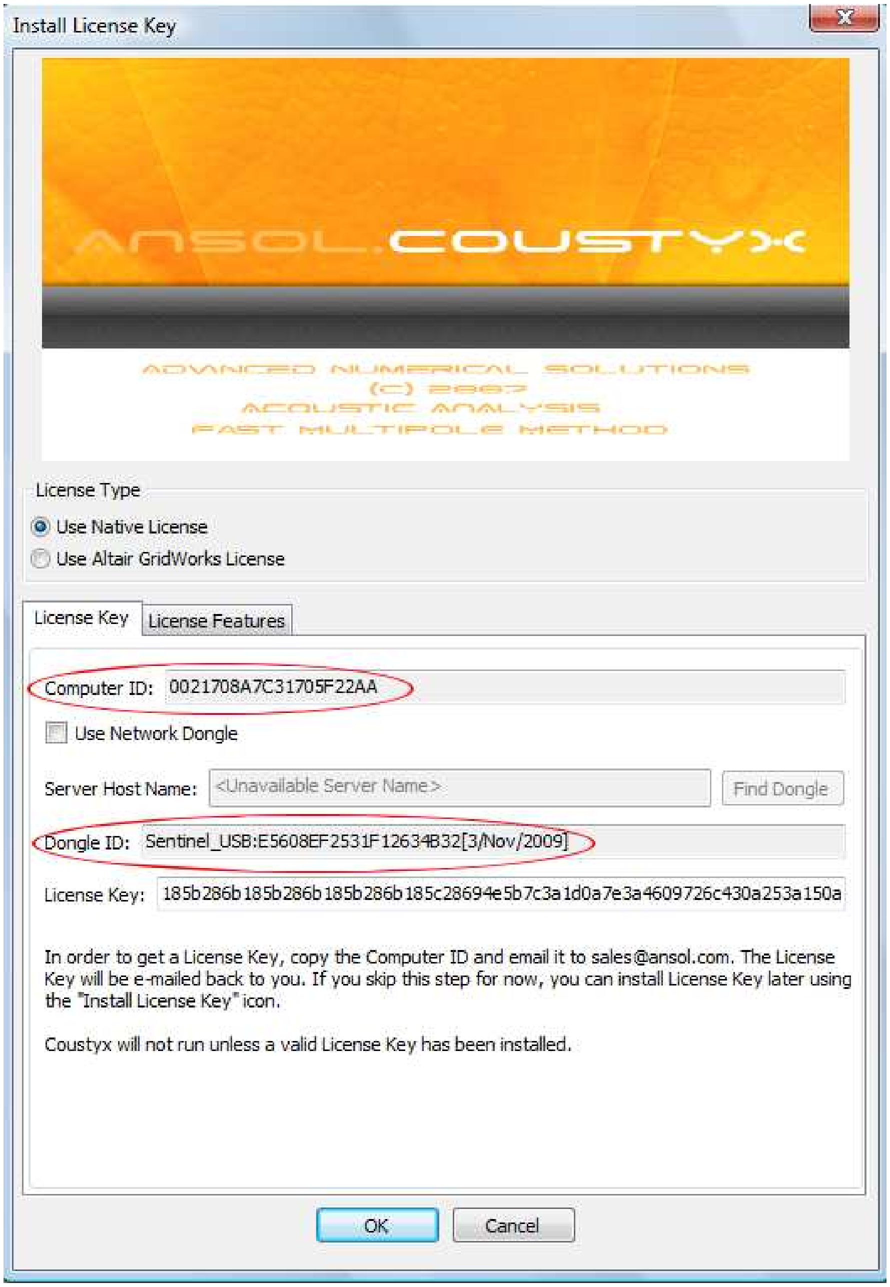


Figure 2.4: Computer and dongle IDs.

Figure 2.5: Steps to install a License Key.

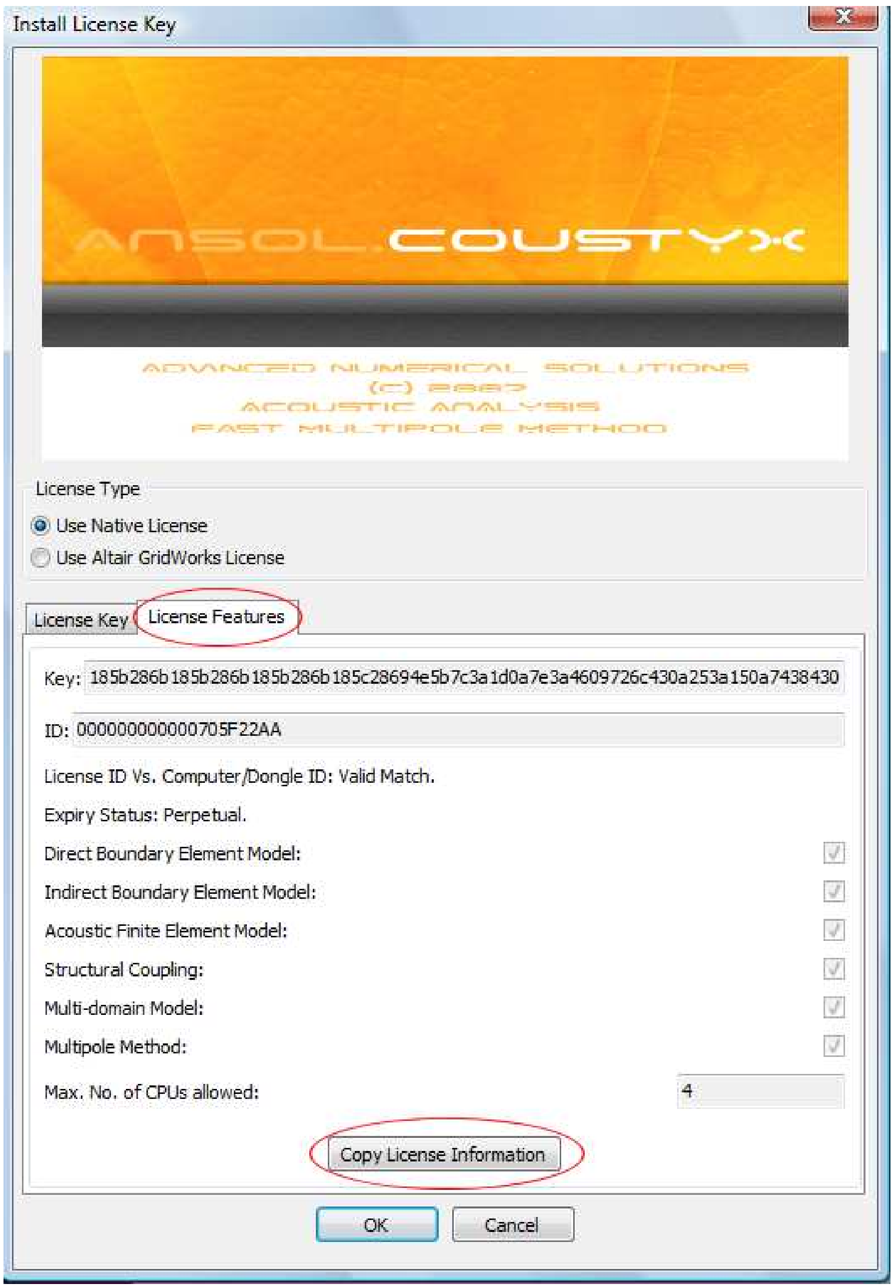


Figure 2.6: License Features window.

Figure 2.7: Invalid or expired License status.

2.3.1.2. If you have a network dongle

If you are using a network dongle, follow the steps below to install the license key.

Step 1: Start the License Server. Attach the dongle to your network server. Install SentinelDongleDeviceDriver.exe on the server (if it is not done already). This automatically starts the "Sentinel Key Server" service. Allow ports 7001 and 7002 to go through the server firewall. The status of the license server can be monitored via a web browser using the url: http://<*Server Internet Protocol (IP) Address*>:7002 (your browser should be java enabled). You can monitor the maximum number of seats (licenses) allowed and the number of seats in use.

Step 2: Install License Key. You can now install the license key from any computer connected to the server. Make sure Coustyx is installed on the computer. Open "Install License Key" window by clicking on the icon found in the "Start menu": Start—All Programs—Coustyx32 or Coustyx64—InstallLicenseKey. Choose the license scheme **Use Native License**. From the tabbed window "License Key", select the option "Use Network Dongle" (Figure 2.8). Enter the Server IP Address or Server Name in the "Server Host Name" box and click on "Find Dongle" to look for the network dongle. When the local computer finds a network dongle the Dongle ID will be updated. The license key will have already been sent to you with the dongle or by email. Copy and paste the license key into the License Key box. You can verify the features licensed under this key from the tabbed window "License Features" (Figure 2.6). If the license key is invalid or expired, appropriate information is highlighted (Figure 2.7). Contact us to renew your expired license. Click on the "Copy License Information" button to copy the information if you need to send us your license details. Press "OK" to install the key on to the network dongle and exit the window, or press "Cancel" to discard changes before exiting the window (Figure 2.5).

Step 3: Register Server Name. Make sure you register the server name on local computers that run Coustyx. Open "Install License Key" window by clicking on the icon found in the "Start menu": Start—All Programs—Coustyx32 or Coustyx64—InstallLicenseKey. Choose the license scheme option **Use Native License**. From the tabbed window "License Key", select the option "Use Network Dongle". Enter the Server IP Address or Server Name in the "Server Host Name" box and click on "Find Dongle" to look for the network dongle. When the local computer finds a network dongle the Dongle ID and the License Key will be updated. Press "OK" to register and exit the window, or press "Cancel" to discard changes before exiting the window (Figure 2.5).

2.3.1.3. If you are using *Coustyx* without a dongle

Copy the Computer ID (Figure 2.4) and email it to sales@ansol.com in order to get your license key. After receiving the key, open Install License Key window from the "Start menu": Start—All Programs—Coustyx32 or Coustyx64—InstallLicenseKey. Copy and paste the license key into the License Key box. You can verify the features licensed under this key from the tabbed window "License Features" (Figure 2.6). If the license key is invalid or expired, appropriate information is highlighted (Figure 2.7). Contact us to renew your expired license. Click on the "Copy License Information" button to copy the information if you need to send us your license details. Press

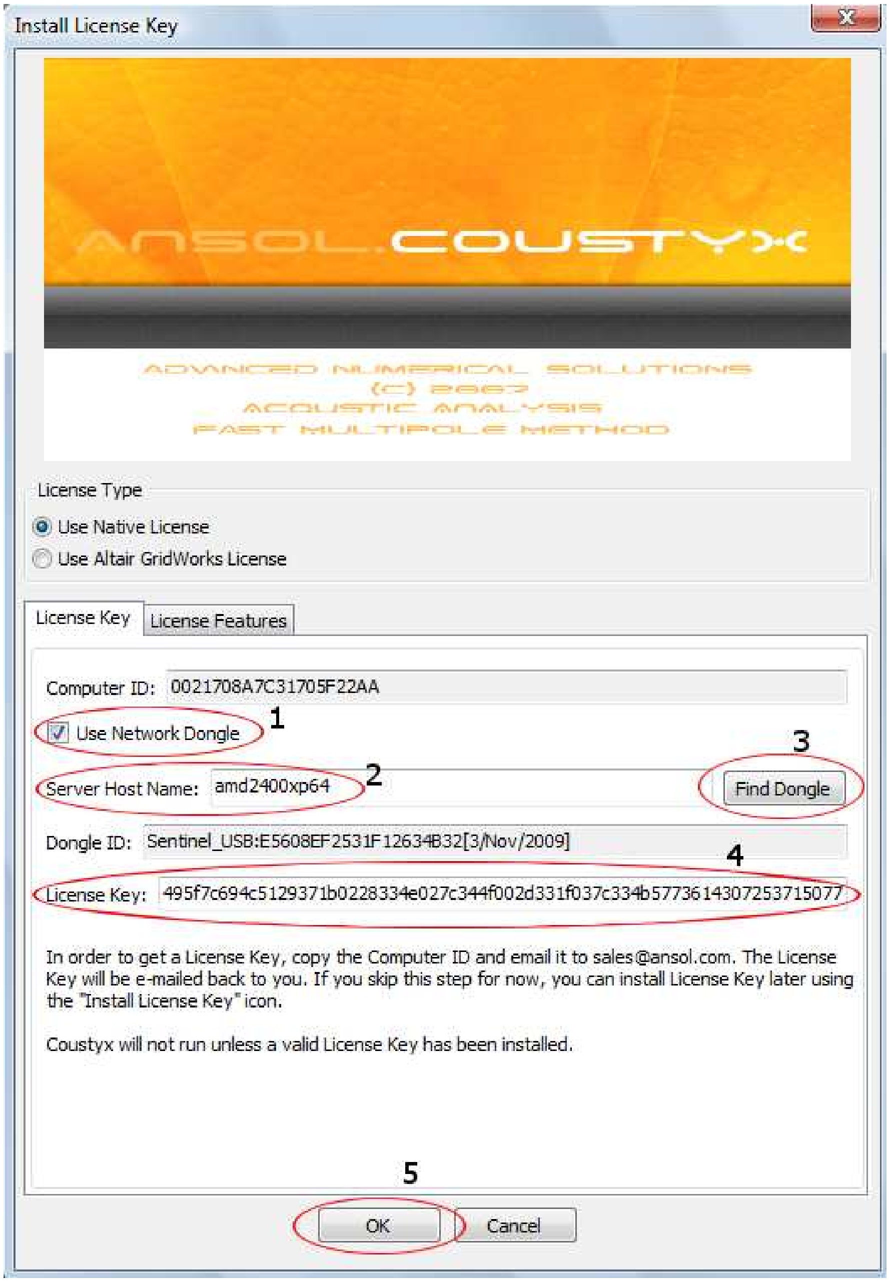


Figure 2.8: Steps to install a network dongle.

"OK" to install the license key and exit the window, or press "Cancel" to discard changes before exiting the window (Figure 2.5).

2.3.1.4. If you are using a Demo Model

If you are using a demo model posted on our web site, you do not need to install a license key. The demo models have embedded license keys which are valid only for those models. Any changes to the model will make the license key invalid.

2.3.2. Use Altair GridWorks License

Coustyx has partnered with Altair to enable users run Coustyx under Altair's GridWorks License Management (ALM) system. For more information on GridWorks License Management system, contact your local Altair software distributor or visit: `http://www.altair.com`. Under this licensing system, a predetermined number of tokens, also called GridWorksUnits (GWUs), will be drawn from the Altair License server each time Coustyx is invoked. These GWUs are returned to the server once the user exists Coustyx. The number of GWUs drawn varies with the number of CPU-cores utilized.

To use this license scheme, make sure your computer is connected to a server running the Altair License manager and the value of the environment variable `LM_LICENSE_FILE` or `ALTAIR_LM_LICENSE_FILE` is set to `7788@servername` or `portnumber@servername`. Contact your network administrator to find the port number and the server name on which Altair License manager is running. Now, open "Install License Key" window by clicking on the icon found in the "Start menu": Start—All Programs—Coustyx32 or Coustyx64—InstallLicenseKey. Choose the license scheme **Use Altair GridWorks License**. Press the **Test** button to check the connection to the license server and the number of available GridWorksUnits (Figure 2.9). Press "OK" to install the license and exit the window, or press "Cancel" to discard changes before exiting the window.

2.3.3. Modify Path Environment Variable

Coustyx allows you to append the installation folder path to your computer's Path environment variable. This helps avoid having to specify the full path name of the folder containing the executable every time you call Coustyx.exe from the command prompt (Figure 2.10).

2.4. Running Coustyx

After installation, you can start using *Coustyx* by opening the program from the "Start" menu: Start—All Programs—Coustyx32 or Coustyx64—Coustyx. You can also run *Coustyx* from the command prompt.

2.4.1. Running from the Command (DOS) Prompt

Coustyx can be run in interactive mode or in batch mode. To run the program interactively, type coustyx at the command prompt and press enter. In order to avoid entering the full path every time you call the executable, we advise you to modify the Path environment variable.

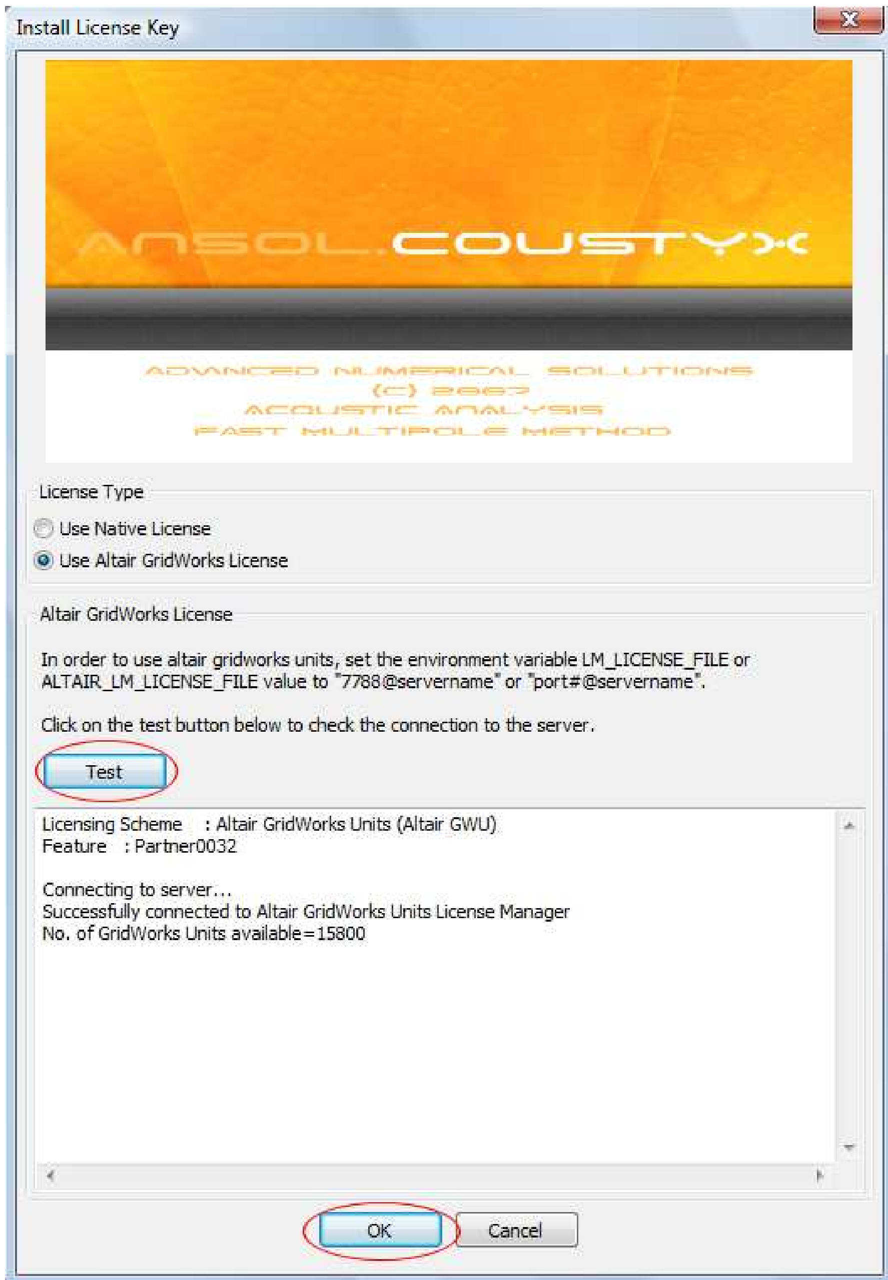


Figure 2.9: Use Altair GridWorks License.

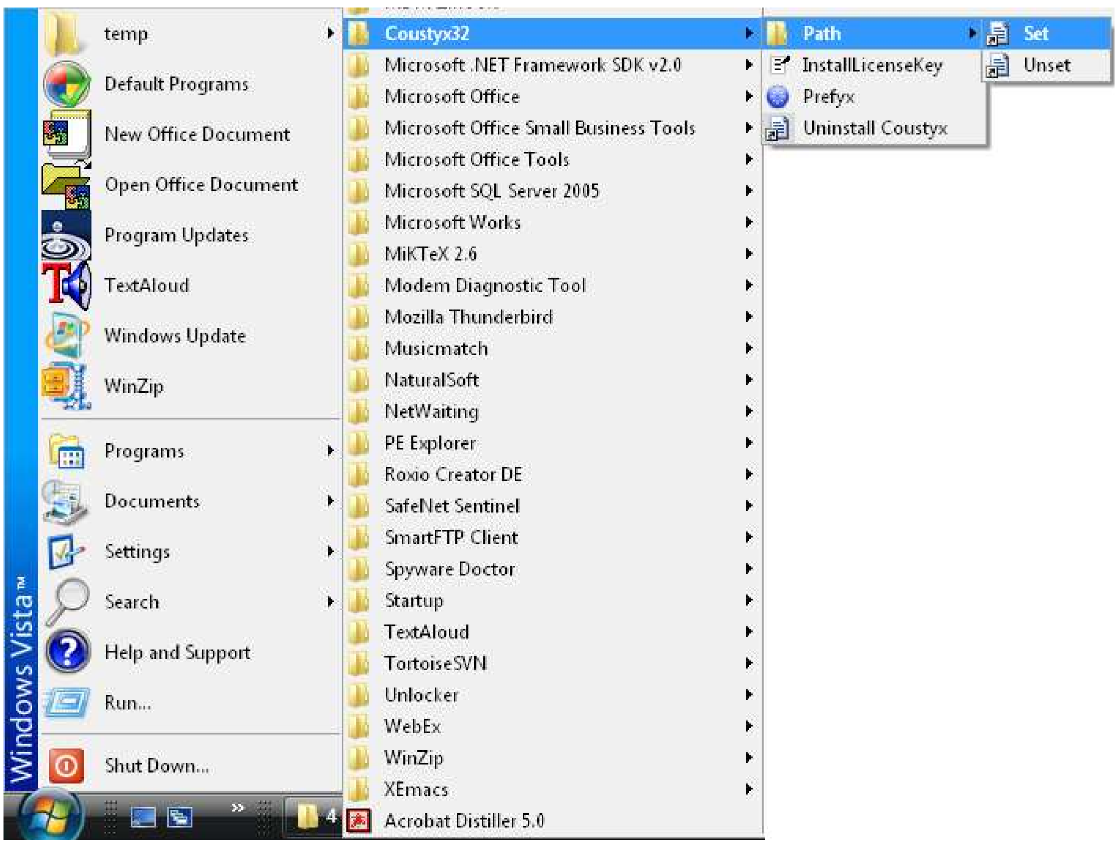


Figure 2.10: Set path variable.

See Section 2.3.3 (Figure 2.10). Coustyx accepts the command line arguments explained below. These command line arguments may appear in any sequence.

- `-model=`*ModelFileName* specifies the name of the model file. There should be no space between the `=` and *ModelFileName*. The *ModelFileName* is a ∗*.cyx* file. If there are spaces in the *ModelFileName*, it should be enclosed within "". This file contains information about the *Coustyx* model. It also can contain analysis sequences which are invoked to run the model.
- `-analysissequence=`*AnalysisSequenceName* specifies the name of the analysis sequence to be used to run the analysis. The *AnalysisSequenceName* is defined inside the model specified by *ModelFileName*. If there are spaces in the *AnalysisSequenceName*, it should be enclosed within "". This contains details on how to run an analysis, the solution method to be used, etc. It is always used in conjunction with argument -model=*ModelFileName*.
- `-nthreads=`*NumberThreads* specifies the number of threads to be used to run the analysis. For a computer with dual-core CPU, you can specify `-nthreads=2` to use multiple cores to speed up the analysis.
- `-command=`*ScriptFileName* specifies the name of the Coustyx script file. There should be no space between the `=` and *ScriptFileName*. The *ScriptFileName* is a ∗*.csr* file. If there are spaces in the *ScriptFileName*, it should be enclosed within "". This file contains commands which tell *Coustyx* from where to read the model, and how to run the analysis. The model file it reads is a ∗*.cyx* file.
- `-workdir=`*WorkingDirectoryName* specifies the name of the working directory. There should be no space between the `=` and *WorkingDirectoryName*. This argument is optional. If omitted, *Coustyx* will assume that the current directory at the time *Coustyx* command was started is the working directory.

The following are the possible ways to run *Coustyx* from command prompt:

```
C:\users\johnsmith>coustyx -model=sphere.cyx -analysissequence="Analysis Sequence"
```

the file `sphere.cyx` contains the information defining the model configuration and the script `Analysis Sequence`. `Analysis Sequence` contains commands telling *Coustyx* how to run the analysis. Note that `Analysis Sequence` is not a separate file but is already defined inside the model. Since the `-workdir` argument has not been provided, *Coustyx* uses the directory `C:\users\johnsmith` as the working directory.

```
C:\users\johnsmith>coustyx -command=radiation.csr
```

the file `radiation.csr` is a Coustyx script file that contains functions to read in the model (∗*.cyx*) file and other commands on how to run the analysis. Since the `-workdir` argument has not been provided, *Coustyx* uses the directory `C:\users\johnsmith` as the working directory.

Chapter 3

Conventions in Coustyx

In this chapter we will discuss some of the important conventions followed in *Coustyx*.

3.1. Time Dependence

The time dependence of oscillating quantities in *Coustyx* follow a $e^{-j\omega t}$ convention, where $j = \sqrt{(-1)}$. For example, the time-harmonic pressure wave $P(t)$ in *Coustyx* is defined as

$$P(t) = \mathbf{Re}\left\{pe^{-j\omega t}\right\}$$

where **Re** stands for "real part of", p is the complex amplitude of the sound pressure, and ω is the frequency of fluctuation.

3.2. Units

3.2.1. Model Units

The analysis process in *Coustyx* is independent of the system of units. However, to assist users in keeping track of units for various physical quantities, *Coustyx* provides an option to choose *model units* before building a new model. *Model units* are stored in the model for *reference only.* It is the user's responsibility to maintain consistent units among various inputs in the model. Figure 4.16 shows the dialog box used to select model units before building a new model. The model units specified are stored in the model tree member "Units" (Figure 3.1). Below are two places where the model units are used:

- Model units are used to setup appropriate material properties (speed of sound and ambient density) for commonly used materials - Air and Water, at the start of a new model setup. Later on material properties can be changed only through "Materials" model tree member and not by modifying model units. Note that editing model units through "Units" model tree member does not rescale the model.

- Model units are also used while computing sound power levels from ISO standards. The acoustic variables are converted from model units to S.I units before using the empirical relations in ISO standards.

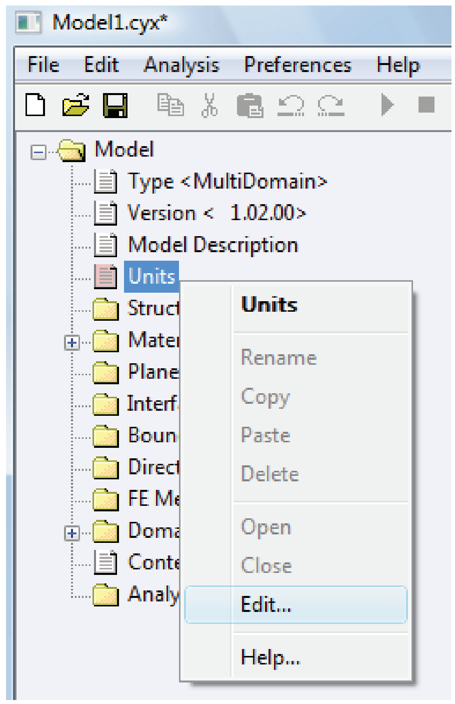


Figure 3.1: Units model tree member.

Select: **Model** → **Units**.

The model units are set at the start of a new model setup (Figure 4.16). To edit units later on, right-click on **Units** in the model tree and select: **Edit** (Figure 3.1). An edit dialog box appears and you can make changes to model units here (Figure 3.2). Note that modifying the model units does not rescale the model.

For convenience, six different standard unit systems are predefined: **meter - kilogram - second** (m – kg – s), **millimeter - newton - second** (mm – N – s), **meter - kilogram force - second** (m – kgf – s), **millimeter - kilogram force - second** (mm – kgf – s), **inch - pound force - second** (inch – lbf – s), and **foot - pound force - second** (ft – lbf – s). Other unit systems can be set by choosing **other** and entering the values for **Length scale factor** and **Mass scale factor** (Figure 3.2).

Length scale factor The length scale factor is the conversion factor from model length units to meters. For example, if the model unit for length is millimeter (mm), then the length scale factor is 0.001, since 1 mm = 0.001 m. Table 3.1 shows length scale factors for some commonly used length units.

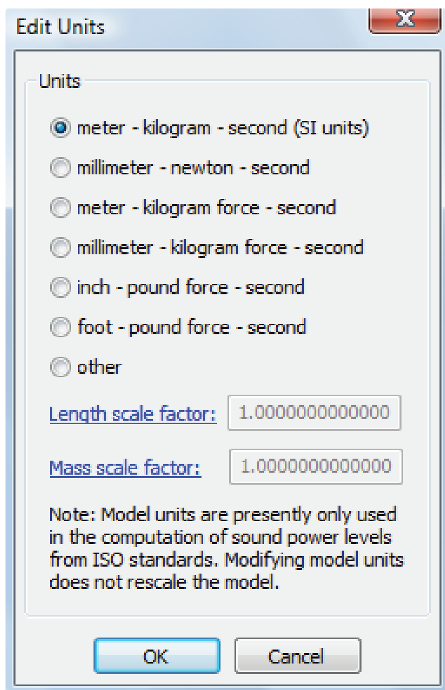


Figure 3.2: Edit units dialog box.

Table 3.1: Length scale factors.

Model units	Length scale factors
meter	1
millimeter	0.001
inch	0.0254
foot	0.3048

Table 3.2: Mass scale factors.

Model units	Mass scale factors
kilogram	1
gram	0.001
pound (lb)	0.45359

Mass scale factor The mass scale factor is the conversion factor from model mass units to kilogram. For example, if the model unit for mass is pound mass (lb), then the mass scale factor is 0.45359. Note that "lb" is international "avoirdupois pound" and $1\,\text{lb} = 0.45359\,\text{kg}$. Table 3.2 shows mass scale factors for some commonly used mass units.

3.2.2. How to choose model units?

It is very important for the user to clearly understand the units he or she is working with. The building block for any *Coustyx* model is the mesh geometry. *First, identify the units for length in the geometry.* Use this as the *model unit for length.* Choose any unit for mass. The commonly used unit of time is *seconds.* Once the model units for **Length**, **Mass** (and **Time**) are identified, all the inputs to the model should be scaled to be consistent with these units. **Speed of Sound** and **Ambient Density** are two such important inputs that should be scaled correctly before they are input into the model. Other inputs, such as boundary conditions, acoustic source strengths, etc., should also be scaled correctly. The units for all the acoustic metrics derived from the analysis, such as pressure, velocity, power, intensity, etc., depend on the units used to define the Speed of Sound and Ambient Density.

- Identify the units for length in the mesh geometry. Select model units based on this.
- Scale all model inputs, such as speed of sound, ambient density, boundary condition values, acoustic source strengths etc., to be consistent with model units.
- Derived acoustic variables will have units consistent with the model units. A simple dimensional analysis would be sufficient to derive the units for any derived quantity.

For example, consider a model with mesh geometry in **meters**. Select the model units to be **meter - kilogram - second** ($\text{m} - \text{kg} - \text{s}$). As shown in Table 3.3, the speed of sound should

Table 3.3: Variables and Units

Model units	Inputs			Outputs		
	Geometry units	Speed of sound	Ambient Density	Pressure	Particle Velocity	Sound Power
$\mathrm{m-kg-s}$	m	m/s	$\mathrm{kg/m^3}$	$\mathrm{kg/m/s^2}$ (Pa)	m/s	$\mathrm{kg-m^2/s^3}$ (W)
$\mathrm{mm-N-s}$	mm	mm/s	$\mathrm{N-s^2/mm^4}$	$\mathrm{N/mm^2}$(MPa)	mm/s	$\mathrm{N-mm/s}$ (mW)
$\mathrm{m-kgf-s}$	m	m/s	$\mathrm{kgf-s^2/m^4}$	$\mathrm{kgf/m^2}$	m/s	$\mathrm{kgf-m/s}$
$\mathrm{mm-kgf-s}$	mm	mm/s	$\mathrm{kgf-s^2/mm^4}$	$\mathrm{kgf/mm^2}$	mm/s	$\mathrm{kgf-mm/s}$
$\mathrm{inch-lbf-s}$	inch	inch/s	$\mathrm{lbf-s^2/inch^4}$	$\mathrm{lbf/inch^2}$	inch/s	$\mathrm{lbf-inch/s}$
$\mathrm{ft-lbf-s}$	ft	ft/s	$\mathrm{lbf-s^2/ft^4}$	$\mathrm{lbf/ft^2}$	ft/s	$\mathrm{lbf-ft/s}$

Table 3.4: Values for acoustic medium properties of air in different units

Model units	Material Properties	
	Speed of Sound	Ambient Density
$\mathrm{m-kg-s}$	343 m/s	$1.21\,\mathrm{kg/m^3}$
$\mathrm{mm-N-s}$	343000 m/s	$1.21\times10^{-12}\,\mathrm{N-s^2/mm^4}$
$\mathrm{m-kgf-s}$	343 m/s	$0.1233857\,\mathrm{kgf-s^2/m^4}$
$\mathrm{mm-kgf-s}$	343000 mm/s	$0.1233857\times10^{-12}\,\mathrm{kgf-s^2/mm^4}$
$\mathrm{inch-lbf-s}$	13503.94 inch/s	$1.132228\times10^{-7}\,\mathrm{lbf-s^2/inch^4}$
$\mathrm{ft-lbf-s}$	1125.33 ft/s	$0.0023478\,\mathrm{lbf-s^2/ft^4}$

be in m/s and the ambient density should in $\mathrm{kg/m^3}$. The derived acoustic quantities, such as acoustic pressure should be in Pascal ($1\,\mathrm{Pa} = 1\,\mathrm{kg/m/s^2}$), the particle velocity in m/s, the sound power in Watts ($1\,\mathrm{W} = 1\,\mathrm{kg.m^2/s^3}$), and the intensity in $\mathrm{Watt/m^2}$.

Consider another model with mesh geometry in **inches**. Set the model units to **inch - pound force - second** ($\mathrm{inch-lbf-s}$). Following Table 3.3, the speed of sound should be in inch/s and the ambient density should be in $\mathrm{lbf-s^2/inch^4}$. The derived acoustic quantities, such as acoustic pressure should be in psi ($1\,\mathrm{psi} = 1\,\mathrm{lbf/inch^2}$), the particle velocity in inch/s, the sound power in $\mathrm{lbf-inch/s}$, and the intensity in lbf/inch/s.

As a reference, Table 3.4 shows the possible values for acoustic medium properties of air in different unit systems.

3.3. Frequency Dependence Type

The acoustic variables pressure, normal derivative of pressure, impedance, or material properties such as sound speed are complex values which can vary with position, normal or frequency.

The dependence on frequency is categorized into three different types shown below.

3.3.1. Constant

The *Frequency Dependence Type* is defined as a *Constant* when the acoustic variable doesn't vary with frequency. The constant real and imaginary values of the variable are entered.

3.3.2. Table

The *Frequency Dependence Type* is defined as a *Table* when the frequency variation of the acoustic variable is given by a table. The frequency variation can be entered manually into the table or it can be *imported* from an *ASCII* formatted text file with values separated by commas, tabs, or spaces. The values can be copied from the table to the clip board or deleted from the table by selecting the menu options *Copy* and *Delete Rows*. First select the desired row by left-clicking the mouse on the row number, and then right-click to see the menu. (Figure 3.3, Figure 3.4, Figure 3.5).

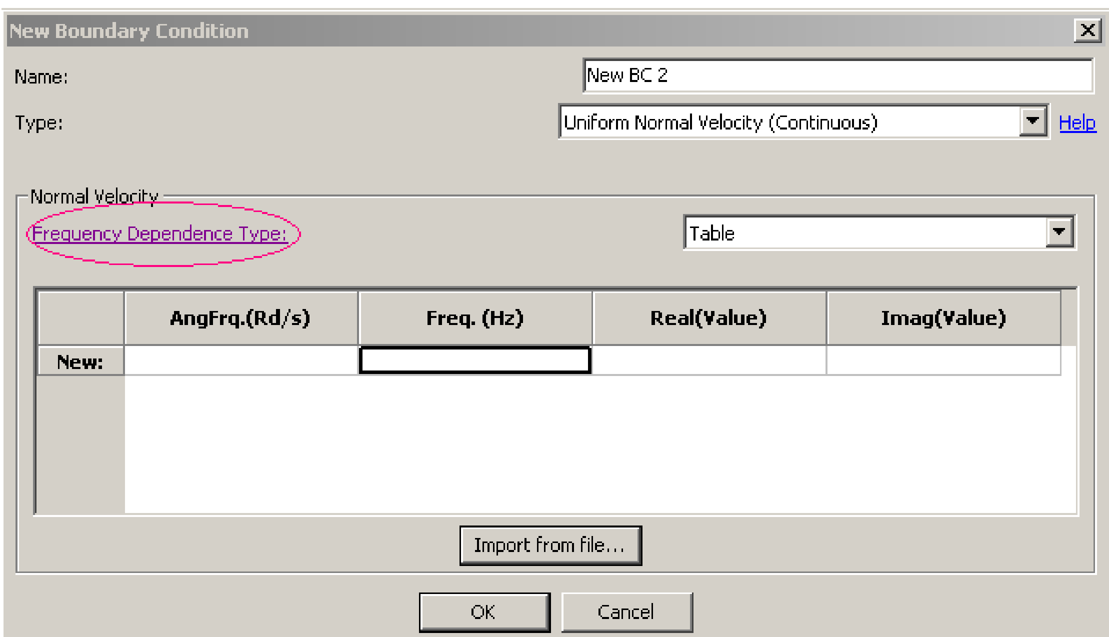


Figure 3.3: Boundary Condition Table.

3.3.2.1. Import from File

The frequency variation of the variable can be imported from an *ASCII* file with components separated by commas, tabs, or spaces. The file must contain three columns with the first column

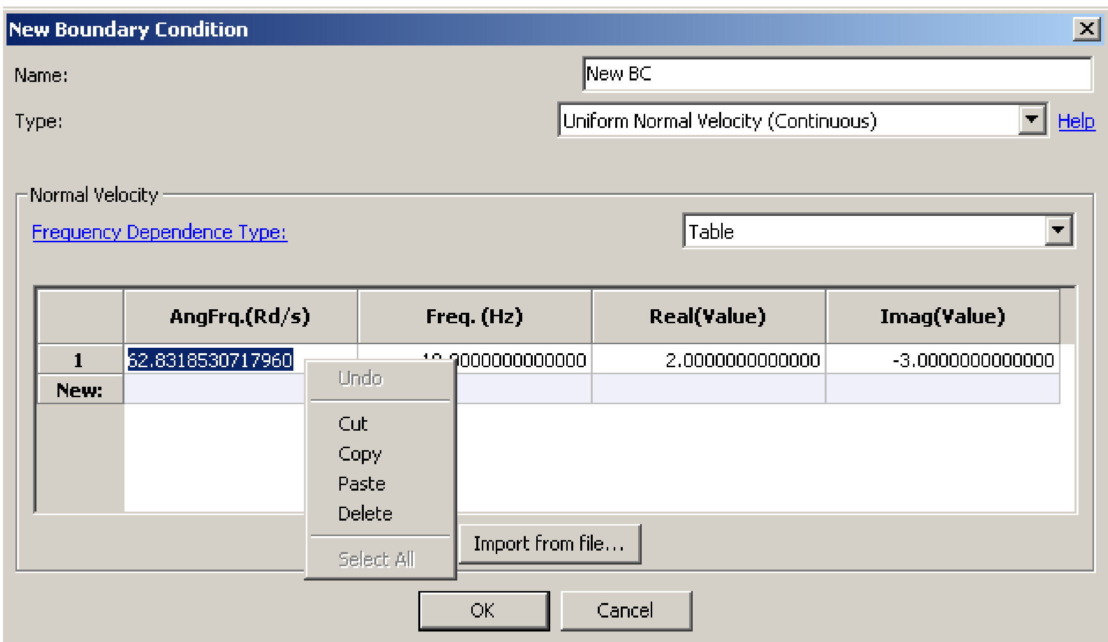


Figure 3.4: Edit Boundary Condition Table.

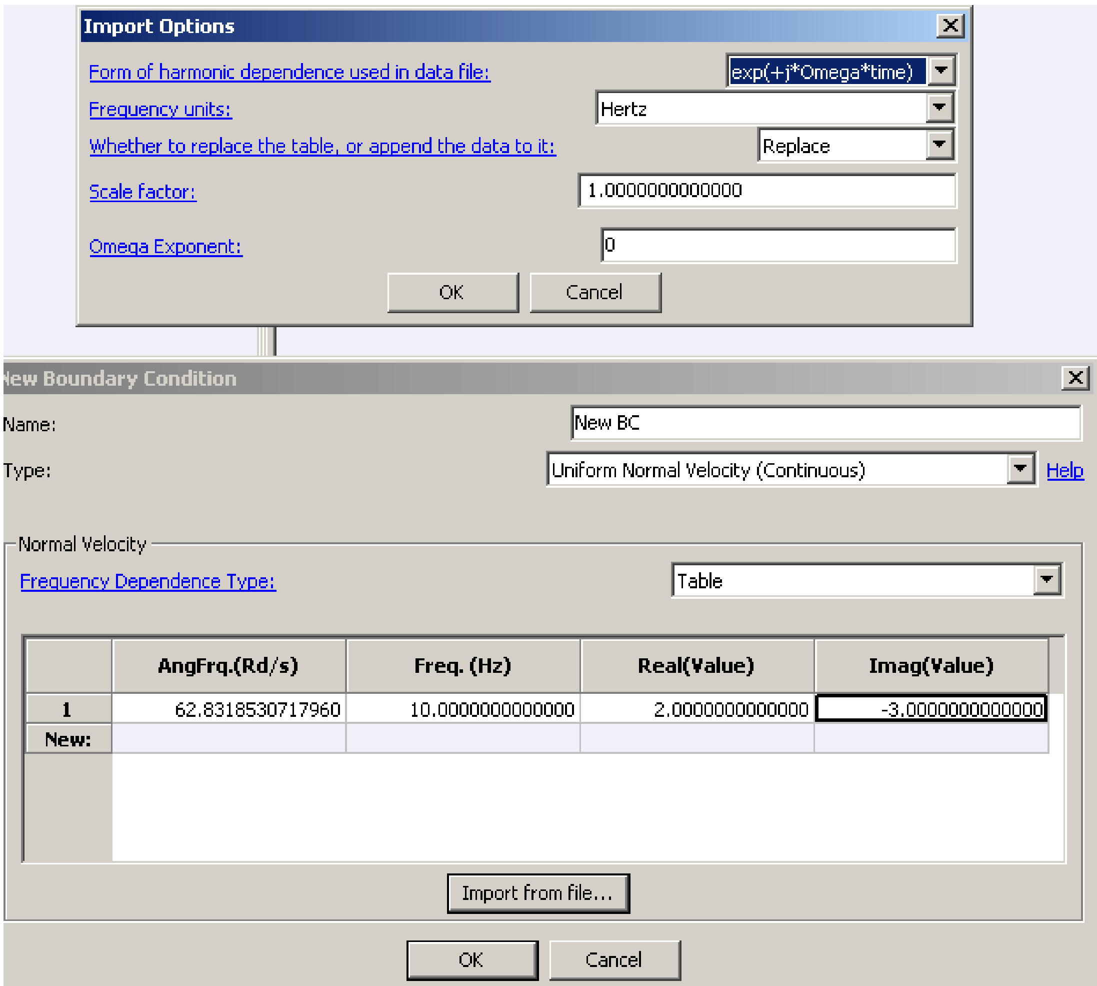


Figure 3.5: Table Boundary Condition example.

for the frequency, the second for real part of the variable and third for its imaginary part. Each new frequency is added to a new row.

3.3.2.2. Import Options

The import options window, Figure 3.6, appears after the selection of the *ASCII* file to be used to import the table. This window controls the interpretation of the data from the file.

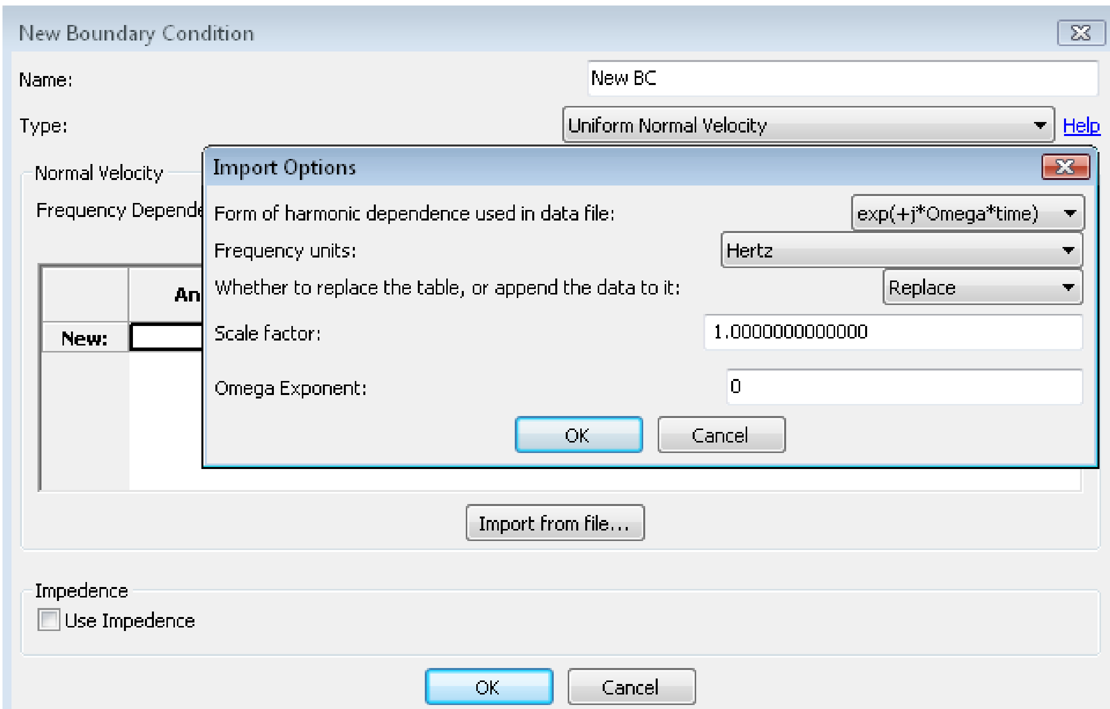


Figure 3.6: Table import options window.

Form of harmonic dependence used in data file **exp**$(+$**j** $*$ **Omega** $*$ **time**$)$ (or $e^{+j\omega t}$) and **exp**$(-$**j** $*$ **Omega** $*$ **time**$)$ (or $e^{-j\omega t}$) are the two available options, where **Omega** (ω) is the angular frequency in $radians/s$, and $j = \sqrt{(-1)}$. Since *Coustyx* follows $e^{-j\omega t}$ convention, the values in the data file are adjusted prior to importing.

Frequency units The first column of the *ASCII* file from which the data for the table is imported contains frequency values. This frequency could be defined in any one of the following units (Figure 3.6). Appropriate unit conversions are applied based on this selection. Available options are **Hertz**, **Radians/sec**, **RPM**.

Whether to replace the table, or append the data to it The imported data from the file can be used to either replace the current table or append to the existing table by the selection of one of the options: **Replace** or **Append**. (Figure 3.6).

Scale factor All the values in the *ASCII* file (except the first column, which is the frequency) are multiplied by the Scale factor before being read into the table (Figure 3.6). This is specifically useful when the imported file has different units compared to *Coustyx* model. A unit conversion factor should be used as the Scale factor to convert these values. For example, when the velocities in the *ASCII* file are in m/s and the *Coustyx* units are in mm/s, a Scale factor of 10^3 is entered to convert the values in the file from m/s to mm/s. A Scale factor of 1 imports the values as they are.

Omega exponent The Omega exponent sets the power of $j\omega$, which is then multiplied with all the values in the *ASCII* file (except the first column which is the frequency ω) before being read into the table (Figure 3.6). This is useful when the imported values from the file vary from the variable values in the table by a factor of $j\omega$. For example,

- Assume that a velocity boundary condition needs to be applied through the Table option. But the user has only the displacement data in an *ASCII* file. The Table can still be populated by importing displacement values from the *ASCII* file and setting the Omega exponent value to 1. That is, the velocity values are obtained by multiplying the displacement values in the file with $j\omega$, $v = (j\omega)s$, where $s = s_o e^{j\omega t}$ is the displacement variation, v is the corresponding velocity variation.
- Similarly, when only acceleration data is available, the velocity values can be obtained by setting the Omega exponent value to -1, $v = (j\omega)^{-1}a$, where $a = a_o e^{j\omega t}$ is the acceleration variation.
- If the values in the file are velocity components then the Omega exponent is to be set to zero.

3.3.3. Script

The *Frequency Dependence Type* is defined as a *Script* when the frequency variation of the acoustic variable is given through a script. Predefined variables for frequency, *AngularFreq* or *Frequency*, can be used here. The *AngularFreq* variable is in radians/sec and *Frequency* is in *Hz*. A sample script with normal velocity varying linearly with frequency is shown in Figure 3.7.

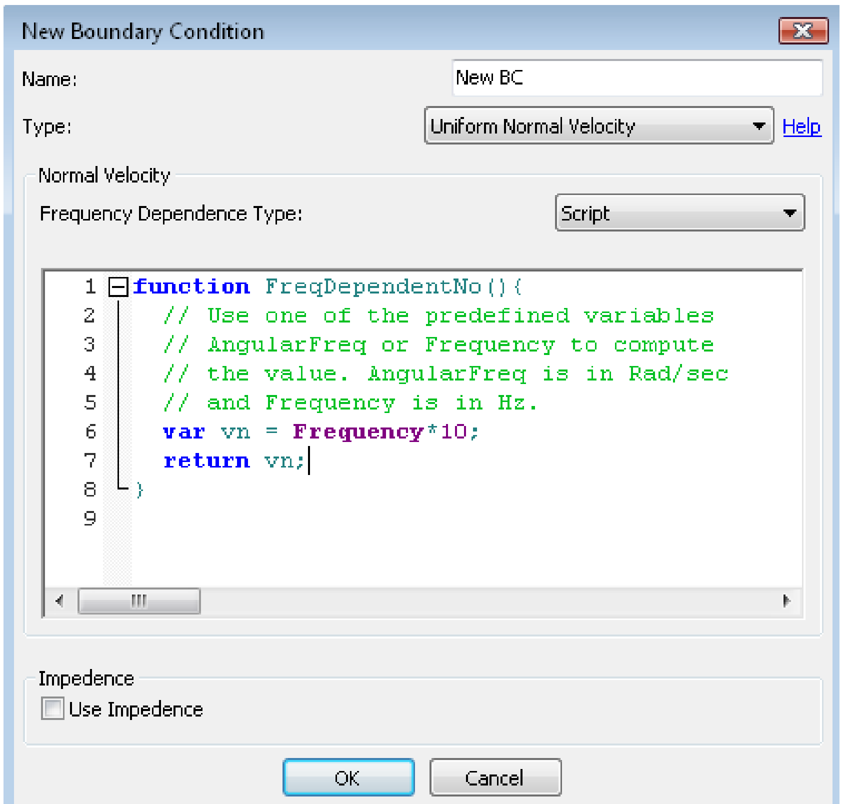


Figure 3.7: Script to define frequency dependent acoustic variable for the boundary condition.

Chapter 4

Getting Started

Coustyx User Interface (UI) assists the user in building an acoustic model by bringing pieces of the model from other files. The user may put the model through several consistency checks, visually inspect it, query objects through dialog boxes, apply boundary conditions, save the model to a file, specify analysis parameters, run analysis and save the results to a file.

In this chapter we will discuss the features provided in *Coustyx* UI and the procedure to be followed in setting up the acoustic problem and running an analysis.

4.1. Main Menu Features

A brief description of items in *Coustyx* UI's main menu bar (Figure 4.1) is given below.

Table 4.1: Description of the main menu bar items in *Coustyx* UI

Menu Items	Description
File	
Open Model File	Open an existing *Coustyx* model (*.cyx files only)
New Model	Create a new *MultiDomain* or *Indirect Coustyx* model
Save Model File	Save all the changes made to a model
Save Model File as	Save the model to a new file
Close	Close the open model

Table 4.1: (continued)

Menu Items	Description
Print Setup	Specify print setup properties. Active only for scripts
Print Preview	Show print preview. Active only for scripts
Print	Print the selected text. Active only for scripts
Recent Files	List the most recently visited model files
Properties	Show the maximum valid frequency for the current acoustic model based on the size of the elements in the mesh
Edit (*Note*: This menu is active only for text in scripts)	
Undo	Undo the last action
Redo	Redo the last action
Cut	Cut the selected text
Copy	Copy the selected text
Paste	Paste the text in clipboard
Select all	Select the entire text in a script
Match braces	Match opening and closing braces in a script
Search Previous	Do backward search for the word typed in the search text box. Search criteria include Case sensitivity search (*Case*), Whole word search (*Word*), and Regular expression search (*RegExp*)
Search Previous Selected	Do backward search for the word selected in the script. Search criteria include Case sensitivity search (*Case*), Whole word search (*Word*), and Regular expression search (*RegExp*)

Table 4.1: (continued)

Menu Items	Description
Search Next	Do forward search for the word typed in the search text box. Search criteria include Case sensitivity search (*Case*), Whole word search (*Word*), and Regular expression search (*RegExp*)
Search Next Selected	Do forward search for the word selected in the script. Search criteria include Case sensitivity search (*Case*), Whole word search (*Word*), and Regular expression search (*RegExp*)
Search	Search
Page	Browse through multi-page window to view *First*, *Previous*, *Next*, and *Last* pages
Analysis	
Run	Run the selected Analysis Sequence
Abort	Abort the current analysis run
Preferences	
Common	(Figure 4.2)
Show splash screen on start up	Enable or disable splash window at the startup of *Coustyx*
Show log messages	Show log messages in the log window
Verbosity Level	This determines how much data that is output while running Coustyx. Verbosity levels, 0=Fatal messages only, 1=Critical errors or warnings, 2=Action notices, 3-5=General information. Set high verbosity level to display verbose output. The verbosity level can only be set when Show log messages is enabled

Table 4.1: (continued)

Menu Items	Description
Save file in binary format	Enable saving *Coustyx* model file (*.cyx) in compressed binary file format. When unchecked the file is saved in *ASCII* (XML) format. It is recommended to save a large *Coustyx* model file in the compressed binary file format
3D Viewer	(Figure 4.3)
Feature angle (deg)	Set the feature angle to be used in the GUI. Feature angle corresponds to the angle between two connected elements, as shown in Figure 4.4. The function Select Connected Elements Through Feature Angle is used to select surface features in a mesh. The default value for the feature angle is set at 15^o. But it can be any value between 0^o and 180^o. Note that the definition is applicable only for 2D elements
Incremental rotation angle (deg)	Set the increments of rotation angle in the GUI
Time for rotation (msec)	Set the speed of the rotation in the GUI
Zoom Increment	Specify the increments at which the zoom takes place in the GUI
Show element edges by default	Show element edges when a structure mesh is opened in the GUI
Element resolution level by default	This option sets the order of the coordinate interpolation used to display the element in the GUI. A resolution level=1 displays the element as a linear element. Higher resolution levels display the element with higher coordinate interpolations. Set level=1 for quick plots
Relative size of nodes	Set the relative size of the nodes displayed in the GUI
Show shadows	Show shadows while displaying a structure mesh in the GUI

Table 4.1: (continued)

Menu Items	Description
Use fast draw mode	Draw the structure mesh faster while using pan, zoom, rotate tools in the GUI. This mode draws only a wire frame of the mesh in the intermediate steps while using these GUI tools
Background Color	Set the background color of the GUI
Help	
Help Content	Show the user manual for help
About	Show the license information

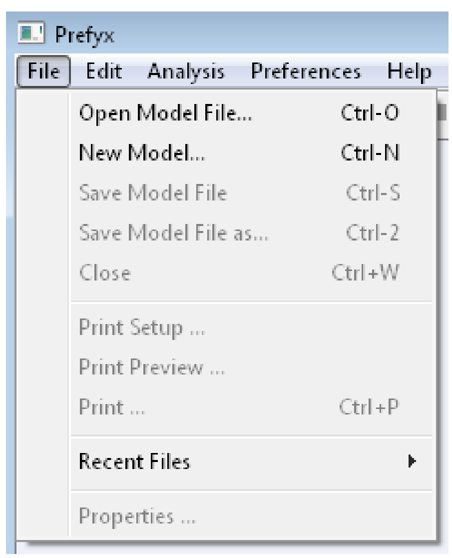


Figure 4.1: Main menu items

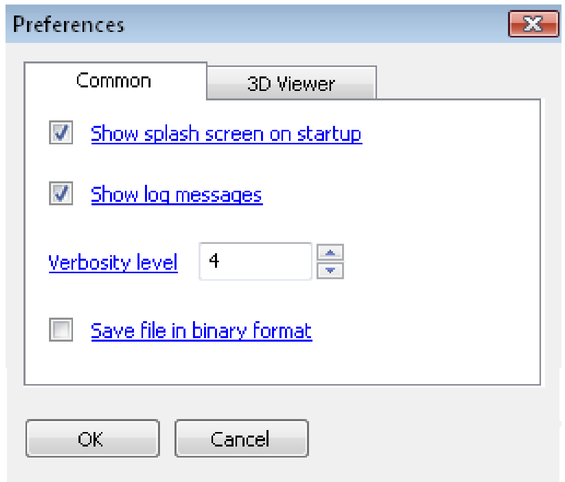


Figure 4.2: Preferences dialog box - Common parameters

4.2. Operations on Mesh Viewer Window

We can view the Finite Element (FE) structure mesh or the Boundary Element (BE) mesh by opening them in a *Mesh Viewer* window. This section describes the user controls and the functions available to manipulate a mesh when it is open in the *Mesh Viewer* window. Note that the *Mesh Viewer* window appears only when a FE or a BE mesh is opened. To open a mesh in the *Mesh Viewer* window, right-click on the desired mesh in the model tree and select **Open**. Figure 4.5 shows *Coustyx* UI with the *Mesh Viewer* window opened for a BE mesh. The *Mesh Viewer* window is made of two panes: the top pane with the Graphical User Interface (GUI) displays a mesh and the bottom pane with the tabbed windows lists various mesh manipulation tasks.

4.2.1. GUI Control Panel Tools

The GUI control panel is located on the top-left corner of the *Mesh Viewer* window (Figure 4.5, Figure 4.6). It has tools to *zoom* and *rotate* the model. Brief descriptions of these GUI tools are in Table 4.2.

Table 4.2: Description of GUI control panel tools

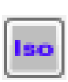
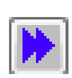
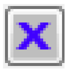
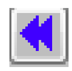
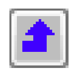
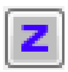
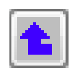
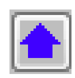
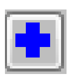
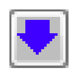
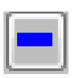
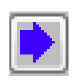
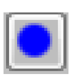
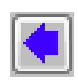
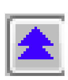
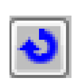
Tool	Used to	Tool	Used to
	show an isometric view of the mesh		rotate the mesh to the right by 90^o

Table 4.2: (continued)

Tool	Usage	Tool	Usage
	show the view perpendicular to the X-axis		rotate the mesh to the left by 90°
	show the view perpendicular to the Y-axis		rotate the mesh CCW by 90°
	show the view perpendicular to the Z-axis		rotate the mesh CW by 90°
	turn the reference axis off or on		rotate the mesh up in increments specified in Preferences menu
	zoom into the mesh		rotate the mesh down in increments specified in Preferences menu
	zoom out of the mesh		rotate the mesh to the right in increments specified in Preferences menu
	fit the mesh in the window		rotate the mesh to the left in increments specified in Preferences menu
	rotate the mesh up by 90°		rotate the mesh CCW in increments specified in Preferences menu
	rotate the mesh down by 90°		rotate the mesh CW in increments specified in Preferences menu
	move and rotate the mesh. This is the default GUI mouse cursor style		perform operations on the selection on the mesh. This cursor style is obtained by holding *shift-key* while the cursor is in the GUI

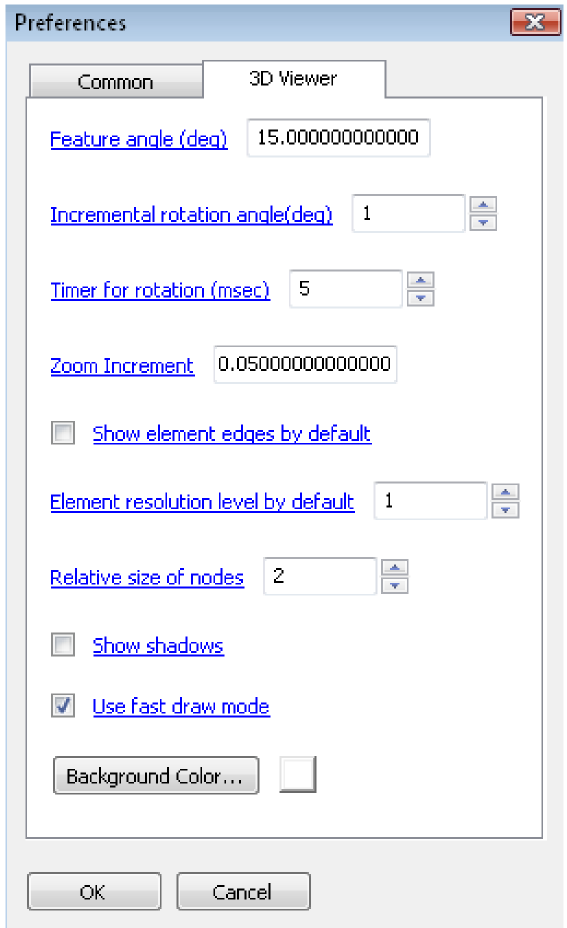


Figure 4.3: Preferences dialog box - 3D Viewer parameters

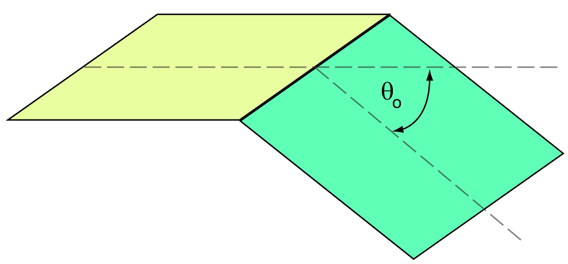


Figure 4.4: Definition of feature angle between two connected elements. Note: This definition is applicable only for 2D elements.

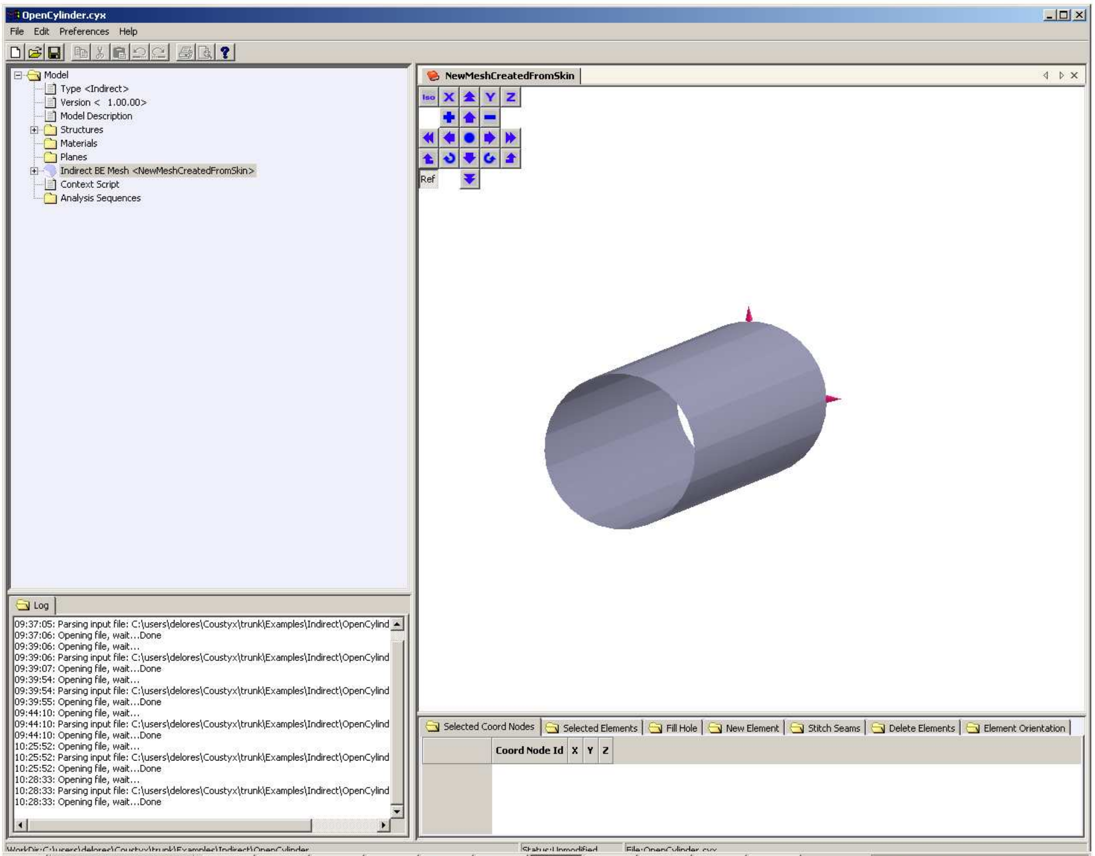


Figure 4.5: Coustyx with Mesh Viewer Window.

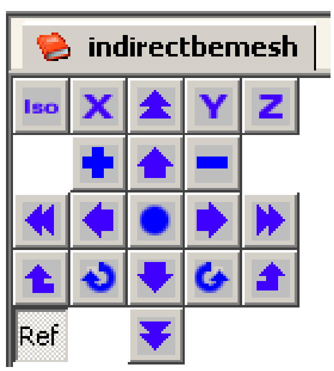


Figure 4.6: GUI Control Panel Tools

4.2.2. Rotate, Pan & Selection Operations

When the mouse cursor is moved into the *Mesh Viewer* window, the cursor style changes to . To rotate the mesh, hold down the left mouse button and drag the mesh with the mouse. To pan, drag while holding down the right mouse button. When the *shift-key* is depressed, the cursor style changes to . When this cursor style is active, the left mouse button may be used to select parts of the mesh. The right mouse button will display a pop-up context menu which lists operations that may be performed on the selection.

4.2.3. Operations on Selection in GUI

Below is a brief description of the operations that may be performed on displayed elements, nodes or faces in the *Mesh Viewer* window. Figure 4.7 shows the context menu that pops up when you right-click any where in the *Mesh Viewer* window while holding down the *shift-key*.

4.2.3.1. Operations on Displayed Elements

Unselect All This un-selects all the elements in the GUI.

Select Elements This lists two different options by which one can select displayed elements in the GUI: **By BC** and **By Set**.

- **By BC** Select elements belonging to any particular boundary condition from the displayed list. Note that this option is available only for boundary element meshes.
- **By Set** Select elements belonging to any particular set from the displayed list.

Select All Displayed Elements This selects all the elements that are displayed in the GUI.

Select All Bad Elements This operation is active only when *Coustyx* finds bad elements in the mesh and throws error messages in the log window mentioning the same. Bad elements are those with bad element connectivities or have other inconsistencies that prevent *Coustyx* to perform its usual tasks. Generally, this set is populated if some elements in the mesh fail the consistency checks done before skinning a FE mesh to get a BE mesh. This operation selects all the bad elements in the mesh. See Table 5.1 for some common types of bad elements.

Selected Elements This sub-menu lists the operations that are performed on selected elements (Figure 4.7). It is activated only elements are selected in the GUI.

- **Unselect** This un-selects the selected elements.
- **Display Style** The user can pick the display style for elements (Figure 4.8).
  - **Display** This option when selected displays the element. The selected element will be hidden if this option is turned off.
  - **Show Faces** Display element faces.
  - **Show Edges** Display element edges.

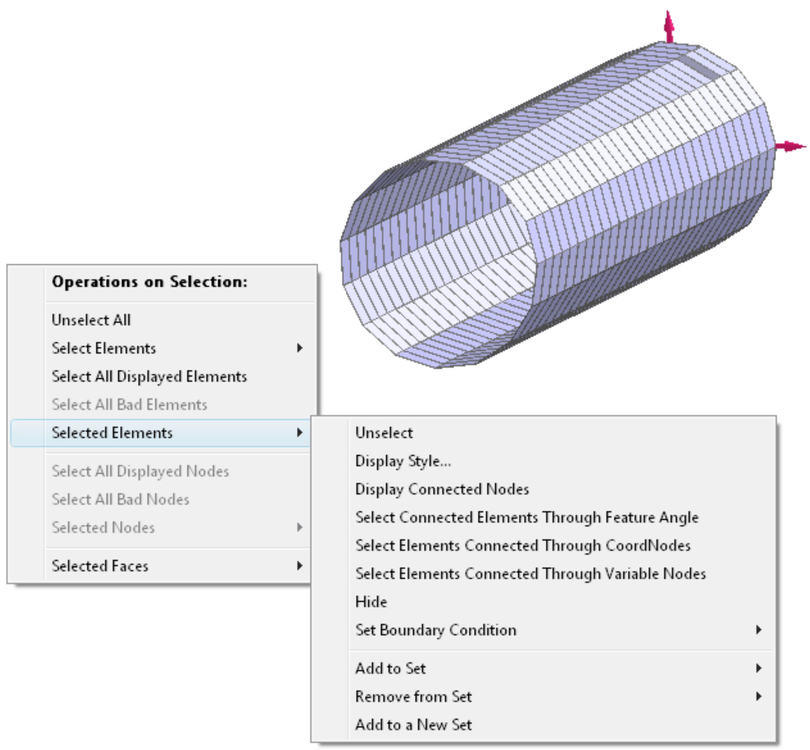


Figure 4.7: GUI operations on elements available through the context menu, activated by the right mouse button with the *shift-key* held down.

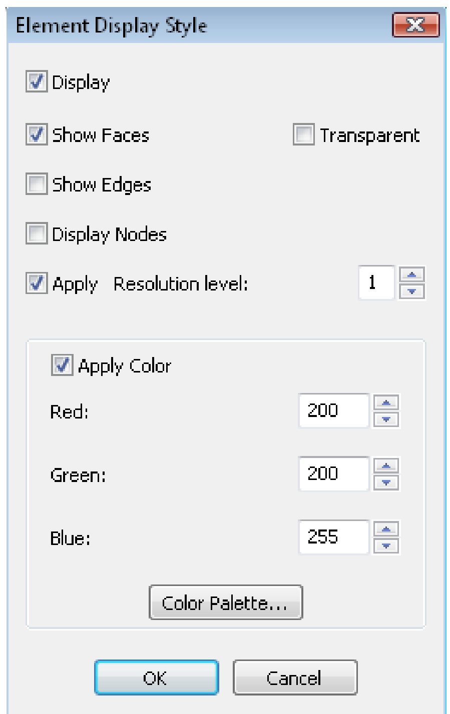


Figure 4.8: Element Display Style Window.

Display Nodes This option when selected displays the coordinate nodes of the selected elements.

Apply Resolution level This option controls the order of the coordinate interpolation used to display the element. A resolution level of one when applied displays the element as a linear element. Higher resolution levels display the element with higher coordination interpolations.

Transparent This option makes the selected element transparent.

Apply Color The color of the element face displayed can be changed using this option. The user can select the color by using the color palette provided, or by changing the composition of red, green and blue. If this checkbox is unchecked, the colors are not changed.

Display Connected Nodes Displays all the coordinate nodes connected to the selected element.

Select Connected Elements Through Feature Angle This option is used to select all the 2D elements that are connected to the current element(s) in the GUI through feature angle(s) less than the specified feature angle in **Preferences—3D Viewer** (Figure 4.3). Refer to Figure 4.4 for the definition of feature angle between two connected elements. Note that this function is applicable only for 2D elements.

Select Elements Connected Through Coord Nodes This option is used to select all the elements that are connected to the current element through coordinate nodes. This can be useful when the mesh consists of several disconnected pieces, and one contiguous set of elements are needed for selection.

Select Elements Connected Through Variable Nodes This option is used to select all the elements that are connected to the current element through variable nodes. The variable node considered for the *MultiDomain* model is **P** (pressure) node and for the *Indirect* model is μ (pressure jumps) node. This function is especially useful to identify pressure jumps along junction constraints in *Indirect* models. Note that this function is available only for BE meshes.

Hide This option is used to hide selected elements.

Set Boundary Condition Boundary conditions are defined with unique names in *Coustyx*. A boundary condition can be applied to selected elements by setting the boundary condition to the unique name defined. Refer to Chapter 6 on how to define boundary conditions.

Add to Set This option allows user to add selected elements to a pre-defined Set. Refer to Section 4.4 on how to define a Set. Elements or nodes are grouped together to form a Set. Operations on a Set is propagated to all its components. For example, it is easy to apply or change boundary conditions on the Set than on each element in the Set.

Remove from Set This option is used to remove selected elements from their current Sets. Refer to Section 4.4 for more information on Sets.

Add to a New Set This option allows user to add selected elements to a new Set. A new set with a default name **New Set** or **New Set i** (i is a number) is created. Refer to Section 4.4 on how to rename the Set.

4.2.3.2. Operations on Displayed Nodes

To use operations on displayed nodes, first make nodes visible in the GUI.

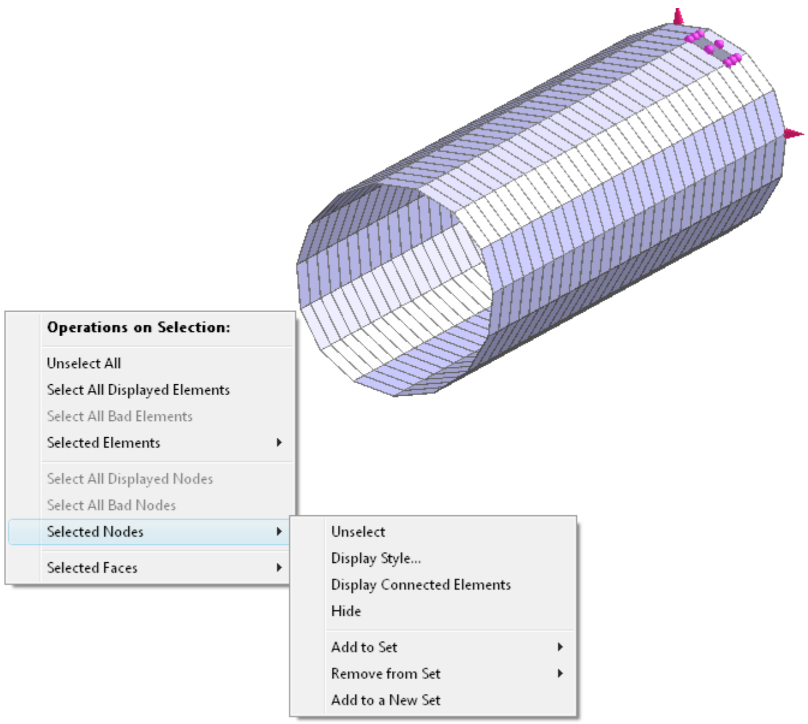


Figure 4.9: GUI operations available for selected nodes.

Select All Displayed Nodes This selects all the nodes that are displayed in the GUI.

Select All Bad Nodes This selects all the Bad nodes identified by *Coustyx*. This operation is active only when *Coustyx* finds Bad nodes in the mesh and throws error messages in the log window mentioning the same. Bad nodes are those nodes with incorrect solid angles or have some inconsistencies that prevent *Coustyx* to perform its usual tasks. Generally, this set is populated if some nodes in the mesh fail the consistency checks done before skinning a FE mesh to get a BE mesh. This operation selects all the bad nodes in the mesh.

Selected Nodes This sub-menu lists the operations that are performed on selected nodes (Figure 4.9). It is activated only when nodes are selected in the GUI.

Unselect This un-selects the selected nodes.

Display Style The user can pick the display style for nodes (Figure 4.10).

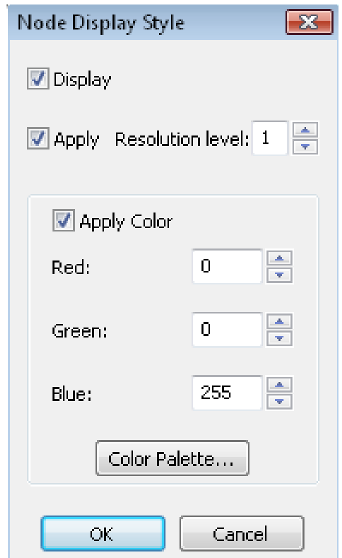


Figure 4.10: Nodes Display Style Window.

Display This option when selected displays the nodes. The selected node will be hidden if this option is turned off.

Apply Resolution level This option controls the resolution of the sphere used to display the node.

Apply Color The color of the element face displayed can be changed using this option. The user can select the color by using the color palette provided, or by changing the composition of red, green and blue. If this checkbox is unchecked, the colors are not changed.

Display Connected Elements Displays all the elements connected to the selected coordinate node.

Hide This option is used to hide selected nodes.

Add to Set This option allows user to add selected nodes to a pre-defined Set. Refer to Section 4.4 on how to define a Set.

Remove from Set This option is used to remove selected nodes from their current Sets. Refer to Section 4.4 for more information on Sets.

Add to a New Set This option allows user to add selected nodes to a new Set. A new set with a default name **New Set** or **New Set i** (i is a number) is created. Refer to Section 4.4 on how to rename the Set.

4.2.3.3. Operations on Displayed Faces

Figure 4.11 shows the menu of operations available on faces. Operations on faces are activated only when faces are selected in the GUI.

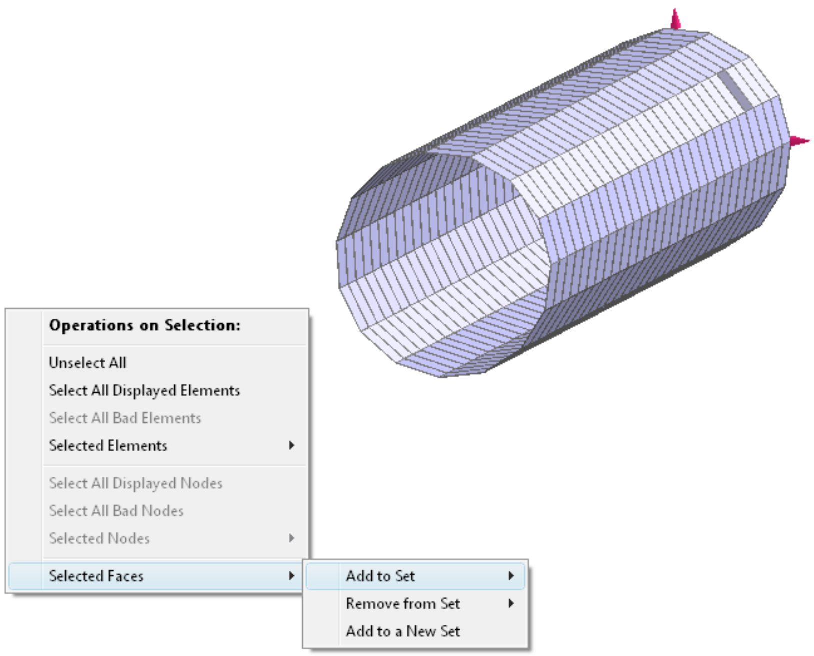


Figure 4.11: GUI operations on faces available through the context menu, activated by the right mouse button with the *shift-key* held down.

Selected Faces This lists the operations that are performed only on the selected faces.

> **Add to Set** This option allows user to add selected faces to a pre-defined Set. Refer to Section 4.4 on how to define a Set.
>
> **Remove from Set** This option is used to remove selected faces from their current Sets. Refer to Section 4.4 for more information on Sets.
>
> **Add to a New Set** This option allows user to add selected faces to a new Set. A new set with a default name **New Set** or **New Set i** (i is a number) is created. Refer to Section 4.4 on how to rename the Set.

4.2.4. Selected Coord Nodes

This tabbed window is part of the bottom pane of the *Mesh Viewer* window (Figure 4.5). *This window is read-only.* The window contains a table that shows the id and the coordinates of the selected nodes in the mesh viewer.

4.2.5. Selected Elements

This tabbed window is part of the bottom pane of the *Mesh Viewer* window (Figure 4.5). *This window is read-only.* The window contains a table that shows properties of the selected elements in the mesh viewer. The details presented in the table vary with the type of mesh, FE or BE mesh, and also vary with the type of the model, *MultiDomain* or Indirect BE model. The possible columns in the table are:

Index The indexes of selected elements. An element index can only vary from one to the total number of elements.

Elem ID The ids of selected elements. An element id can be any unique positive number.

B.C. The boundary conditions applied on selected elements are displayed here.

Selected Faces The face number of the selected face of the element. Face 1 refers to the face on the positive side of the element normal, and Face 2 refers to the face on the negative side of the normal.

Type For a BE mesh, only two type of elements are considered: *Triangle* and *Quadrilateral*.

Coord Conn Type The coordinate connectivity type describes the type of interpolation scheme used to define the geometry on an element. This can be *linear*, *quadratic* or *cubic*. This detail is displayed only for BE meshes.

Coord Nodes The list of coordinate nodes used to define selected elements.

Var Conn Type The variable connectivity type describes the interpolation scheme used for acoustic variables on an element. The variable connectivity can be *linear*, *quadratic* or *cubic*. This detail is displayed only for BE meshes. *Coustyx* allows independent interpolation schemes for coordinates and acoustic variables. This increases the flexibility in modeling. For example, consider a uniformly pulsating sphere modeled for acoustic analysis. The sphere geometry should be modeled with *quadratic* connectivity to accurately model the surface. But we don't need to do the same for variable connectivity, as the pressure is constant over a pulsating sphere. Hence, a simple *linear* variable connectivity should result in good accuracy. This kind of flexibility decreases the analysis run times without reducing the accuracy.

Baffled (*Only* for *Indirect* BE meshes). This shows whether the selected element lies on a baffle plane or not. For a model with baffled planes the acoustic variables include normal derivative of pressures on both sides in addition to the single layer (σ) and double layer (μ) potentials).

P Nodes (*Only* for *MultiDomain* BE meshes). This displays a list of pressure nodes (p nodes) used in *MultiDomain* models.

Pn Nodes (*Only* for *MultiDomain* BE meshes). This displays a list of normal derivative of pressure nodes (*pn* nodes) used in *MultiDomain* models.

Pn Plus Nodes (Used in *Indirect* BE meshes with baffled elements). This displays the normal derivative of pressure on the positive side of an element (*pn plus*) in Indirect BE elements.

Pn Minus Nodes (*Only* for *Indirect* BE meshes with elements on baffled plane). This displays the normal derivative of pressure on the negative side of an element (*pn minus*) in Indirect BE elements.

Sigma Nodes (*Only* for *Indirect* BE meshes). This displays single layer potential (σ) nodes used in defining the variable connectivity of the element. The single layer potential at a point is defined as the difference between the normal derivative of pressures on the positive (p_n^+) and the negative (p_n^-) sides of an element, that is, $\sigma = p_n^+ - p_n^-$.

Mu Nodes (*Only* for *Indirect* BE meshes). This displays double layer potential (μ) nodes used in defining the variable connectivity of the element. The double layer potential is defined as the difference between the pressures on the positive (p^+) and the negative (p^-) sides of an element, that is, $\mu = p^+ - p^-$. Each μ node is displayed with a positive or a negative sign. These signs are defined to accommodate BE meshes with adjacent elements not oriented consistently. Refer to Section 4.3.1.2 for more details.

4.2.6. Fill Hole

This tabbed window is located at the bottom pane of the *Mesh Viewer* window. This is used to fill holes in both FE and BE meshes. *Coustyx* uses *optimized delaunay triangulation method* to fill holes with two-dimensional triangle elements. This method needs a closed path to fill a hole (Figure 4.12). A detailed discussion on how to fill holes is provided in Section 5.4.

4.2.7. Skin

This tabbed window located at the bottom of the *Mesh Viewer* window is used to generate a 2-D surface mesh (BE mesh) by skinning the 3-D FE mesh. Seams are defined to stop the propagation of skin to unwanted regions of the mesh. This function appears only for FE meshes. More details on how to skin a structure mesh are discussed in Section 5.5. (Figure 4.13).

4.2.8. New Element

This tabbed window located at the bottom of the *Mesh Viewer* window is used to fill gaps in the mesh by creating new surface elements. This function is especially useful to create new elements for cases where the functions Fill Hole and Stitch Seams can not be used. A new element is added by specifying the coordinate node ids and the type of the element to be created. Refer to Section 5.6 for more details.

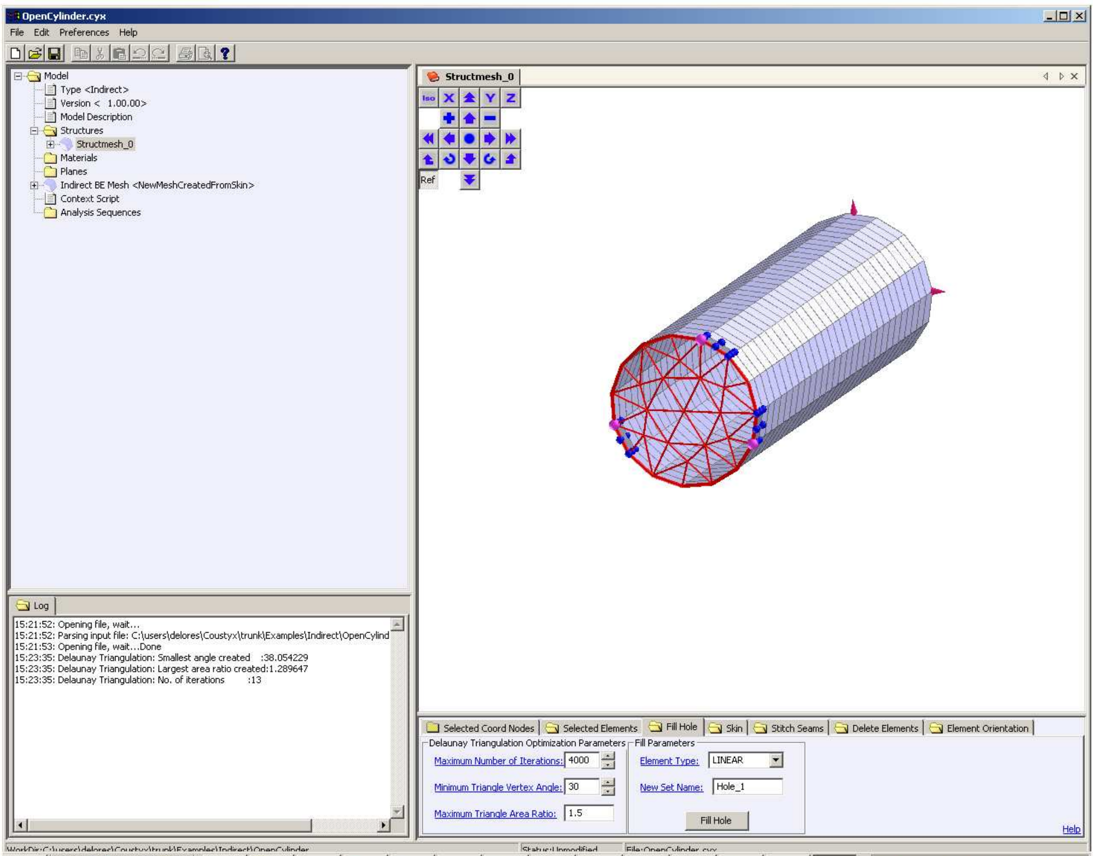


Figure 4.12: Fill Hole

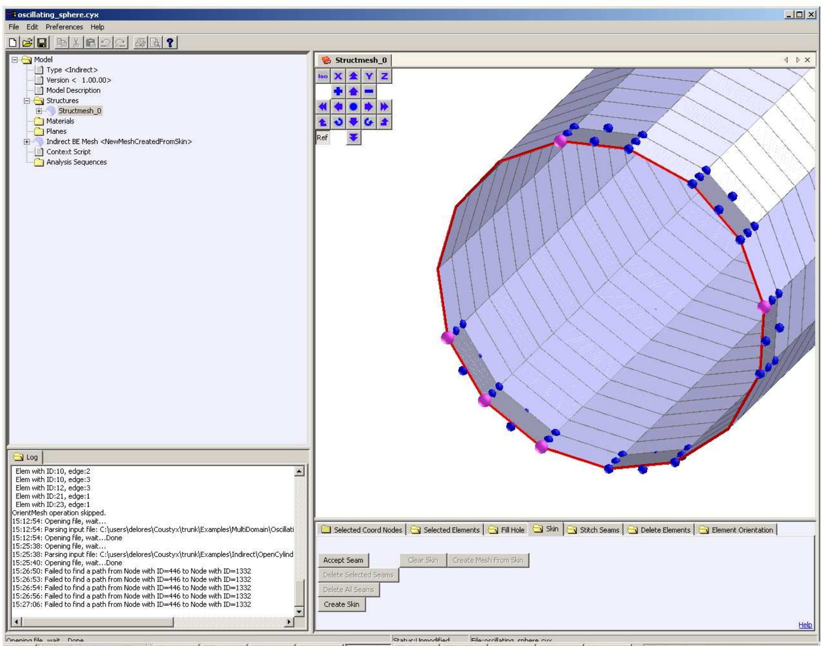


Figure 4.13: Create Skin

4.2.9. Stitch Seams

This tabbed window located at the bottom of the *Mesh Viewer* window is used to fill gaps between disjoint parts of a mesh (FE or BE) by generating new triangle elements between two seams (Figure 4.14). The seams are created by selecting coordinate nodes. More details on how to create and stitch seams are discussed in Section 5.7.

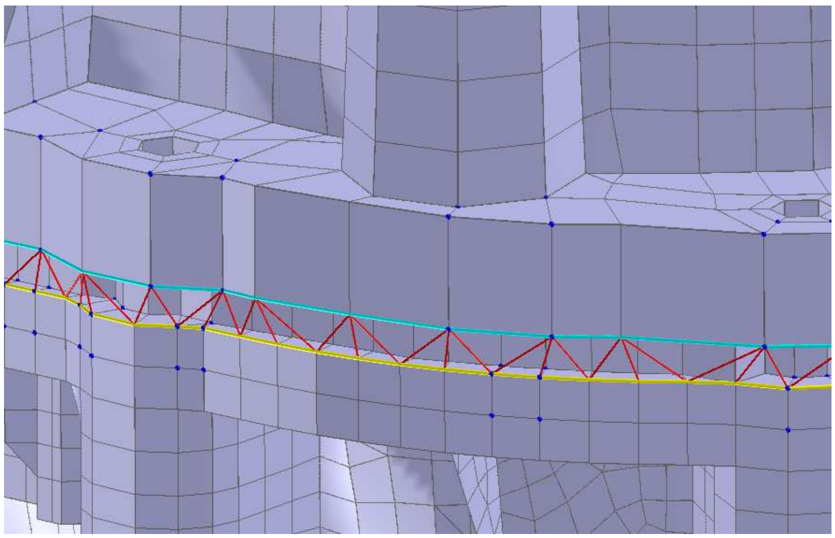


Figure 4.14: Stitch Seams

4.2.10. Delete Elements

This tabbed window located at the bottom of the *Mesh Viewer* window is used to delete selected elements in the GUI by pressing the **Delete Selected Elements** button. *Coustyx* provides options to remove unshared coordinate nodes and unshared variable nodes of the deleted element. Refer to Section 5.8 for more details.

4.2.11. Merge Nodes

This tabbed window located at the bottom of the *Mesh Viewer* window is used to merge selected nodes with all coincident nodes in the GUI. Refer to Section 5.9 for more details.

4.2.12. Split Pn Nodes / Split Sigma Nodes

This tabbed window located at the bottom of the *Mesh Viewer* window is used to split selected Pn (or Sigma) nodes in a MultiDomain BE mesh (or Indirect BE mesh). Use this function only to split nodes along the common edge of adjacent elements which have different velocity boundary conditions but no duplicate nodes. During skinning, Coustyx automatically creates duplicate nodes for most other cases, such as geometry discontinuities, edges, corners, junctions etc. Refer to Section 5.10 for more details.

4.2.13. Element Orientation

This tabbed window located at the bottom of the *Mesh Viewer* window is used to view and/or flip the normals of the selected elements in the GUI. Refer to Section 5.11 for more details.

4.3. Model Setup

In this section we will briefly discuss the procedure one needs to follow to set up the *Coustyx* model and run the acoustic analysis. We recommend that these steps be followed in the order described below. To start using the software, open *Coustyx* from the start menu or from a shortcut on your desktop.

Coustyx window shows a *tree* structure named **Model** at the top-most level and several branch members fulfilling specific tasks in building the model or running the analysis. The model tree appears at the top left portion of the *Coustyx* window when a model is open or when a new model is created from the *File* menu. Figure 4.15 shows the structure of the **Model** tree for *MultiDomain* and *Indirect Coustyx* models.

A brief description on each of the model tree member tasks are provided below.

4.3.1. New Model/Open Model File

To create a new model from the *Coustyx* Main Menu select: **File → New Model** (Figure 4.16). Choose the model type from the options: **MultiDomain** or **Indirect**. Also, choose **Model Units** from the selection. See Section 3.2.2 for more details on model units. Press OK to accept selections and create a new model.

To open an existing model select: **File → Open Model File**.

4.3.1.1. MultiDomain Model

A *MultiDomain* model is created to solve the acoustic problem using *Direct Boundary Integral Formulation*. The primary variables in the direct formulation are *pressure* (p) and *normal derivative of pressure* (p_n).

The *normal derivative of pressure* (p_n) and *normal velocity* (v_n) at a point are related as follows:

$$p_n = i\rho_o\omega v_n$$

where ρ_o is the ambient density, ω is the frequency.

This formulation allows several bounded *interior* domains and/or one unbounded *exterior* domain. It requires all the elements of the BE mesh to be oriented consistently. The domain can be either on the positive side of the mesh or on the negative side of the mesh at a time, but not both. Each boundary element domain can have a different set of material (fluid) properties. Table 4.3 shows some of the important differences between *MultiDomain* and *Indirect* models.

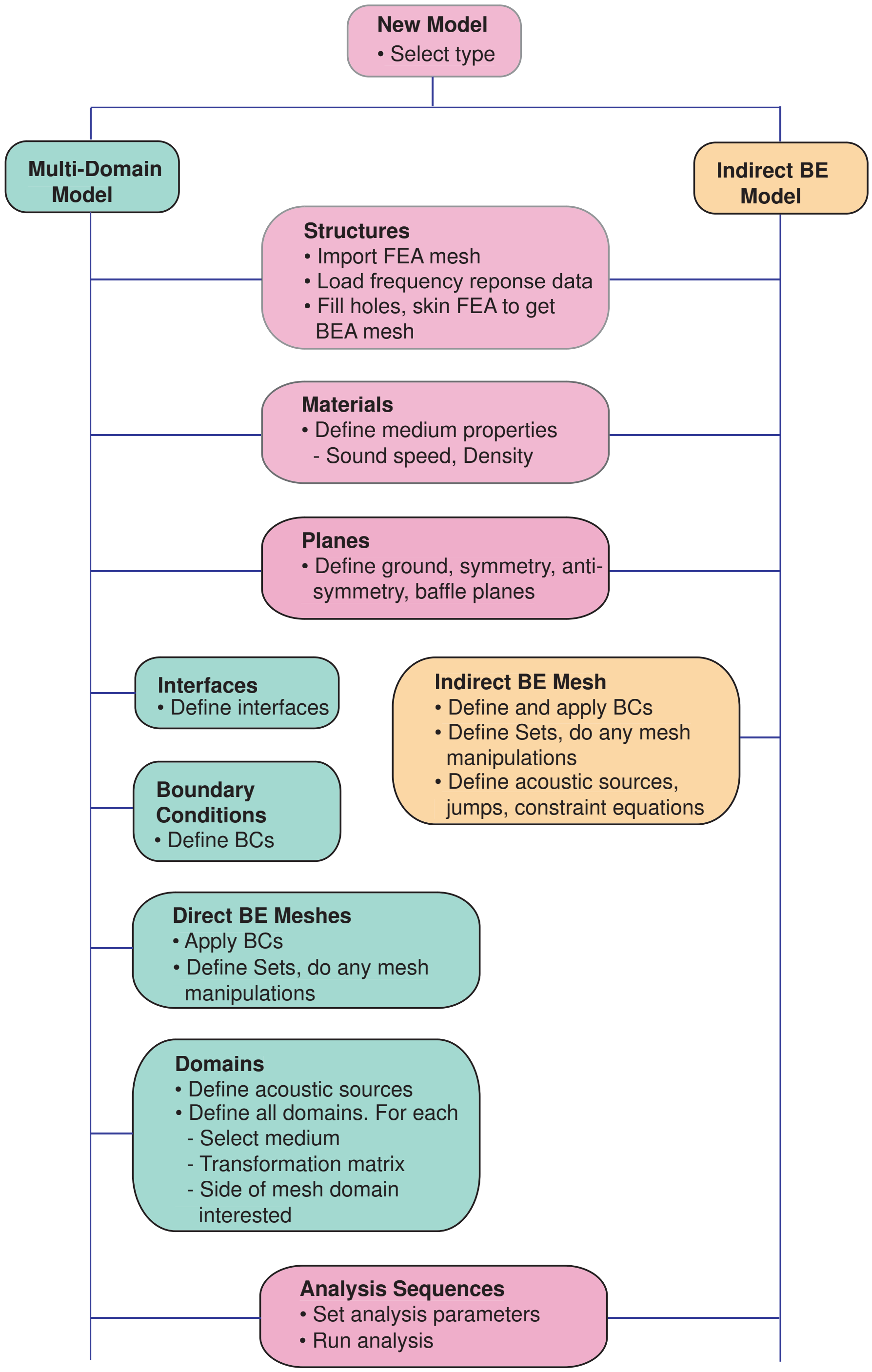


Figure 4.15: Coustyx model tree structure.

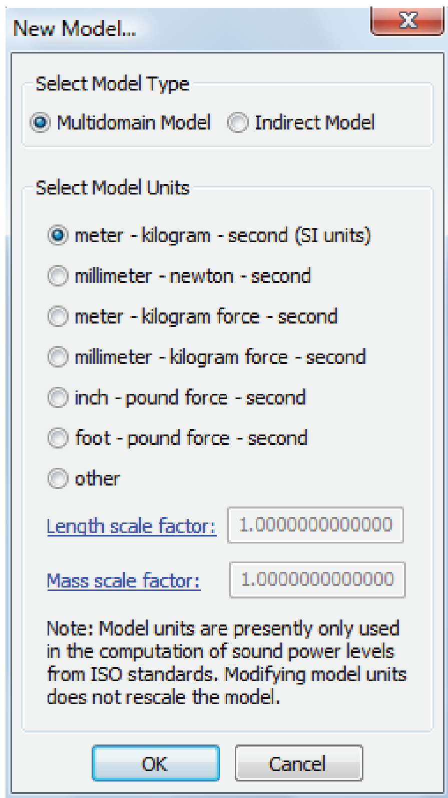


Figure 4.16: New model dialog box.

4.3.1.2. Indirect Model

An *Indirect* model is created to solve the acoustic problem using the *Indirect Boundary Integral Formulation*. The primary variables used in this formulation are *pressure jumps* or *pressure discontinuities* and *velocity jumps* or *velocity discontinuities* across the boundary. These are also known as *double layer potentials* (μ) and *single layer potentials* (σ) respectively.

The single layer potential (σ) is the difference between the normal derivative of pressures on positive (p_n^+) and negative (p_n^-) sides of an element, that is,

$$\sigma = p_n^+ - p_n^-$$

The double layer potential (μ) is the difference between the pressures on the positive (p^+) and negative (p^-) sides of an element , that is,

$$\mu = p^+ - p^-$$

This formulation solves both the upper and lower sides of the boundary simultaneously. For a closed mesh, it solves both the exterior and interior domains at once. Note that only one mesh is allowed in this model type. Table 4.3 shows some of the important differences between *MultiDomain* and *Indirect* models.

Variable Node Concept The concept of *Variable node* is introduced in *Indirect* model to allow greater flexibility in the modeling of an acoustic problem. The concept of *Variable node* relates the value of a variable (μ) at a geometry node in an Element i to its variable node through a positive or a negative sign (Figure 4.17). Each μ node in an element is assigned with a positive or a negative sign to allow meshes with adjacent elements oriented inconsistently. This adds the flexibility to even handle one-sided surfaces such as *Mobius strip*. The sign coefficient for the first element at the node is chosen arbitrarily. The sign coefficients for the remaining elements at a geometry node are determined uniquely based on their orientations with respect to the first element. *Coustyx* automatically creates these nodes with appropriate signs when it skins the structure mesh.

$$\mu_{\mathrm{i}}(\text{Within Element}) = \text{sign coefficient} \bullet \mu(\text{Variable Node}) \tag{4.1}$$

This concept is illustrated using Figure 4.18, for two elements, element i and element j oriented in opposite directions. Local node 4 of element i and local node 1 of element j share the same coordinate location. Also they share the same variable nodes μ_{vn} and σ_{vn}.

$$\mu_{\mathrm{i}}(4) = p_T - p_B = \mu_{\mathrm{vn}}$$

$$\sigma_{\mathrm{i}}(4) = \frac{\partial p_T}{\partial n_{\mathrm{i}}} - \frac{\partial p_B}{\partial n_{\mathrm{i}}} = \frac{\partial p_T}{\partial z} - \frac{\partial p_B}{\partial z} = \sigma_{\mathrm{vn}}$$

Element j has the opposite orientation as element i, since the common edge PQ is traversed in the same sense in both element i and element j.

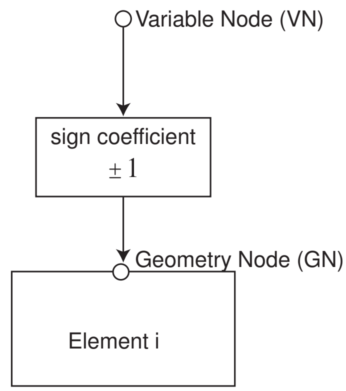


Figure 4.17: Variable Nodes in Indirect BEM

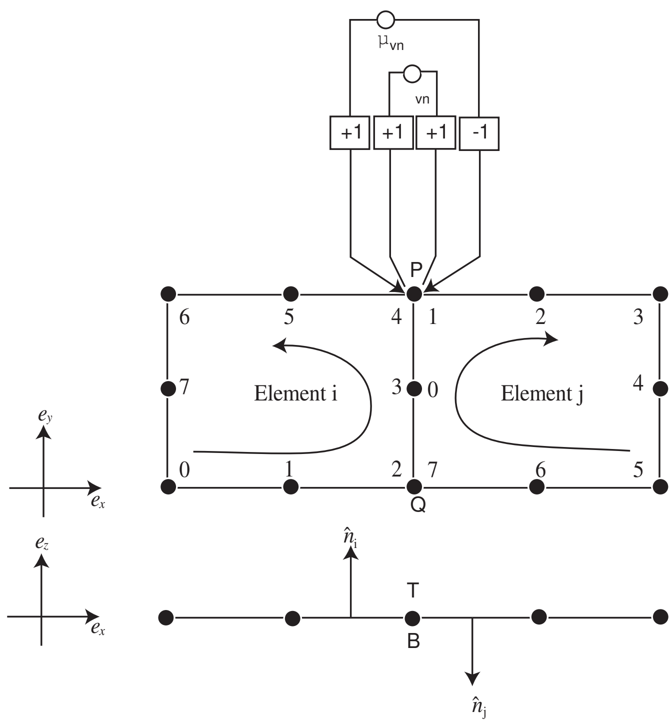


Figure 4.18: Variable Nodes in Indirect BEM

$$\mu_{\mathrm{j}}(1) = p_B - p_T = -\mu_{\mathrm{vn}}$$

$$\sigma_{\mathrm{j}}(1) = \frac{\partial p_B}{\partial n_{\mathrm{j}}} - \frac{\partial p_T}{\partial n_{\mathrm{j}}} = -\frac{\partial p_B}{\partial z} + \frac{\partial p_T}{\partial z} = \sigma_{\mathrm{vn}}$$

The sign coefficient for $\mu_{\mathrm{j}}(1)$ is -1, negative of the sign coefficient for $\mu_{\mathrm{i}}(4)$.

$$\mu_{\mathrm{i}}(4) = \mu_{\mathrm{vn}}$$
$$\mu_{\mathrm{j}}(1) = -\mu_{\mathrm{vn}}$$

However, the sign coefficient for $\sigma_{\mathrm{j}}(1)$ is equal to the sign coefficient for $\sigma_{\mathrm{i}}(4)$. This is true in both cases – if Element j has the same orientation as Element i, if or Element j has the *opposite* orientation as Element i.

$$\sigma_{\mathrm{i}}(4) = \sigma_{\mathrm{vn}}$$
$$\sigma_{\mathrm{j}}(1) = \sigma_{\mathrm{vn}}$$

Table 4.3: Differences between MultiDomain and Indirect models

No.	MultiDomain	Indirect
1	Uses Direct Boundary Integral Formulation.	Uses Indirect Boundary Integral Formulation.
2	Allows several interior domains and one exterior domain. Each mesh should enclose a volume.	Allows only one domain. It solves the problem on both sides of the boundary simultaneously. The mesh does not need to enclose a volume.
3	Allows multiple meshes in a single model.	Allows only one mesh.
4	Primary variables in the formulation are pressure (p) and normal derivative of pressure (p_n) on the surface.	Primary variables in the formulation are single layer (σ) and double layer (μ) potentials, $\sigma = p_n^+ - p_n^-$, $\mu = p^+ - p^-$, where $+$ is for variables on the positive side of the element, $-$ is on the negative side.
5	The surface pressures (p) and normal velocities (v_n) are directly obtained from the solution.	The surface pressures and normal velocities are derived from the single layer (σ) and double layer (μ) potentials on the surface.

Table 4.3: (continued)

No.	MultiDomain	Indirect
6	Can't model ribs or any two-dimensional panels which don't enclose a volume. These are automatically removed during skinning.	Allows modeling of ribs or any two-dimensional panels which don't enclose a volume. These are automatically generated during skinning.
7	Can't model acoustic problems with pressure jumps at the boundary.	Allows modeling of acoustic problems with pressure jumps. For example, can model a *2-D* circular disk with zero pressure jumps at the edges.

4.3.2. Structures

Select: **Model** → **Structures**.

Structures is a model tree member which is used to import a FE mesh, load frequency response data and natural mode data. The mesh imported here is the main geometry source for generating BE mesh for acoustic analysis. The imported FE mesh can be manipulated using *Manipulation Task Functions* and then skinned to generate the BE mesh (Refer to Section 5.5).

4.3.3. Materials

Select: **Model** → **Materials**.

Materials is the model tree member used to define the **Speed of Sound** and **Ambient Density** of the fluid medium surrounding the meshes in the model. These properties are defined through selecting any of the frequency dependent types: *Constant*, *Table*, or *Script*. Any other relevant property is derived from these two.

- **Define a New Material** Right-click on **Materials** and select: **New** and proceed with entering new parameters information. Click OK to accept. See Figure 4.19 and Figure 4.20.

- **Edit Materials** Select: **Materials** → <*Material Name*>. Right-click on <*Material Name*> and select: **Edit**. Proceed with editing the parameters. Click OK to accept (Figure 4.20).

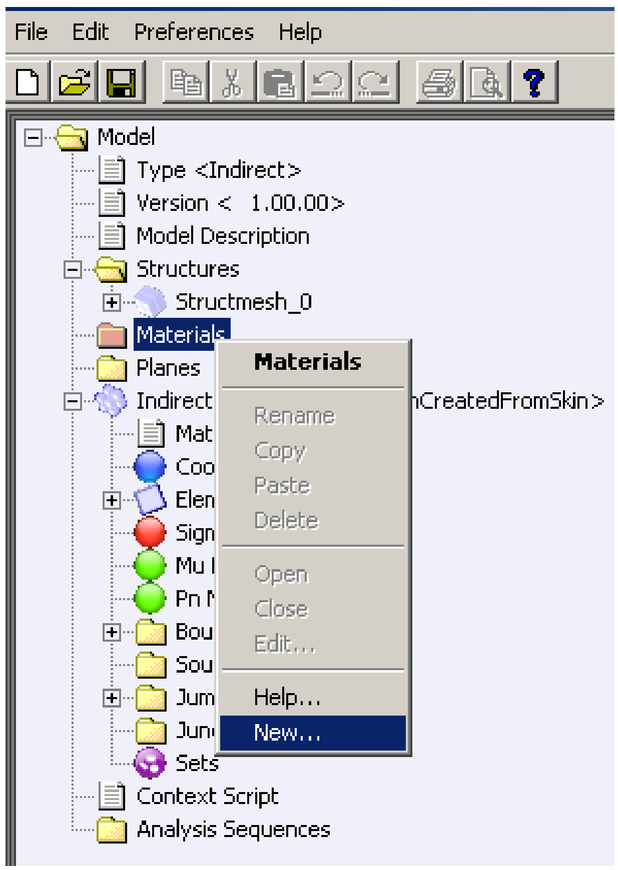


Figure 4.19: Create New Material

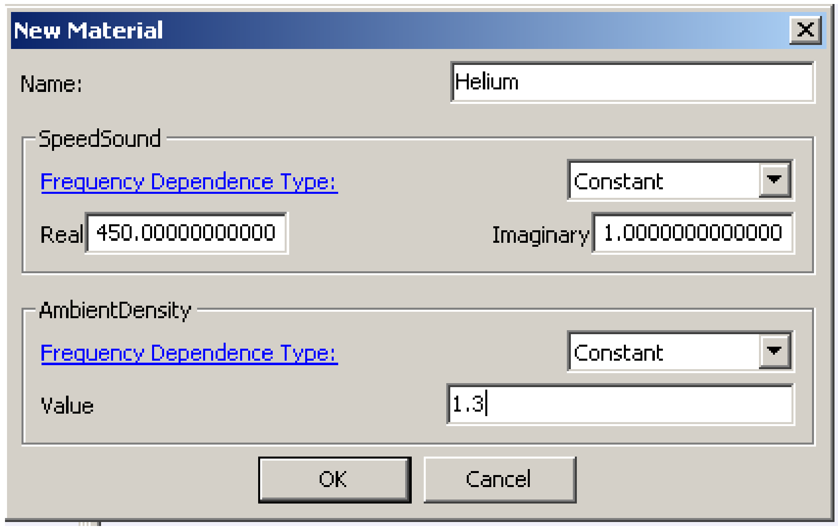


Figure 4.20: Edit Material Properties

4.3.3.1. Speed of Sound

The value for the speed of sound (c) in a fluid medium is defined here. For most of the cases the speed of sound is defined as a purely constant real value. But it can also be defined as a frequency dependent complex value, $c = c_r + jc_i$, to introduce damping in the system. The variation with frequency is simulated by defining the speed of sound to be a frequency dependent type: *Table* or *Script*.

Note that for a decaying wave the imaginary part of the speed of sound should always be negative, that is $c_i < 0$ (Refer to Figure 4.21). This is due to the adoption of the following convention in *Coustyx*,

$$P = \mathbf{Re}\left(pe^{-j\omega t}\right)$$

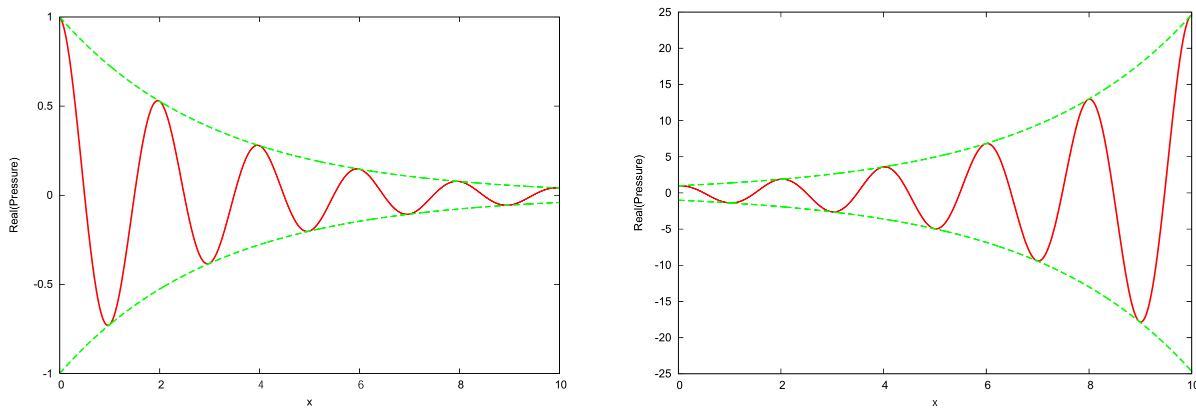
where **Re** stands for "real part of", P is the time-harmonic pressure wave, p is the complex amplitude of the sound pressure, ω is the frequency of fluctuation, and $j = \sqrt{(-1)}$.

Consider a point source in a medium with the speed of sound $c = c_r + jc_i$. The outgoing wave has the form

$$p = p_o e^{jkr} = p_o e^{Rc_i r} e^{jRc_r r}$$

where $k = \omega/(c_r + jc_i)$ is the wave number, $R = \omega/(c_r^2 + c_i^2)$ is purely real, p_o is real. If $c_i < 0$, the pressure p is bounded with an exponential decay in the amplitude as shown Figure 4.21, where as for $c_i > 0$ the pressure p exponentially grows which is physically improbable.

Note that the units used for speed of sound along with the units for ambient density determine *Coustyx* model units. For more information on the unit conventions followed in *Coustyx* refer



(a) Decaying (bounded) pressure with $c_i < 0$

(b) Unbounded pressure with $c_i > 0$

Figure 4.21: Effect of complex speed of sound, $c = c_r + jc_i$, on the pressure variation with distance from a point source.

to Chapter 3.

4.3.3.2. Ambient Density

The value for the ambient density (ρ_o) of the fluid medium is defined here. Ambient density can be defined as a frequency dependent real value. The variation with frequency can be introduced by defining the ambient density to be a frequency dependent type: *Table* or *Script*.

Note that the units used for ambient density along with the units for speed of sound determine *Coustyx* model units. For more information on the unit conventions followed in *Coustyx* refer to Chapter 3.

4.3.4. Planes

Select: **Model** → **Planes**.

Planes is the model tree member used to define ground or symmetry planes. Symmetry planes are defined to exploit the symmetry of a structure geometry and its boundary conditions by modeling only a portion of the full model. This reduces the size of the problem significantly and also results in faster analysis. *Coustyx* provides options to define four different types of planes: *Symmetry*, *Anti-symmetry*, *Ground* and *Baffle*. A plane is uniquely defined by a point on the plane and its normal vector. A maximum of three planes from symmetry or anti-symmetry plane types can be defined per model. When multiple planes (other than baffled plane) are defined, they must be orthogonal to each other.

- **Define a New Plane** Right-click on **Planes** and select: **New** and proceed with entering new parameters information. Click OK to accept.

- **Edit Planes** Select: **Planes** → <*Plane Name*>. Right-click on <*Plane Name*> and select: **Edit**. Proceed with editing the parameters. Click OK to accept.

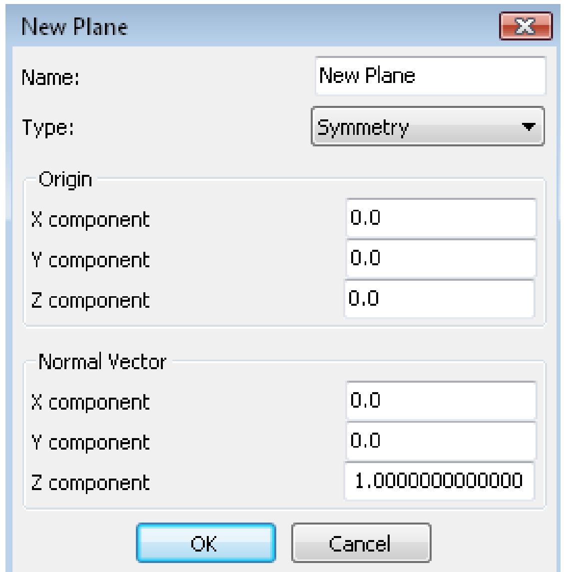


Figure 4.22: Creating a new plane.

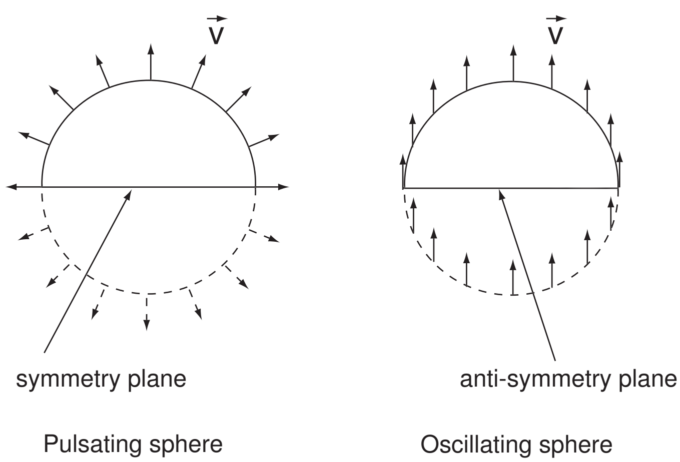


Figure 4.23: Descriptions for different types of planes. Note arrows represent velocity vectors at those points.

Name The name of the new plane can be entered here.

Type The different types of planes that can be defined in *Coustyx* are

- **Symmetry** A symmetry plane is defined when both the geometry and boundary conditions are symmetric with respect to the plane. Example: Consider a sphere model with uniform radial velocity (pulsating sphere). The sphere geometry and the radial velocity (boundary condition) are symmetric with respect to any plane passing through the center of the sphere. Figure 4.23 shows a sphere with the velocity vectors plotted by arrows. The size of this problem could be halved by defining a symmetry plane and modeling only half of the sphere geometry.
- **Anti-symmetry** An anti-symmetry plane is defined when the geometry is symmetric and the boundary conditions are anti-symmetric with respect to the plane. Example: Consider a sphere model with axial velocity (oscillating sphere). The sphere geometry is symmetric and the axial velocity (boundary condition) is anti-symmetric with respect to any plane perpendicular to the axial direction and passing through the center of the sphere. Figure 4.23 shows a sphere with the velocity vectors plotted by arrows. The size of this problem could be halved by defining an anti-symmetry plane and modeling only half of the sphere geometry.
- **Ground** A ground plane is defined when the reflection from the ground needs to be taken into account. The ground plane defined in *Coustyx* is a perfect reflector.
- **Baffle** Infinite acoustic baffles can be defined by this plane. This feature is not implemented in the current version and will be available in future versions of *Coustyx*. Do not use this feature for now.

Origin The point through which the plane passes. The coordinates of the point are input in **X component**, **Y component**, and **Z component**.

Normal Vector The normal vector of the plane is described here. The components of the normal vector are input in **X component**, **Y component** and **Z component**.

4.3.5. Interfaces

Select: **Model → Interfaces**.

This model tree member is present only for *MultiDomain* models. An acoustic interface connects two fluid domains. It defines the relationship between the acoustic pressure and velocity at coincident points on either side of the separating surface. The normal vector that is used is that of the interface elements. The interface is defined by a geometry mesh and certain types of boundary conditions, such as *perforated* or *arbitrary* BCs.

4.3.6. Boundary Conditions

Select: **Model → Boundary Conditions**.

The boundary conditions to be imposed on the model are defined here. Boundary conditions are the acoustic physical constraints, such as *sound pressure*, *particle velocity*, *acoustic impedance*, that are applied on the surface. *Coustyx* provides options to define a wide variety of boundary conditions in a user friendly way. Refer to Section 6.3 for a detailed discussion on different types of boundary conditions. The user defines the boundary conditions as separate entities which are named uniquely here. These boundary conditions are applied later to BE elements directly or through sets (group of elements), before running the analysis (refer to Section 6.5.2).

- **Define a New Boundary Condition** Right-click on **Boundary Conditions** and select: **New** and proceed with entering new parameters information. Click OK to accept.

- **Edit an Existing Boundary Condition** Select: **Boundary Conditions** → <*Boundary Condition Name*>. Right-click on <*Boundary Condition Name*> and select: **Edit**. Proceed with editing the parameters. Click OK to accept.

4.3.7. Direct BE Meshes

Select: **Model** → **Direct BE Meshes**.

This model tree member is present only for *MultiDomain* models. It consists of a list of BE meshes generated by skinning FE structure meshes. Refer to Section 5.5 for details on how to skin a FE mesh to generate a BE mesh. To open a BE mesh in the GUI, expand **Direct BE Meshes** to find the desired BE Mesh <*Direct BE Mesh Name*> and select the mesh by left-clicking on it. Then right-click and select **Open**. The mesh in the GUI can be closed by selecting **Close**.

The *Mesh Manipulation Functions* located at the bottom of the GUI are used to manipulate the mesh opened in the GUI. Use **Selected Coord Nodes** to view the coordinates of the selected nodes in the GUI, **Selected Elements** to view details of the selected elements in the GUI, **Fill Hole** to fill holes, **New Element** to create a new boundary element, **Stitch Seams** to fill gaps by stitching seams, **Delete Elements** to delete elements from the mesh, **Merge Nodes** to merge coincident nodes in the mesh, **Split Pn Nodes** to split selected Pn nodes to create duplicate nodes along the common edge of elements with discontinuities in velocity boundary conditions, **Element Orientation** to view and flip the element normals. Refer to Section 4.2 for more details on the mesh viewer functions and GUI tools used to manipulate the BE mesh.

Each BE mesh member can be expanded to see the sub-tree members mentioned below.

4.3.7.1. Coord Nodes

Select: **Model** → **Direct BE Meshes** → <*Direct BE Mesh Name*> → **Coord Nodes**.

This sub-tree member is used to review the coordinate nodes of the mesh. Right-click on **Coord Nodes** to find the following menu options:

Open Opens the list of all coordinate nodes with their ID's and coordinates in a new window. The table rows can be copied to a clip board by right-clicking on the rows and selecting *Copy*.

Close Closes the window opened by **Open** menu item.

Display All Displays all of the coordinate nodes in the mesh opened in the GUI. This menu item is active only for meshes open in the GUI.

Hide All Hides all the displayed coordinate nodes in the GUI. This menu item is active only for meshes open in the GUI.

Display Style Changes the display style of the coordinate nodes. Refer to Section 4.2.3.2 for more details on each of the options in the Display Style dialog window.

4.3.7.2. Elements

Select: **Model** → **Direct BE Meshes** → <*Direct BE Mesh Name*> → **Elements**.
This sub-tree member is used to review the elements of the mesh. Right-click on **Elements** to find the following menu options:

Open Opens the list of all elements with their ID's and connectivity details in a new window. The table rows can be copied to a clip board by right-clicking on the rows and selecting *Copy*.

Close Closes the window opened by **Open** menu item.

Display All Displays all the elements in the mesh opened in the GUI. This menu item is active only for meshes open in the GUI.

Hide All Hides all the displayed elements in the GUI. This menu item is active only for meshes open in the GUI.

Display Style Changes the display style of the elements. Refer to Section 4.2.3.1 for more details on each of the options in the Display Style dialog window.

The **Elements** sub-tree can be expanded to browse through individual elements using *Prev* and *Next* or by directly entering the ID of the element. The right-click menu on the element ID provide these options:

Display Displays the chosen element.

Hide Hides the chosen element.

Select Selects the chosen element.

Unselect Unselects the chosen element.

Add to Set Adds the chosen element to a set. Refer to Section 4.4 for more details on Sets.

Remove from Set Removes the chosen element from its current set.

Set Boundary Condition Sets the selected boundary condition on the chosen element. Boundary condition should already be defined.

4.3.7.3. P Nodes

Select: **Model** → **Direct BE Meshes** → <*Direct BE Mesh Name*> → **P Nodes**.

Use this sub-tree member to review **P Nodes** in the mesh. **P Nodes** are variable nodes associated with *pressure* at the location of the node. *pressure* (p) and *normal derivative of pressure* (p_n) are the primary acoustic variables in Direct BE formulation used for *MultiDomain* models.

The right-click on **P Nodes** provides these menu options:

Open Opens the list of all *pressure* nodes. The table shows the ID of the *pressure* node, location coordinates (X,Y,Z), mean normal components (mean(Nx), mean(Ny), mean(Nz)), and solid angle covered by the *P* node. The mean normal components are obtained by averaging the normals at the node on all the elements connected to the node.

Close Closes the window opened by **Open** menu item.

4.3.7.4. Pn Nodes

Select: **Model** → **Direct BE Meshes** → <*Direct BE Mesh Name*> → **Pn Nodes**.

Use this sub-tree member to review **Pn Nodes** in the mesh. **Pn Nodes** are variable nodes associated with *normal derivative of pressure* (p_n) at the location of the node. The *normal derivative of pressure* (p_n) and *normal velocity* (v_n) at a point are related as follows:

$$p_n = i\rho_o \omega v_n$$

where ρ_o is the ambient density, ω is the frequency.

pressure (p) and *normal derivative of pressure* (p_n) are the primary acoustic variables in Direct BE formulation used for *MultiDomain* models.

The right-click on **Pn Nodes** provides these menu options:

Open Opens the list of all *Pn* nodes. The table shows the ID of the *Pn* node, location coordinates (X,Y,Z), mean normal components (mean(Nx), mean(Ny), mean(Nz)), and solid angle covered by the *Pn* node. The mean normal components are obtained by averaging the normals at the node on all the elements connected to the node.

Close Closes the window opened by **Open** menu item.

4.3.7.5. Constraint Equations

Select: **Model** → **Direct BE Meshes** → <*Direct BE Mesh Name*> → **Constraint Equations**.

This sub-tree member is used to review or create a new set of constraint equations. Constraint equations are necessary to correctly setup an acoustic problem when the BE model created has slightly mismatched or disjoint meshes. In constraint equations, pressure *P* nodes along the demarcating borders of the mismatched meshes are combined together to satisfy the pressure continuity at a point. These equations are not required when the model is created from a single, contiguous mesh; in which case the pressure continuity among adjacent elements is automatically satisfied by the sharing of common P nodes.

4.3.7.6. Sets

Select: **Model** → **Direct BE Meshes** → <*Direct BE Mesh Name*> → **Sets**.

This sub-tree member is used to review or create a new *Set* and add elements, faces or coordinate nodes to it. A *Set* is a group of elements, faces and coordinate nodes grouped together for organization and manipulation convenience. Refer to Section 4.4 for more details.

4.3.8. Indirect BE Mesh

Select: **Model** → **Indirect BE Mesh**.

This model tree member is present only for *Indirect* models. It consists of one BE mesh generated by skinning a FE structure mesh. Refer to Section 5.5 for more details on how to skin a FE mesh to generate a BE mesh. To view the Indirect BE mesh in the GUI: Right-click on **Indirect BE Mesh** and select **Open**. The mesh in the GUI can be closed by selecting **Close**.

The *Mesh Manipulation Functions* located at the bottom of the GUI are used to manipulate the mesh opened in the GUI. Use **Selected Coord Nodes** to view the coordinates of the selected nodes in the GUI, **Selected Elements** to view details of the selected elements in the GUI, **Fill Hole** to fill holes, **New Element** to create a new boundary element, **Stitch Seams** to fill gaps by stitching seams, **Delete Elements** to delete elements from the mesh, **Merge Nodes** to merge coincident nodes in the mesh, **Split Sigma Nodes** to split selected Sigma nodes to create duplicate nodes along the common edge of elements with discontinuities in velocity boundary conditions, **Element Orientation** to view and flip the element normals. Refer to Section 4.2 for more details on the mesh viewer functions and GUI tools used to manipulate the BE mesh.

The Indirect BE mesh when expanded shows the following sub-tree members.

4.3.8.1. Material

Select: **Model** → **Indirect BE Mesh** → **Material**.

This sub-tree member is used to select the fluid medium on either side of the boundary for acoustic analysis. To select: Right-click and select **Material** → <*Material Name*> (refer to Figure 4.24). The material defined by the <*Material Name*> is created in **Model** → **Materials**. See Section 4.3.3 on how to create a new material or edit an existing one.

4.3.8.2. Coord Nodes

Select: **Model** → **Indirect BE Mesh** → **Coord Nodes**.

This sub-tree member is used to review the coordinate nodes of the mesh. Refer to Section 4.3.7.1 for more details.

4.3.8.3. Elements

Select: **Model** → **Indirect BE Mesh** → **Elements**.

This sub-tree member is used to review the elements of the mesh. Refer to Section 4.3.7.2 for more details.

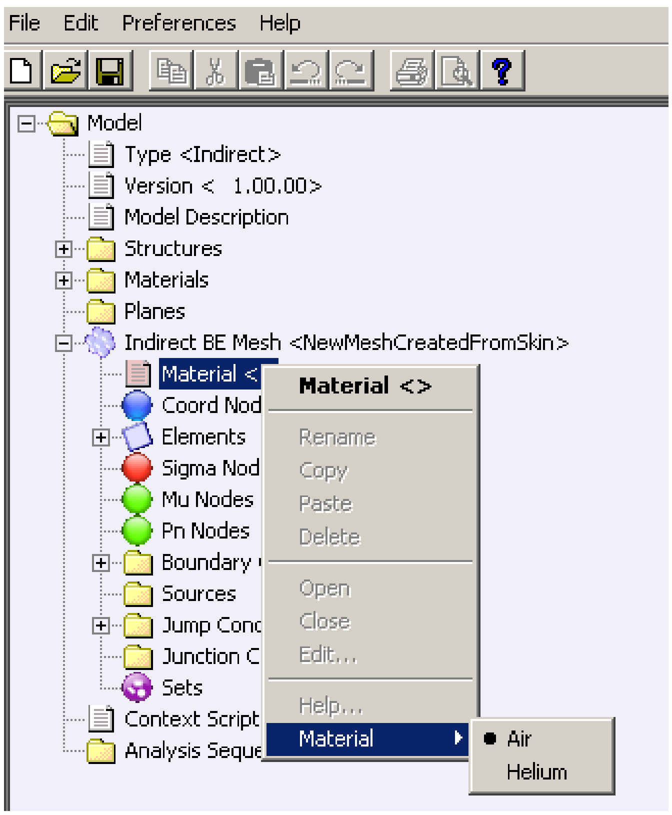


Figure 4.24: Selecting Material

4.3.8.4. Sigma Nodes

Select: **Model → Indirect BE Mesh → Sigma Nodes**.

Use this sub-tree member to review **Sigma Nodes**. **Sigma Nodes** are variable nodes associated with the single layer potential (σ) at the location of the node. The single layer potential at a point is defined as the difference of the normal derivative of pressures on the positive side (p_n^+) of the element to the negative side (p_n^-), that is, $\sigma = p_n^+ - p_n^-$.

Single layer potential (σ) and double layer potential (μ) are the primary acoustic variables in the Indirect BE formulation used for *Indirect* models.

The right-click menu provides the following options:

Open Opens the list of all *Sigma* nodes. The table shows the *Sigma* node ID, its location coordinates (X,Y,Z), and the *Status* of the node. The *Status* is determined by the solid angle covered by the node. When the solid angle is 4π the status shows: *On Free Edge.*

Close Closes the opened *Sigma* nodes window.

4.3.8.5. Mu Nodes

Select: **Model → Indirect BE Mesh → Mu Nodes**.

Use this sub-tree member to review **Mu Nodes**. **Mu Nodes** are variable nodes associated with double layer potential (μ) at the location of the node. The double layer potential at a point is defined as the difference of the pressure on the positive side (p^+) of the element to the pressure on the negative side (p^-), that is, $\mu = p^+ - p^-$.

Single layer potential (σ) and double layer potential (μ) are the primary acoustic variables in the Indirect BE formulation used for *Indirect* models.

The right-click menu provides the following options:

Open Opens the list of all *Mu* nodes. The table shows the *Mu* node ID, its location coordinates (X,Y,Z), and the *Status* of the node. The *Status* is determined by the solid angle covered by the node. When the solid angle is 4π the status shows: *On Free Edge.*

Close Closes the opened *Mu* nodes window.

4.3.8.6. Pn Nodes

Select: **Model → Indirect BE Mesh → Pn Nodes**.

This sub-tree member is used to review the *normal derivative of pressure* (*Pn*) nodes defined on a *Baffled* plane.

4.3.8.7. Boundary Conditions

Select: **Model → Indirect BE Mesh → Boundary Conditions**.

The boundary conditions to be imposed on the *Indirect* model are defined here. Boundary conditions are applied in terms of the surface *sound pressure*, *particle velocity*, *acoustic* *impedance*. *Coustyx* has a user friendly interface to define a wide variety of boundary conditions. Refer to Section 6.4 for a detailed discussion on different types of boundary conditions available for an *Indirect* model. The user defines the boundary conditions as separate entities which are named uniquely and these are applied over BE elements directly or through sets (group of elements) before running the analysis (refer to Section 6.5.2).

- **Define a New Boundary Condition** Right-click on **Boundary Conditions** and select: **New** and proceed with entering new parameters information. Click OK to accept. (Figure 4.25).

- **Edit an Existing Boundary Condition** Select: **Boundary Conditions** → <*Boundary Condition Name*>. Right-click on <*Boundary Condition Name*> and select: **Edit**. Proceed with editing the parameters. Click OK to accept.

4.3.8.8. Sources

Select: **Model** → **Indirect BE Mesh** → **Sources**.

This is used to include acoustic sources in the model. Available acoustic sources in *Coustyx* are: **Monopole**, **Dipole**, **Quadrupole**, **Plane Wave**, **Cylindrical** and **User Defined**. Refer to Section 4.5 for detailed discussion on how to define an acoustic source. To add a new acoustic source right-click on **Sources** and select **New** (Figure 4.26). The right-click menu items **Edit** and **Delete** are used to edit and delete selected sources.

4.3.8.9. Jump Conditions

Select: **Model** → **Indirect BE Mesh** → **Jump Conditions**.

The Jump conditions in the *Indirect* model can be reviewed here. Right-click and select **Open** to see the list of *Mu* nodes and their jumps in a table. The table shows the list of *Mu* node IDs and the magnitude of jumps at these nodes. For a node on *Free Edge* the magnitude of jump will be zero, that is $\mu = p^+ - p^- = 0$, as the pressures p^+ and p^- on either side of the surface are equal at the edge. *Coustyx* automatically generates this list for all *Free Edges* while skinning the FE structure mesh.

4.3.8.10. Junction Constraints

Select: **Model** → **Indirect BE Mesh** → **Junction Constraints**.

The Junction constraints in the *Indirect* model can be reviewed here. Right-click and select **Open** to see the list of constraint equations in a table. The table shows the *Constraint ID* of the equation; the left hand side consisting of the *Node Type* with its *ID* and the *Coefficient*; the right hand side (*RHS*) of the constraint equation. The *Node Type* can be any of the following types: *Sigma*, *Mu*, or *Pn*.

These constraint equations are automatically generated during the skinning of the FE structure mesh. These equations typically represent the junction constraints between double layer potentials (*Mu*s) associated with various acoustic subspaces. For example, the nodes connecting a two dimensional rib to a mesh are to be associated with junction constraints as they connect

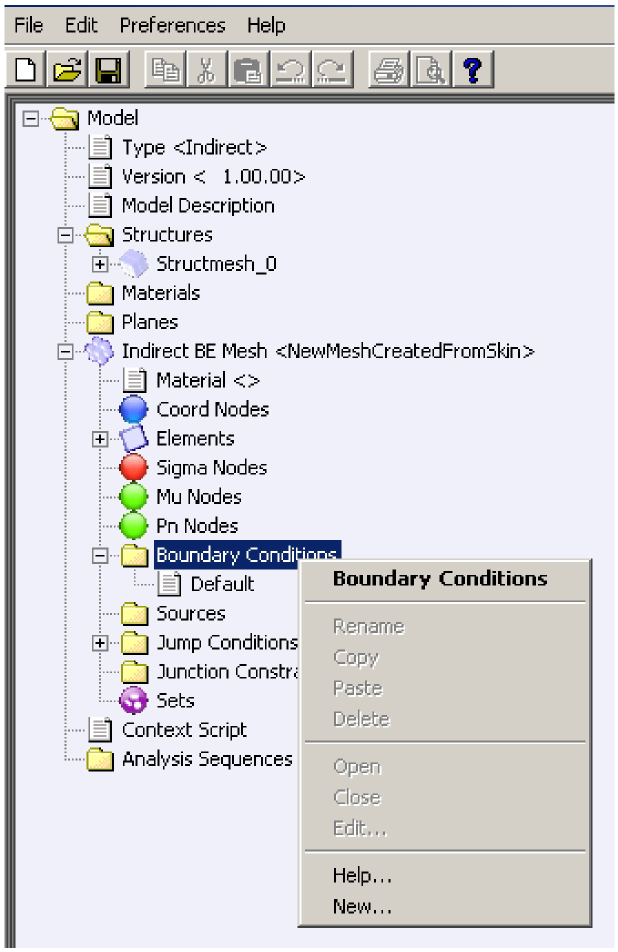


Figure 4.25: Create Boundary Conditions

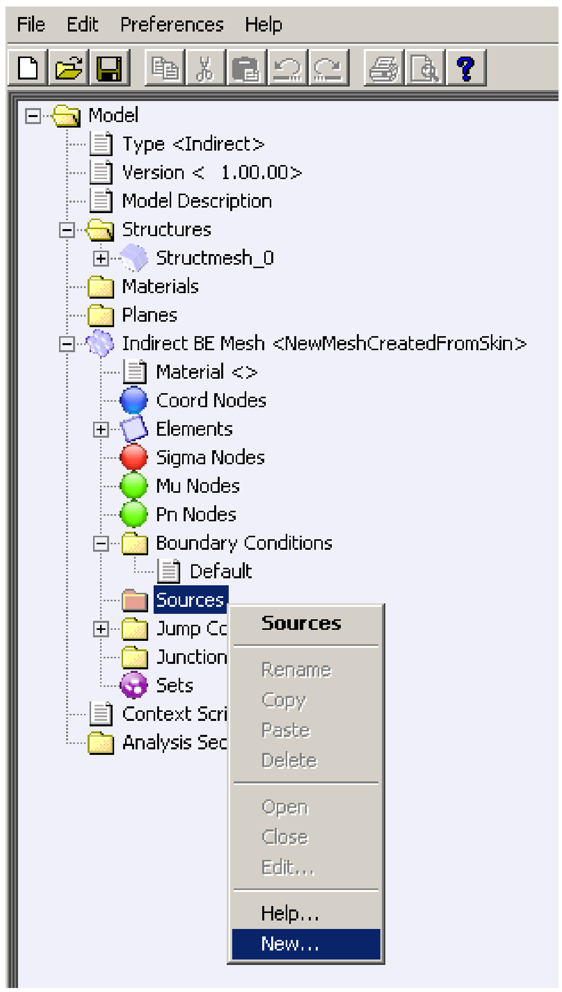


Figure 4.26: Create New Source

different subspaces. The double layer potentials at this junction are interdependent. Figure 4.27 shows a junction constraint with surfaces dividing the space into three regions. The nodes μ_1, μ_2, μ_3 represent the value of the double layer density at the same spatial location. By definition, μ_1, μ_2, μ_3 represent the following pressure differences.

$$\mu_1 = p_a - p_c$$
$$\mu_2 = p_b - p_a$$
$$\mu_3 = p_c - p_b$$

From the above expressions, the constraint equation at the junction is

$$\mu_1 + \mu_2 + \mu_3 = 0 \tag{4.2}$$

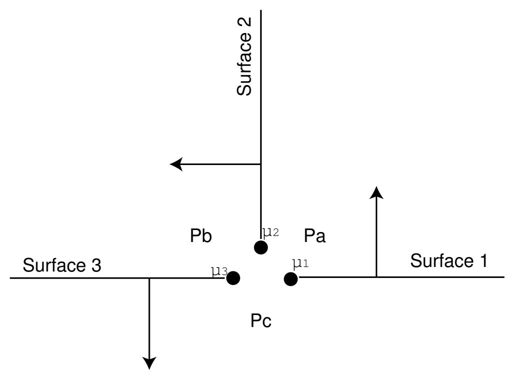


Figure 4.27: Junction constraints

4.3.8.11. Sets

Select: **Model → Indirect BE Mesh → Sets**.

This sub-tree member is used to review or create a new *Set* and add elements and nodes to it. A *Set* is a group of elements and nodes grouped together for organization and manipulation convenience. Refer to Section 4.4 for more details.

4.3.9. Domains

Select: **Model → Domains**.

This model tree member is present only for *MultiDomain* models. Multiple domains can be defined here. Each domain can have a different acoustic medium and is identified with a unique name and type. Currently only domains of type *Direct BE* are allowed. A default domain is automatically created when a new *MultiDomain* model is opened. For each domain we need to set the following options with proper care in order to solve the acoustics problem correctly.

4.3.9.1. Sources

Select: **Model** → **Domains** → <*Domain Name*> → **Sources**.

This is used to include acoustic sources in the model. Available acoustic sources in *Coustyx* are: **Monopole**, **Dipole**, **Quadrupole**, **Plane Wave**, **Cylindrical** and **User Defined**. Refer to Section 4.5 for detailed discussion on how to define an acoustic source. To add a new acoustic source right-click on **Sources** and select **New** (Figure 4.26). The right-click menu items **Edit** and **Delete** are used to edit and delete selected sources.

4.3.9.2. Boundedness

Select: **Model** → **Domains** → <*Domain Name*> → **Boundedness**.

This is used to set the boundedness of the Domain. Right-click on **Boundedness** and select **Boundedness** → **Unbounded** or **Bounded** (Figure 4.28). If the domain of interest is *exterior* then set the flag to **Unbounded**. If the domain of interest is *interior* then set the flag to **Bounded**. Please note that in a *MultiDomain* model we can have multiple *Bounded* Domains but only one *Unbounded* Domain.

4.3.9.3. Material

Select: **Model** → **Domains** → <*Domain Name*> → **Material**.

This sub-tree member is used to select the fluid medium present in the domain. To select: Right-click and select **Material** → <*Material Name*>. The material defined by the <*Material Name*> is created in **Model** → **Materials**. See Section 4.3.3 on how to create a new material or edit an existing one.

4.3.9.4. Direct BE Meshes

Select: **Model** → **Domains** → <*Domain Name*> → **Direct BE Meshes**.

Each domain can have multiple BE meshes. By default all the meshes available in the **Model** → **Direct BE Meshes** are included. To delete any particular mesh from the domain right-click on **Model** → **Domains** → <*Domain Name*> → **Direct BE Meshes** → <*Direct BE Mesh Name*> and select **Delete**. To modify the existing mesh in the domain to a different mesh, right-click and select **BE Mesh** → <*Different Direct BE Mesh Name*>.

For each Direct BE mesh the following options should be verified before running the analysis.

Select: **Model** → **Domains** → <*Domain Name*> → **Direct BE Meshes** → <*Direct BE Mesh Name*>.

Xfm Matrix This defines a 4×4 coordinate transformation matrix. The current location of the mesh in the domain is obtained by applying the transformation to all the coordinates of the BE mesh. The default transformation matrix is a unit matrix which places the BE mesh at its original position. The transformation matrix is given by

$$T = \left[\begin{array}{ccc|c} R_{11} & R_{12} & R_{13} & \Delta x \\ R_{21} & R_{22} & R_{23} & \Delta y \\ R_{31} & R_{32} & R_{33} & \Delta z \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

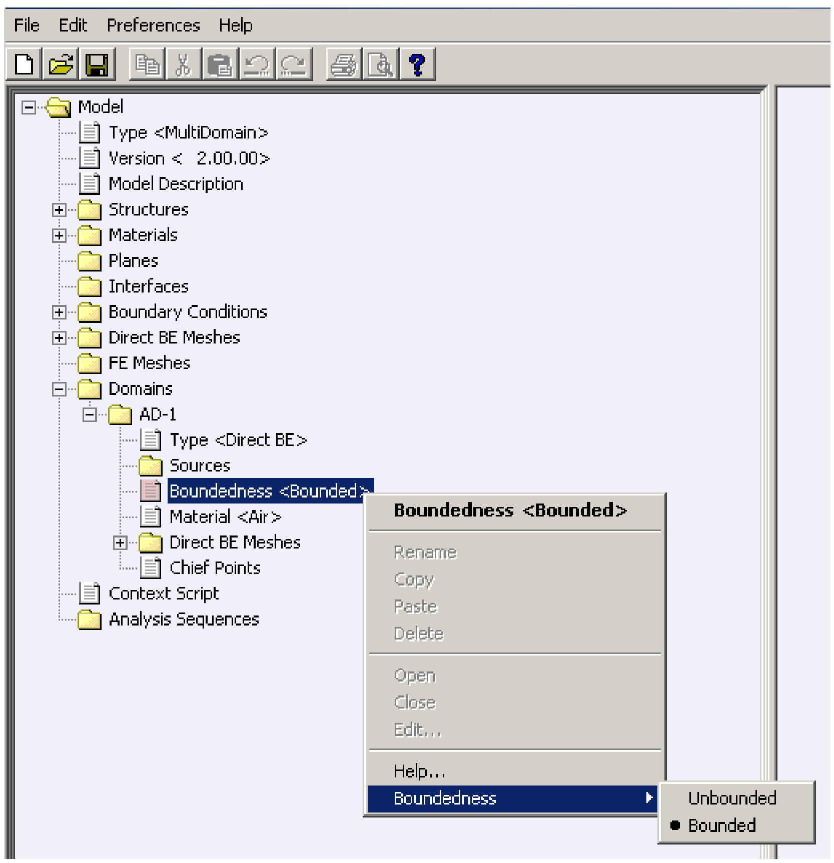


Figure 4.28: Choosing Boundedness

where the first 3×3 entries are for rotation transformation; Δx, Δy, Δz are the translations in x, y and z directions.

Note that the above transformation matrix is a general matrix that can be applied to an absolute vector (**s**), such as position vector, or to a relative vector (**n**), such as normal and velocity vector. These vectors are represented in *Coustyx* as follows:

$$\mathbf{s} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad \mathbf{n} = \begin{bmatrix} n_x \\ n_y \\ n_z \\ 0 \end{bmatrix}$$

where x, y, z are components of absolute vector; n_x, n_y, n_z are components of relative vector.

Side of Mesh on which Domain is This is used to set the side of the Mesh on which the Domain is present. Right-click on **Side of Mesh on which Domain is** to select **Side of Mesh on which Domain is → Positive** or **Negative** (Figure 4.29). The *positive* side of a mesh is defined as the side with positive mesh normal. For example, consider a sphere mesh with all the element normals pointing inward. The sphere *interior* is on the positive side of the mesh (as the mesh normals point inward) and the *exterior* is on the negative side of this mesh. If you are interested in solving the exterior acoustics problem for this mesh (where the mesh normals point inward), then you need to set the flag to **Negative**. To solve the interior problem set the flag to **Positive**. To view element orientations use the tabbed window *Element Orientation* located at the bottom of the *Mesh Viewer* window (refer to Section 5.11).

Note that in order to specify this flag we need to have all the elements of the mesh to be oriented consistently. For a *MultiDomain* model *Coustyx* automatically generates consistent BE mesh normals when skinned from a FE mesh.

Boundary Condition Mapping This lists all the boundary conditions applied on the elements of a mesh.

4.3.9.5. Chief Points

Select: **Model → Domains →** <*Domain Name*> **→ Chief Points**.

Chief points are used to specify additional constraints in the *interior* region of a *MultiDomain* model while solving an *exterior* radiation problem. Chief points, which are also called Over-determination points, are widely used to avoid large errors in the solutions to a radiation problem at the natural frequencies of the interior region under complimentary boundary conditions. To obtain accurate solutions at all frequencies, specify random distribution of chief points in the interior region.

A chief point is identified with an ID, and coordinates X, Y, Z. Right-click and select **Edit** to review or modify the list of Chief Points from a table. (Figure 4.30)

Id The id of a Chief Point. Note that each Chief Point should have unique Id.

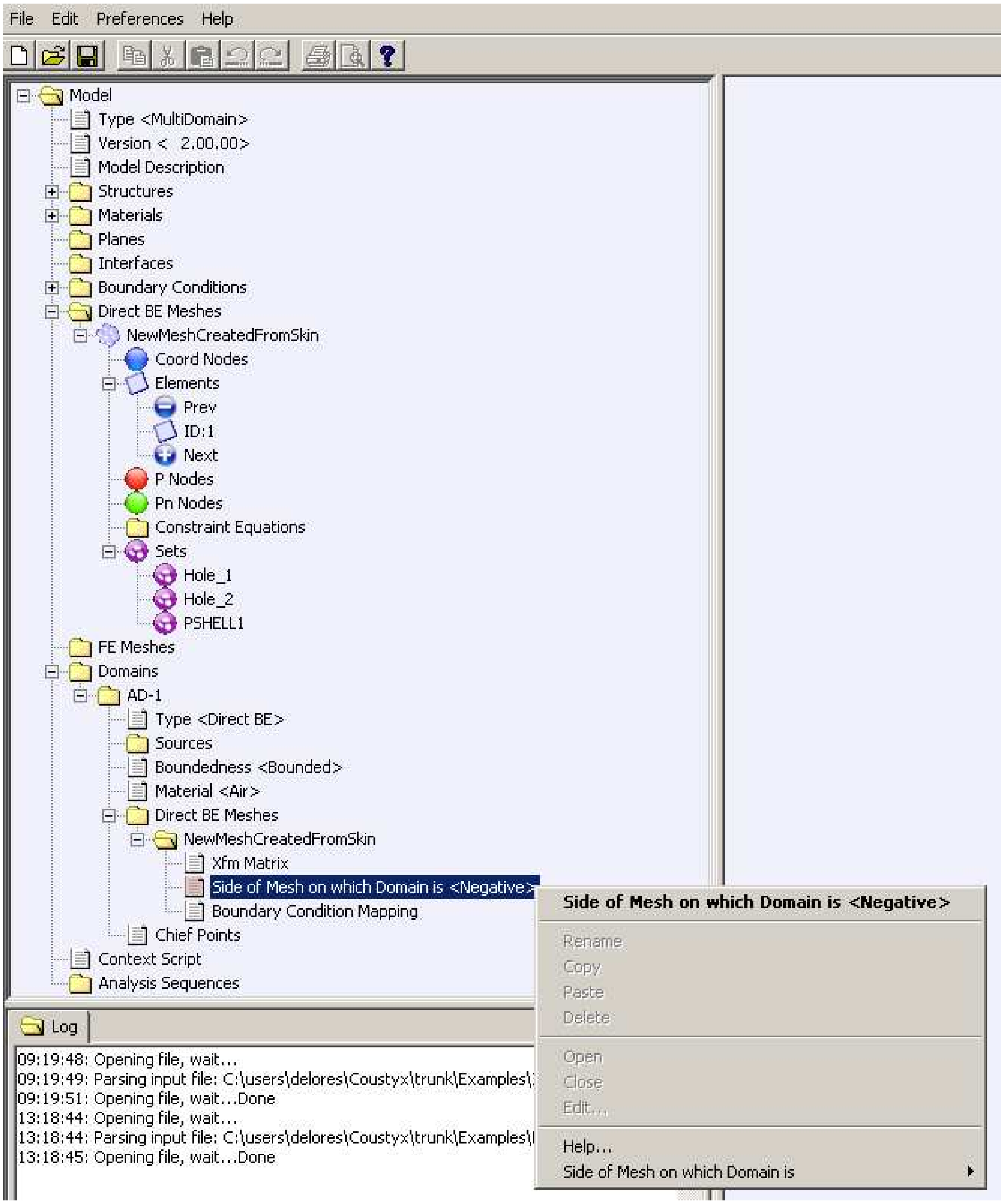


Figure 4.29: Side of Mesh on Domain Function

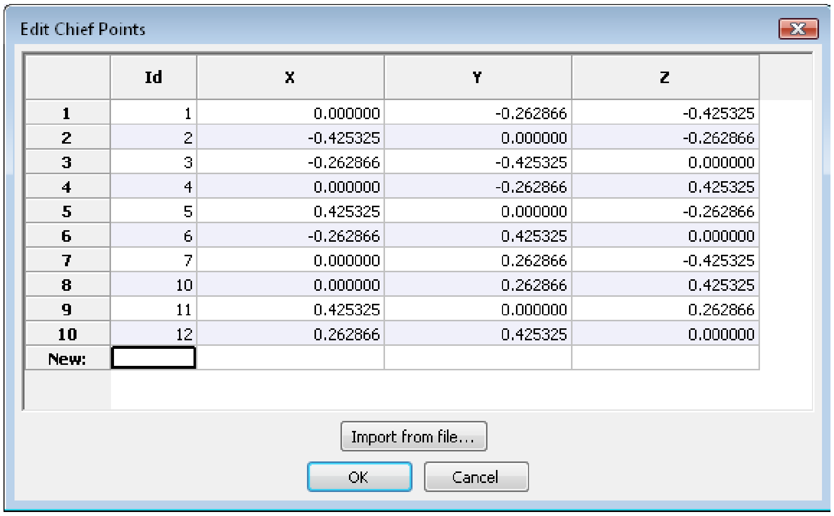


Figure 4.30: Chief Points Edit Dialog Box.

X The x coordinate of a Chief Point.

Y The y coordinate of a Chief Point.

Z The z coordinate of a Chief Point.

Import From File Chief Points can be imported from an *ASCII* file with components separated by commas, tabs or spaces. The file must contain four columns with **Id** and, **X**, **Y** and **Z** coordinates of a Chief Point (Figure 4.31). Each new Chief Point is added to a new row. Import Options window opens up after the selection of the file with the following options:

Whether to replace the table, or append the data to it The imported data from the file can be used to either replace the current table or append to the existing table by the selection of one of the options: **Replace** or **Append**. (Figure 4.31)

Scale factor The X, Y, Z coordinates of Chief Points in the *ASCII* file are multiplied by the Scale factor before being read into the table (Figure 4.31). This is specifically useful when the imported file has different units compared to *Coustyx* model. A unit conversion factor should be used as the Scale factor to convert these values. For example, when the coordinates in the *ASCII* file are in m and the *Coustyx* units are in mm, a Scale factor of 1000 is entered to convert the values in the file from m to mm. A Scale factor of 1 imports the values as they are.

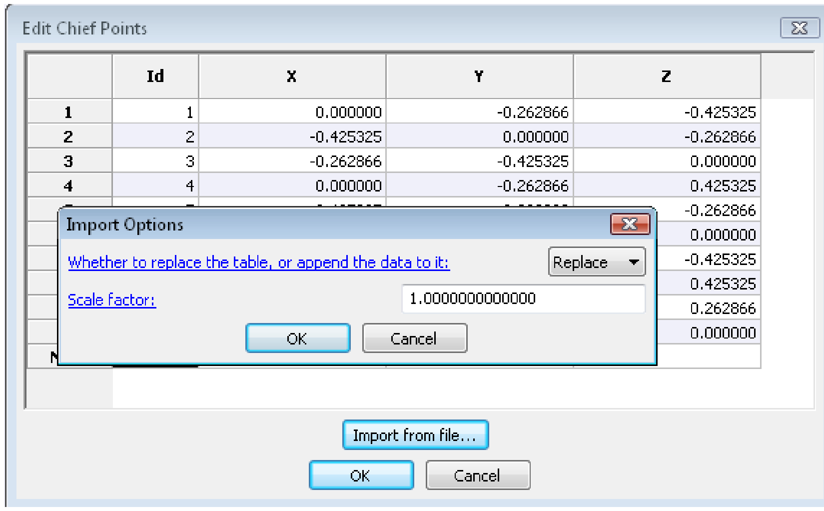


Figure 4.31: Import Chief Points from a File.

4.3.10. Context Script

Select: **Model** → **Context Script**.

Coustyx provides the option to define global variables through Context Script. These variables can then be used in any other scripts, such as Boundary Condition, Analysis Sequence, etc. The Context script is executed once at the start of the analysis to compute the global variable values. The predefined read-only variables that can be used in the Context script are: *AngularFreq* (ω) – frequency in radians/sec, *SoundSpeed* (c) – speed of sound in the fluid medium with the same units as those defined in materials, *WaveNumber* ($k = \frac{\omega}{c}$), *AmbientDensity* (ρ) – density of the fluid medium with the same units as those defined in materials. Right-click and select **Open** to open the Context script for editing, select **Close** to close the window. (Figure 4.32).

4.3.11. Analysis Sequences

Select: **Model** → **Analysis Sequences**.

Analysis Sequences is a model tree member which is used to set the options, such as frequency range, solution methods, etc., required to run an analysis. New analysis sequence is created by right-clicking and selecting the menu item **New**. Multiple Analysis Sequences can be created by repeating this action. Refer to Chapter 7 for a detailed discussion on selection of various parameters for the analysis.

Once the *Coustyx* model setup is completed, run the acoustic analysis by right-clicking on the desired sequence: **Model** → **Analysis Sequences** → <*Analysis Sequence Name*> and selecting **Run**. The analysis results are stored in the output files referred in <*Analysis Sequence Name*>.

4.4. Sets

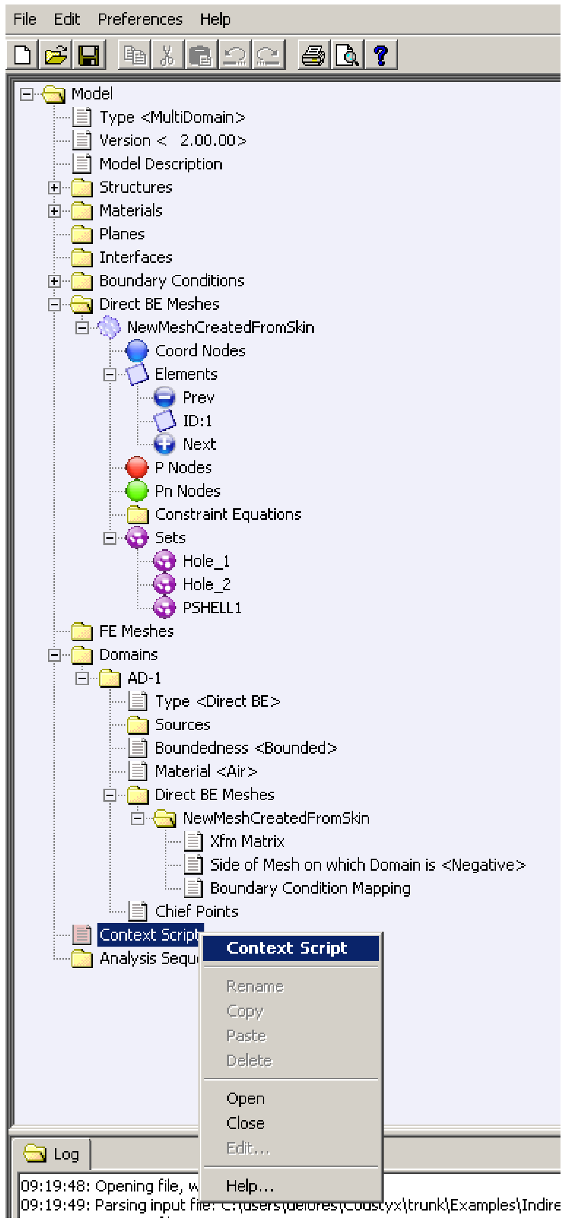


Figure 4.32: Context Script

For a *MultiDomain* BE mesh, Select: **Model** → **Direct BE Meshes** → <*Direct BE Mesh Name*> → **Sets**.

For an *Indirect* BE mesh, Select: **Model** → **Indirect BE Mesh** → **Sets**.

For a structure mesh, Select: **Model** → **Structures** → <*Structure Name*> → **Sets**.

A *Set* is a collection of elements, coordinate nodes, and faces grouped together for organization and manipulation convenience. This sub-tree member is used to review or create a new *Set* and add elements, coordinate nodes, or faces to it.

Create a new Set Right-click on **Sets** and select **New** to create a new Set (Figure 6.41).

Rename To rename the new Set, right-click on <*Set Name*> and select **Rename**.

Copy To copy contents of a Set, right-click on <*Set Name*> and select **Copy**.

Paste To paste the copied contents of a Set to a new Set, right-click on <*Set Name*> and select **Paste**.

Open To open the list of contents of a Set in a table, right-click on <*Set Name*> and select **Open**. The table lists the components in the Set. The first column shows the **Type** of the component, that is Elements, Nodes or Faces, the second column shows the **ID** of the component, and the third column shows the **Data** related to the component. The data exists only for a face and it represents the face number in the element.

Close To close the list of contents of a Set opened in a table, right-click on <*Set Name*> and select **Close**.

Replicate To replicate the contents of a Set, right-click on <*Set Name*> and select **Replicate**. This function performs the tasks: **Copy** and **Paste** together.

Elements Right-click on <*Set Name*> and select **Elements** to find the following sub-menu options. Note that all the options are not enabled when the mesh is not open in the GUI. Figure 4.34.

- **Display Style** Changes the display style of the elements in the Set. Refer to Section 4.2.3.1 for more details on each of the options in the Display Style dialog window.
- **Display All** Displays all the elements of the Set in the GUI.
- **Hide All** Hides all the displayed elements of the Set in the GUI.
- **Set Boundary Condition** Sets the selected boundary condition to all the elements of the Set. This option is available only for BE meshes.
- **Select** Selects all the elements of the Set in the GUI.
- **Unselect** Unselects elements of the Set in the GUI.
- **Add Selection to Set** Adds selected elements to the current Set.
- **Remove Selection from Set** Removes the selected elements from the current Set.

CoordNodes Right-click on <*Set Name*> and select **CoordNodes** to find the following sub-menu options. Note that all the options are disabled when the mesh is not open in the GUI. Figure 4.35.

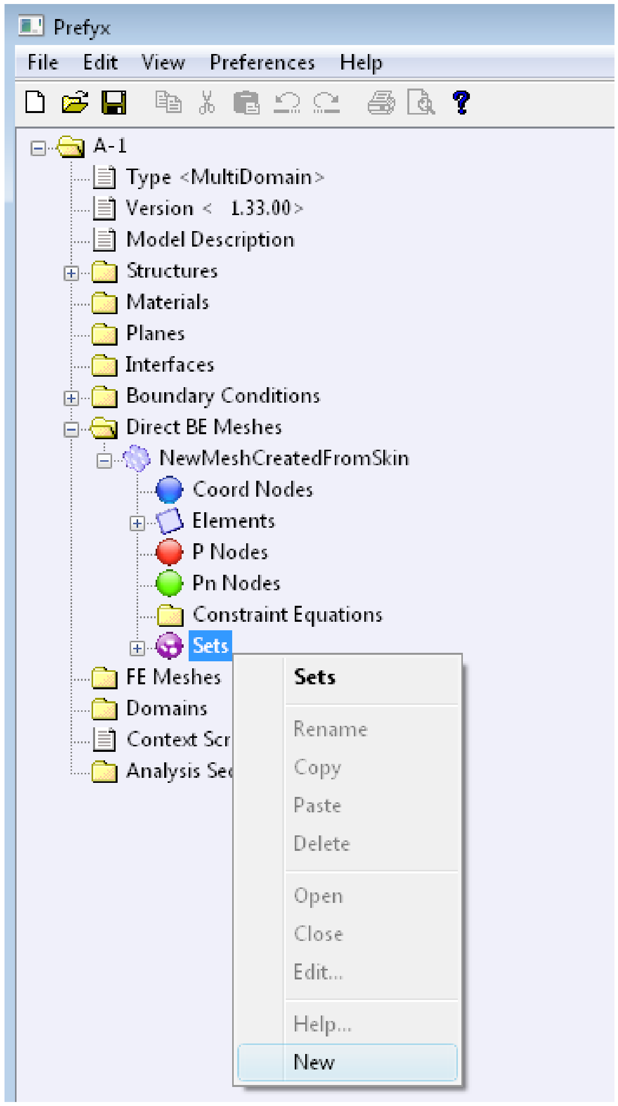


Figure 4.33: Creating a new set, a group of elements and nodes.

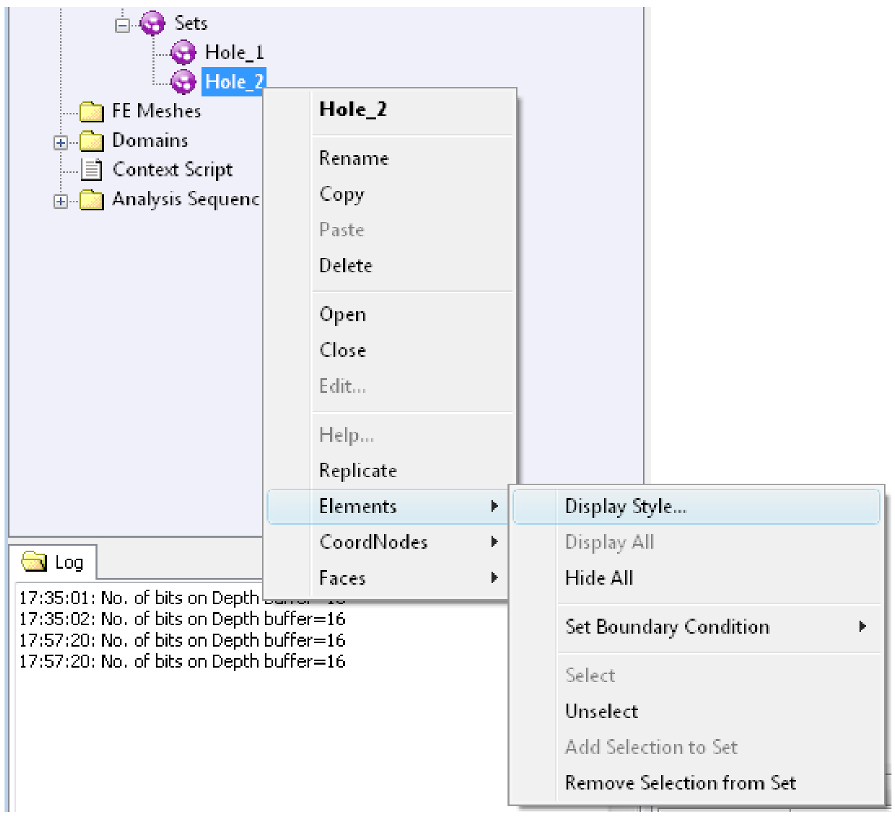


Figure 4.34: Menu options for elements in a Set.

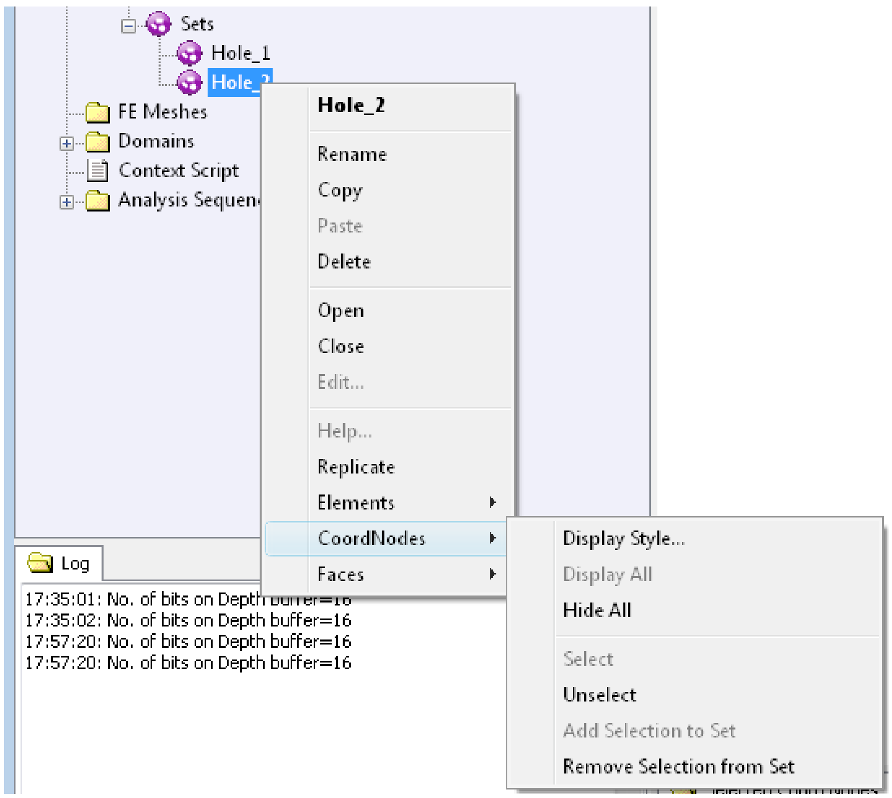


Figure 4.35: Menu options for coordinate nodes in a Set.

Display Style Changes the display style of the coordinate nodes in the Set. Refer to Section 4.2.3.2 for more details on each of the options in the Display Style dialog window.

Display All Displays all the coordinate nodes of the Set in the GUI.

Hide All Hides all the displayed coordinate nodes of the Set in the GUI.

Select Selects all the coordinate nodes of the Set in the GUI.

Unselect Unselects coordinate nodes of the Set in the GUI.

Add Selection to Set Adds selected coordinate nodes to the current Set.

Remove Selection from Set Removes the selected coordinate nodes from the current Set.

Faces Right-click on <*Set Name*> and select **Faces** to find the following sub-menu options. Note that all the options are disabled when the mesh is not open in the GUI. Figure 4.36.

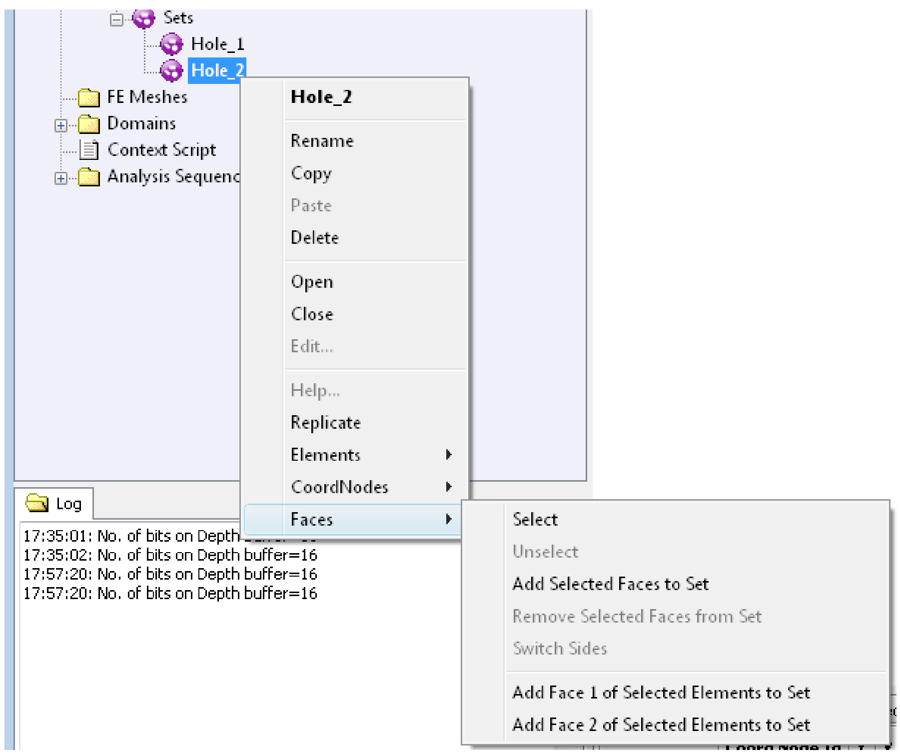


Figure 4.36: Menu options for faces in a Set.

Select Selects all the faces of the Set in the GUI.

Unselect Unselects faces of the Set in the GUI.

Add Selection to Set Adds selected faces to the current Set.

Remove Selection from Set Removes selected faces from the current Set.

Switch Sides This option is available for BE meshes (surface elements) only. See Figure 6.16 for the description of Side 1 and Side 2 of a surface element. When selected, *Coustyx* switches the selection side of the element from the current face to its opposite. It modifies the component list of the Set with the new face selection. For example, consider a Set which contains Side 1 of an element as its Face component. When Switch Sides is applied, the Set is modified and the Side 2 of the element (that is, the opposite side) is saved in the place of Side 1. If you press Switch Sides again you will get back to the original Set configuration. Note that this option is enabled only when the Face components of the Set are selected in the GUI. Also, this option is available only for BE meshes.

Add Face 1 of Selected Elements to Set This option is available for BE meshes (surface elements) only. See Figure 6.16 for the description of Side 1 and Side 2 of a surface element. When selected, *Coustyx* adds Face 1 (or Side 1) of selected elements in the GUI to the Set.

Add Face 2 of Selected Elements to Set This option is available for BE meshes (surface elements) only. See Figure 6.16 for the description of Side 1 and Side 2 of a surface element. When selected, *Coustyx* adds Face 2 (or Side 2) of selected elements in the GUI to the Set.

4.4.1. Add elements, coordinate nodes, or faces to a Set

Select elements, coordinate nodes, and faces to be added to a Set from the GUI by left-clicking on them while holding down the *shift-key*. Right-click to see the menu on **Operations on Selection**. To add the selected to a set: **Selected Elements**, **Selected Nodes**, or **Selected Faces** → **Add to Set** → <*Set Name*> or **Add to New Set**. Figure 4.37.

4.5. Acoustic Sources

Acoustic sources can be introduced in addition to BE meshes in *Coustyx*. These acoustic sources have analytical solutions which are incorporated into the BE formulation. *Coustyx* offers the following acoustic wave sources.

4.5.1. Monopole

A *monopole* is a spherical wave source which produces spherically symmetric waves in an un-bounded space. This source is defined by a position vector $\mathbf{R}(x_r, y_r, z_r)$, and amplitude, A (or volume velocity, V) (Figure 4.38).

The monopole amplitude A and the volume velocity V are related as follows:

$$A = jkZ_oV$$

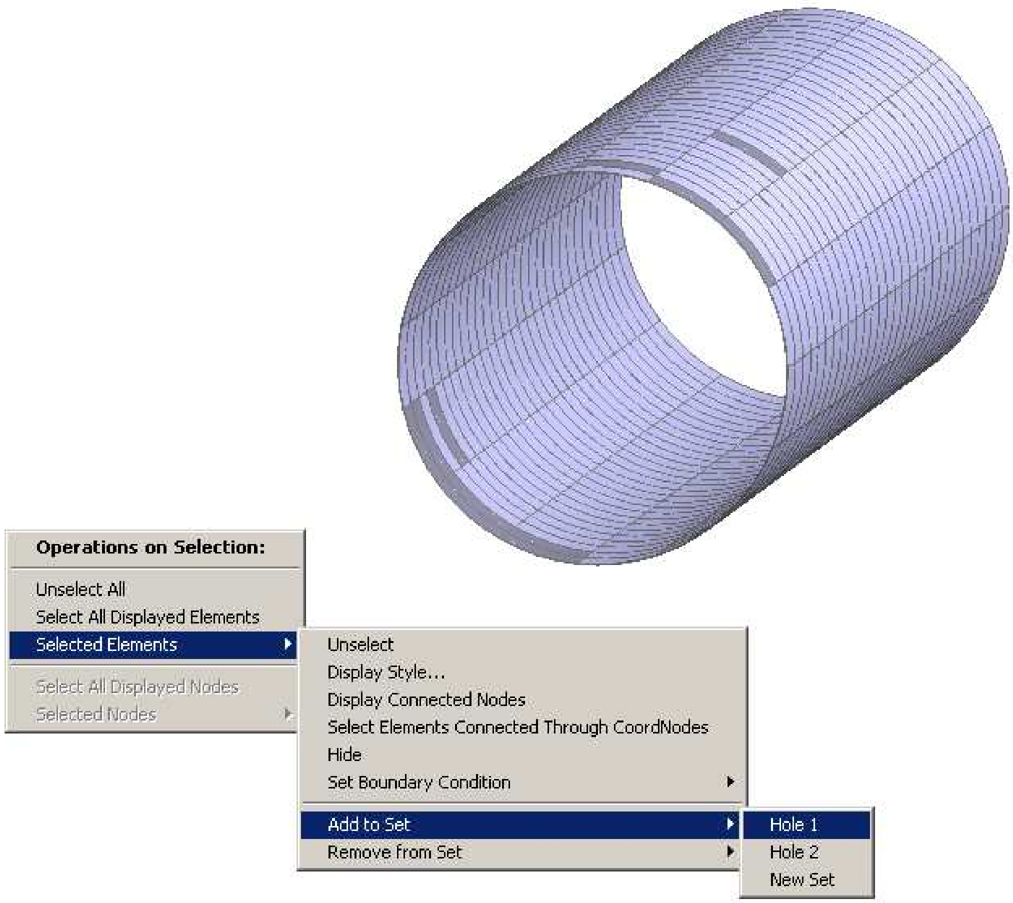


Figure 4.37: Add elements to a set.

where $Z_o = \rho_o c$ is the characteristic impedance of the fluid medium, c is sound speed, ρ_o is ambient density, $k = \omega/c$ is wave number, ω is frequency in radians/sec, and $j = \sqrt{(-1)}$. Note that the amplitude A has dimensions M/T^2, whereas the volume velocity V has dimensions L^3/T.

The wave equation for a *monopole* source in frequency domain and its solution is given by

$$\nabla^2 p(\mathbf{r}) + k^2 p(\mathbf{r}) = jkZ_oV\delta(\mathbf{r} - \mathbf{R}) \tag{4.3}$$

$$p(\mathbf{r}) = -A\frac{e^{jk|\mathbf{r}-\mathbf{R}|}}{4\pi|\mathbf{r}-\mathbf{R}|} = -jkZ_oV\frac{e^{jk|\mathbf{r}-\mathbf{R}|}}{4\pi|\mathbf{r}-\mathbf{R}|} \tag{4.4}$$

where $p(\mathbf{r})$ is sound pressure at position $\mathbf{r}$(x,y,z).

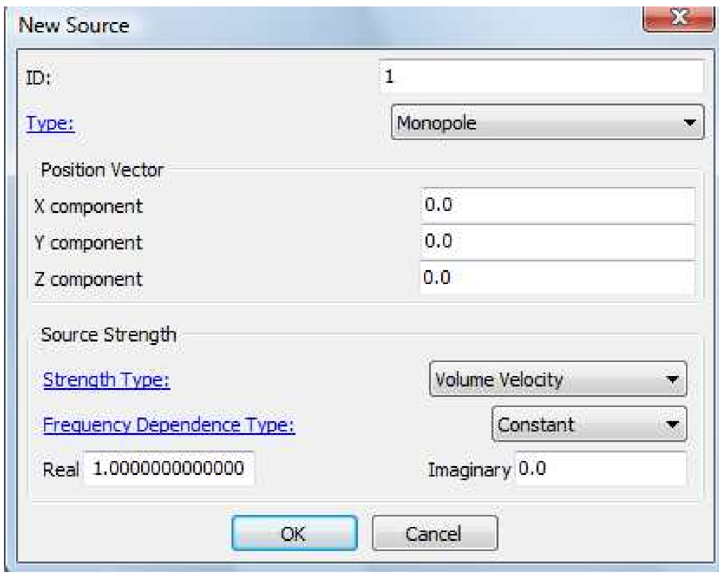


Figure 4.38: Monopole

Position Vector The location **R** of the monopole source is set by **X component**, **Y component**, and **Z component**. Note that the units should be consistent with the geometry units.

Source Strength The monopole source strength could be set to be any of the following two types: **Amplitude**, A, or **Volume Velocity**, V. Choose the type from the drop-down menu: **Strength Type**. Then define the value of the monopole source strength (amplitude, A, or volume velocity, V) through any of the frequency dependent types: *Constant*, *Table*, or *Script*. Note that the units used here should be consistent with the rest of the model inputs.

4.5.2. Plane Wave

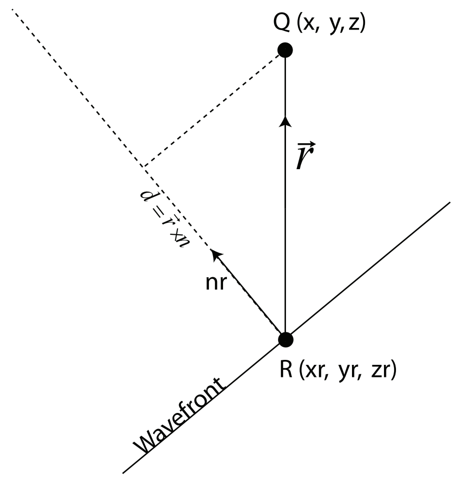


Figure 4.39: Description of a plane wave acoustic source.

A plane wave source is defined by a location $\mathbf{R}(x_r, y_r, z_r)$, an amplitude A and a direction of propagation $\mathbf{n}_r$. A has the same dimensions as pressure (Figure 4.39, Figure 4.40). The pressure p due to a plane wave at any point $\mathbf{Q}(x, y, z)$ is

$$\tilde{p}(Q) = Ae^{(jk\mathbf{r}\cdot\mathbf{n}_r)}$$
$$\mathbf{r} = \begin{Bmatrix} x - x_r \\ y - y_r \\ z - z_r \end{Bmatrix} \tag{4.5}$$

Position Vector The location **R** of the plane wave source is set by **X component**, **Y component**, and **Z component**. Note that the units should be consistent with the geometry units.

Source Strength Set the amplitude A of the plane wave through any of the frequency dependent types: *Constant*, *Table*, or *Script*. Note that the units used here should be consistent with the rest of the model inputs.

Direction Vector The plane wave propagation direction $\mathbf{n}_r$ is set by **X component**, **Y component**, and **Z component**.

4.5.3. Cylindrical

A cylindrical wave source is defined by a location $\mathbf{R}(x_r, y_r, z_r)$, an amplitude A and its axial direction of propagation $\mathbf{n}_r$ (Figure 4.41, Figure 4.42). The wave equation for a cylindrical wave, propagating in z direction is

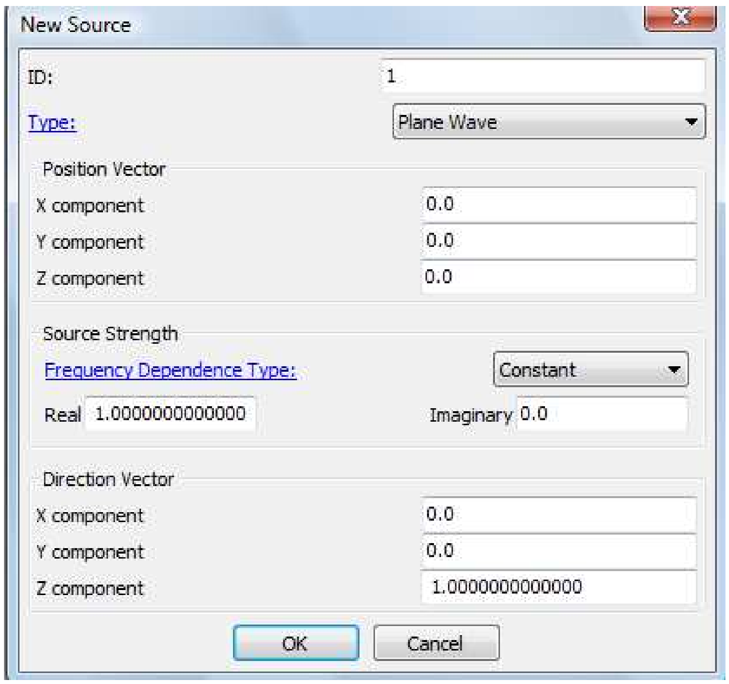


Figure 4.40: Plane Wave

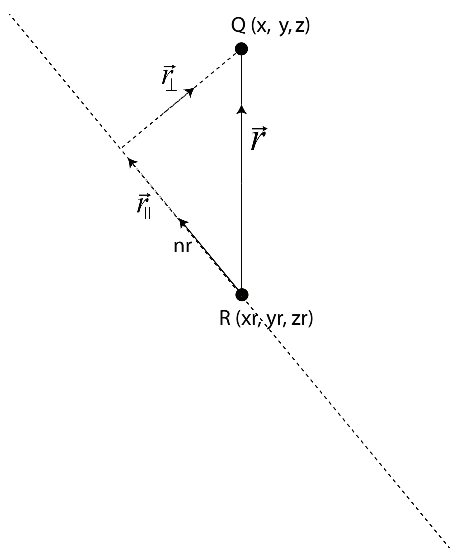


Figure 4.41: Description of a cylindrical wave acoustic source.

$$\nabla^2 p + k^2 p = A\delta(x - x_r)\delta(y - y_r) \tag{4.6}$$

where p is the pressure at a point $\mathbf{Q}(x, y, z)$.

Position Vector The location **R** of the cylindrical wave source is set by **X component**, **Y component**, and **Z component**. Note that the units should be consistent with the geometry units.

Source Strength Set the amplitude A of the cylindrical wave through any of the frequency dependent types: *Constant*, *Table*, or *Script*. Note that the units used here should be consistent with the rest of the model inputs.

Direction Vector The cylindrical wave axial direction $\mathbf{n}_r$ is set by **X component**, **Y component**, and **Z component**.

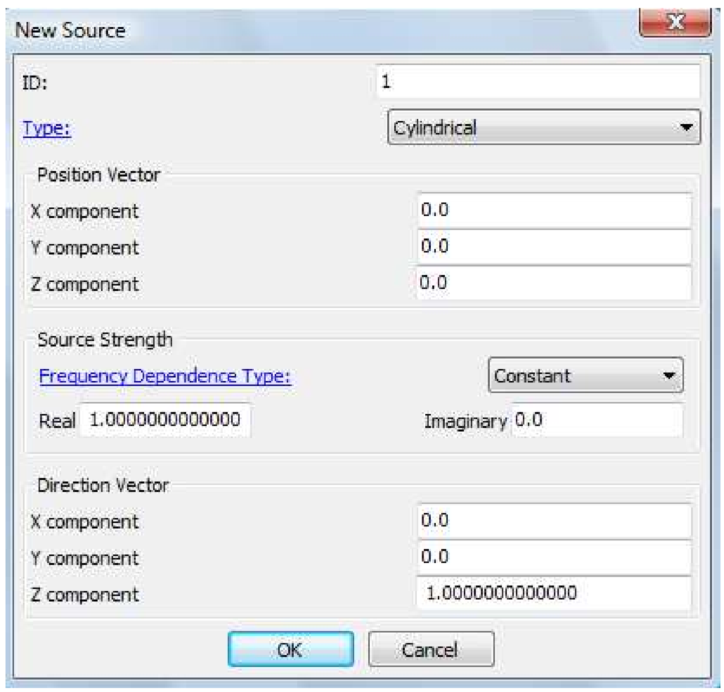


Figure 4.42: Cylindrical

4.5.4. Dipole

A dipole source is defined by a location, an amplitude, A (or dipole moment, D) and the direction of the dipole. Figure 4.43 shows an acoustic dipole source at a location $\mathbf{R}(x_r, y_r, z_r)$

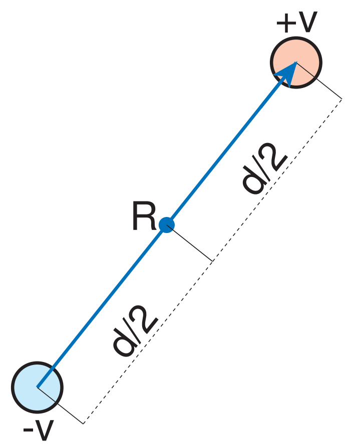


Figure 4.43: An acoustic dipole at $\mathbf{R}(x_r, y_r, z_r)$, modeled by two point monopole sources with equal and opposite volume velocities $-V$ and $+V$.

modeled by two point monopole sources. The direction of the dipole source $\mathbf{n}_r$ is from the monopole sink $(-V)$ to the monopole source $(+V)$, where V is the monopole volume velocity. The dipole moment, $D = Vd$, where d is the distance between the two monopole sources that make up the dipole.

The dipole amplitude A and the dipole moment D are related as follows:

$$A = jkZ_oD$$

where $Z_o = \rho_o c$ is the characteristic impedance of the fluid medium, c is sound speed, ρ_o is ambient density, $k = \omega/c$ is wave number, ω is frequency in radians/sec, and $j = \sqrt{(-1)}$. Note that the amplitude A has dimensions ML/T^2, whereas the dipole moment D has dimensions L^4/T.

The wave equation for a dipole source is given by

$$\nabla^2 p + k^2 p = jkZ_o \left[-V\delta(\mathbf{R} - \frac{d}{2}\mathbf{n}_r - \mathbf{Q}) + V\delta(\mathbf{R} + \frac{d}{2}\mathbf{n}_r - \mathbf{Q}) \right] \tag{4.7}$$

where p is the pressure at a point $\mathbf{Q}(x, y, z)$.

Position Vector The location $\mathbf{R}$ of the dipole source is set by **X component**, **Y component**, and **Z component**. Note that the units should be consistent with the geometry units.

Source Strength The dipole source strength could be set to be any of the following two types: **Amplitude**, A, or **Dipole Moment**, D. Choose the type from the drop-down menu: **Strength Type**. Then define the value of the dipole source strength (amplitude, A, or dipole moment, D) through any of the frequency dependent types: *Constant*, *Table*, or *Script*. Note that the units used here should be consistent with the rest of the model inputs.

Direction Vector The direction of the dipole $\mathbf{n}_r$ is set by **X component**, **Y component**, and **Z component**.

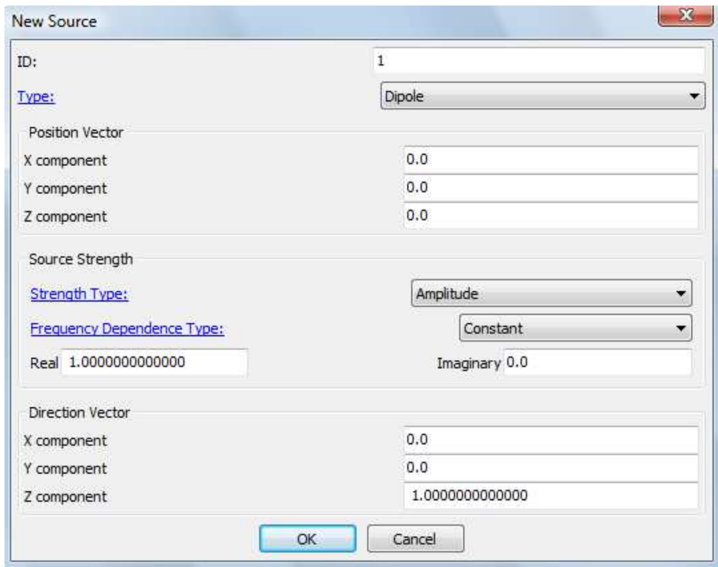


Figure 4.44: Dipole

4.5.5. Quadrupole

A quadrupole source is defined by a location $\mathbf{R}(x_r, y_r, z_r)$, and a tensor quantity $\mathbf{T}$ with its quadrupole moments or amplitudes in various directions (Figure 4.45).

$$\mathbf{T} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix}$$

The components in the tensor of quadrupole amplitudes (T_{ij}^A) are related to the components in the tensor of quadrupole moments (T_{ij}) as follows:

$$T_{ij}^A = jkZ_oT_{ij}$$

where $Z_o = \rho_o c$ is the characteristic impedance of the fluid medium, c is sound speed, ρ_o is ambient density, $k = \omega/c$ is wave number, ω is frequency in radians/sec, and $j = \sqrt{(-1)}$. In the tensor of quadrupole source moments, each term has a dimension of L^5/T. Dimensionally, this is equivalent to Vl^2 or Dl, where V is volume velocity, l is some characteristic length, and D is dipole moment. The terms in the tensor of quadrupole amplitudes have dimensions of ML^2/T^2.

The pressure p at any point $\mathbf{Q}(x, y, z)$ due to a quadrupole is given by the following expression.

$$\tilde{p}(Q) = -jkZ_o \left\{ \begin{array}{l} T_{11}\frac{\partial^2 G(Q,R)}{\partial x_s^2} + T_{22}\frac{\partial^2 G(Q,R)}{\partial y_s^2} + T_{33}\frac{\partial^2 G(Q,R)}{\partial z_s^2} \\ +(T_{12}+T_{21})\frac{\partial^2 G(Q,R)}{\partial x_s \partial y_s} \\ +(T_{13}+T_{31})\frac{\partial^2 G(Q,R)}{\partial x_s \partial z_s} \\ +(T_{23}+T_{32})\frac{\partial^2 G(Q,R)}{\partial y_s \partial z_s} \end{array} \right\} \tag{4.8}$$

where $G(Q,R)$ is the Green's function evaluated between **Q** and **R**.

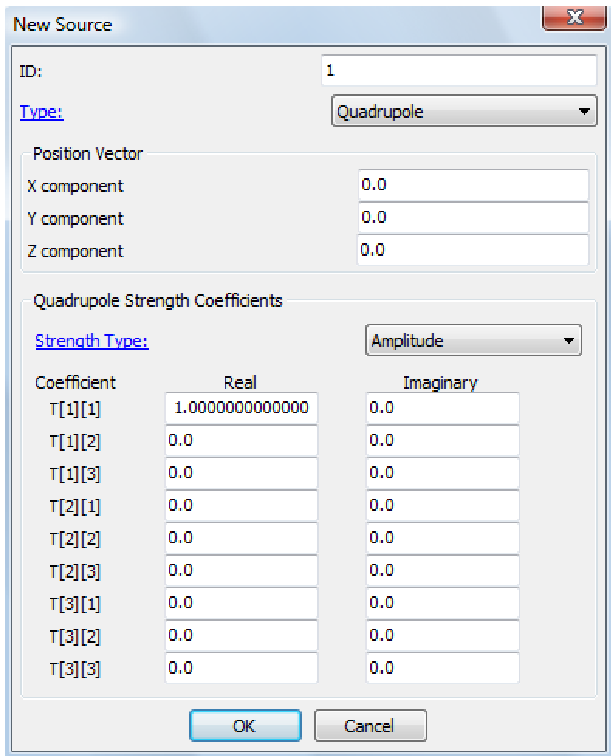


Figure 4.45: Quadrupole Dialog Box

Position Vector The location **R** of the quadrupole source is set by **X component**, **Y component**, and **Z component**. Note that the units should be consistent with the geometry units.

Quadrupole Strength Coefficients The quadrupole strength is set through the tensor components or coefficients. These components could be set to be any of the following two types: **Amplitude**, T_{ij}^A or **Quadrupole Moment**, T_{ij}. Choose the type from the drop-down menu **Strength Type**. Note that the units used here should be consistent with the rest of the model inputs.

4.5.6. User Defined

Coustyx offers users the choice to define their own source. The *User Defined* source is defined by the location of the source through **Position Vector** and the function `GetDirectionalResponseAtFieldPoint`. The relative position of a field point with respect to the source position and the field normal are input to the function through the arguments: `RelativePosnVec` and `NormalVec`. The pressure and the normal derivative of pressure at the field point are output through the arguments: `P` and `Pn` (Figure 4.46). Note that a valid user defined acoustic source should have a pressure field that satisfies the Helmholtz equation exactly.

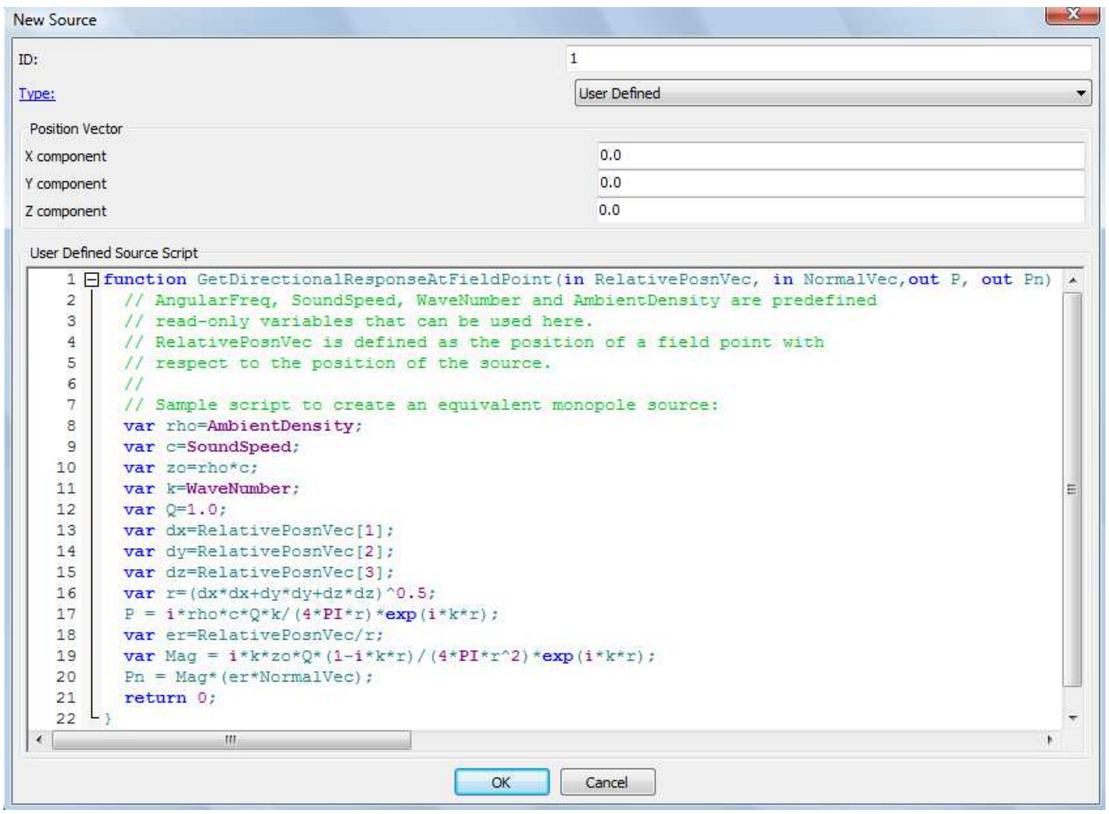


Figure 4.46: User Defined Acoustic Source

Chapter 5

Pre-processing Features

Coustyx provides a large set of pre-processing tools to manipulate meshes in the model. In addition, translators for most commonly used data formats are available to import mesh and velocity data. The pre-processing tools are found at the bottom pane of *Mesh Viewer* window while a mesh is open in the GUI. The features include: **Fill Hole**, **Skin**, **New Elements**, **Stitch Seams**, **Delete Elements**, **Merge Nodes**, **Split Pn Nodes/Split Sigma Nodes** and **Element Orientation**. These features allow users to fix problems with imported meshes by deleting elements, creating new elements, filling holes, stitching gaps, merging coincident nodes, flipping element normals etc. The imported finite element structure meshes could be skinned to generate surface BE meshes for acoustic analysis.

5.1. Importing FE Data

The main geometry source for a Coustyx Boundary Element (BE) model is the Finite Element (FE) mesh of the structure. This section explains the steps involved in importing FE data. *Coustyx* provides translators for most commonly used file formats created by finite element packages: NASTRAN, ABAQUS, ANSYS, and I-DEAS (Universal Files).

Select: **Model → Structures**.

- To import FE mesh: Right-click on **Structures** and select **Import → Nastran Bulk Data (.bdf) File** (see Figure 5.1) to import mesh from Nastran bulk data format. The FE meshes from other data formats can be imported by selecting **Abaqus (.inp) File**, **Ansys Results (.rst) File**, or **I-DEAS Universal (.unv) File**.

- To load frequency response data from FE analysis: Click on the imported structure <*Structure Mesh Name*>, right-click on it and select **Load Freq Response Data → Nastran Punch File** (see Figure 5.2). The other supported data formats are **Nastran OP2 File**, **Ansys rst File**, **Ansys rfrq File**, **I-DEAS Universal File**. Select desired frequencies to load. Only these frequencies from the data read are loaded into the model. The **Natural Mode Data** could be loaded in a similar fashion.

5.1.1. Importing Structure Mesh

Select: **Model** → **Structures**.

Right-click on **Model** → **Structures** and select any of the file format options from **Import** (refer to Figure 5.1). *Coustyx* has the capability to import FE meshes from the following data formats.

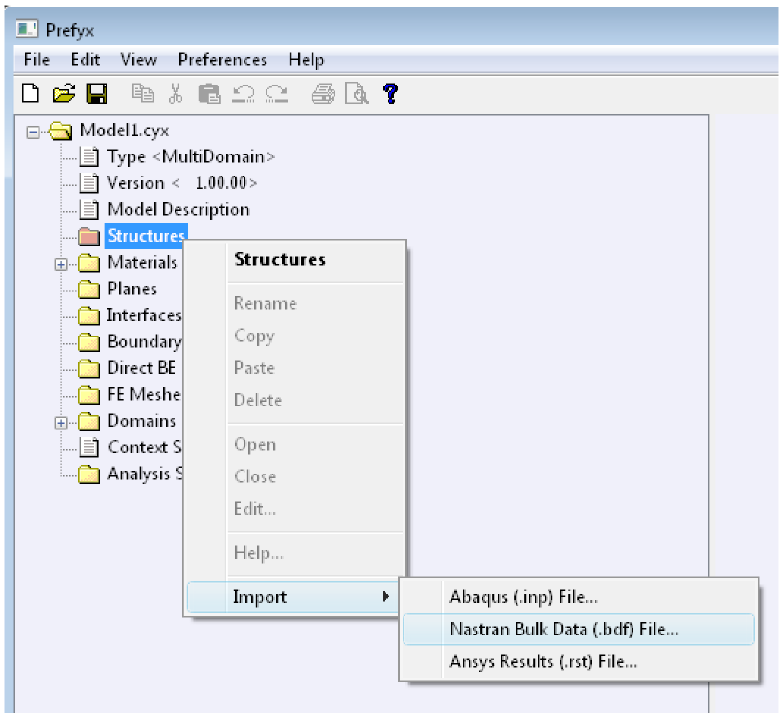


Figure 5.1: Importing a finite element structure mesh.

5.1.1.1. Abaqus *.inp files

ABAQUS is a commercial FEA software for general purpose non-linear finite element analysis used for engineering simulations and analysis. The ABAQUS input file is an *ASCII* file with an extension **.inp*. This file contains information about mesh geometry, properties of the material, boundary conditions and other commands to control output data. When imported, *Coustyx* reads only the information regarding the geometry of the model from these files.

5.1.1.2. Nastran Bulk Data *.bdf files

The NASTRAN bulk data format (**.bdf*) is widely used in the industry and can be generated from a large number of commercially available FE mesh generators. The **.bdf* file contains information about the geometry of the mesh which is extracted by *Coustyx*.

5.1.1.3. Ansys Results *.rst files

ANSYS is another commercially available FE modeling package used widely for engineering simulations. The ANSYS **.rst* files are binary files containing information about the nodes and elements of the FE model along with the results. *Coustyx* extracts the geometry of the mesh from these files.

5.1.1.4. I-DEAS Universal *.unv files

I-DEAS Universal file formats are widely used in structural dynamics, noise and vibrations community. *Coustyx* provides translator to extract geometry information of a mesh from these files.

5.1.2. Loading Frequency Response Data

Select: **Model** → **Structures** → <*Structure Mesh Name*>.

Right-click on <*Structure Mesh Name*> and select the frequency response data from any of the file format options through **Load Freq. Response Data** (refer to Figure 5.2). If the data is read successfully, a new dialog box with a list of frequencies available in the data appears. See Figure 5.3. The user can select what frequencies to load into the model. Use **Select All**, **Unselect All**, or **Select by Frequency Range...** buttons to choose the frequencies of interest. To clear the data already uploaded select **Clear Freq. Response Data** on the right-click menu. Run forced response FE analysis on the structure in the desired frequency range to generate the frequency response data. The frequency response data (or structure velocity) could later be used as a boundary excitation on the BE model (which is generated from the structure FE mesh). More details on how to define and apply boundary conditions in *Coustyx* are provided in Chapter 6. The following are the supported frequency response data formats.

5.1.2.1. Nastran OP2 File

The *OP2* (Output2) file is a binary output file from the forced response analysis of a FE mesh in NASTRAN. It contains the output acceleration, velocity or displacement at all the nodes of the FE mesh. *Coustyx* extracts the frequencies and the velocity response at these frequencies from this file. Note: To create an OP2 file that can be read by *Coustyx*, set the value of the parameter **POST** in the Nastran Bulk Data to **-1**.

```
PARAM, POST, -1
```

5.1.2.2. Nastran Punch File

The NASTRAN *punch file* also contains the output from the NASTRAN FE analysis, similar to the *OP2 file*, but in *ASCII* format. *Coustyx* extracts the frequencies and the velocity response at these frequencies from this file.

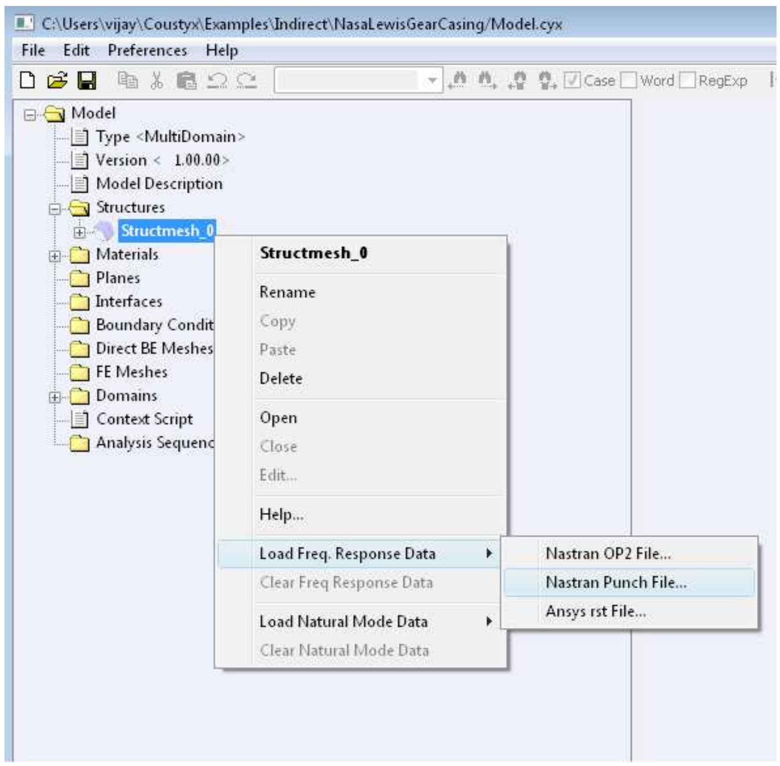


Figure 5.2: Loading frequency response data into *Coustyx* model.

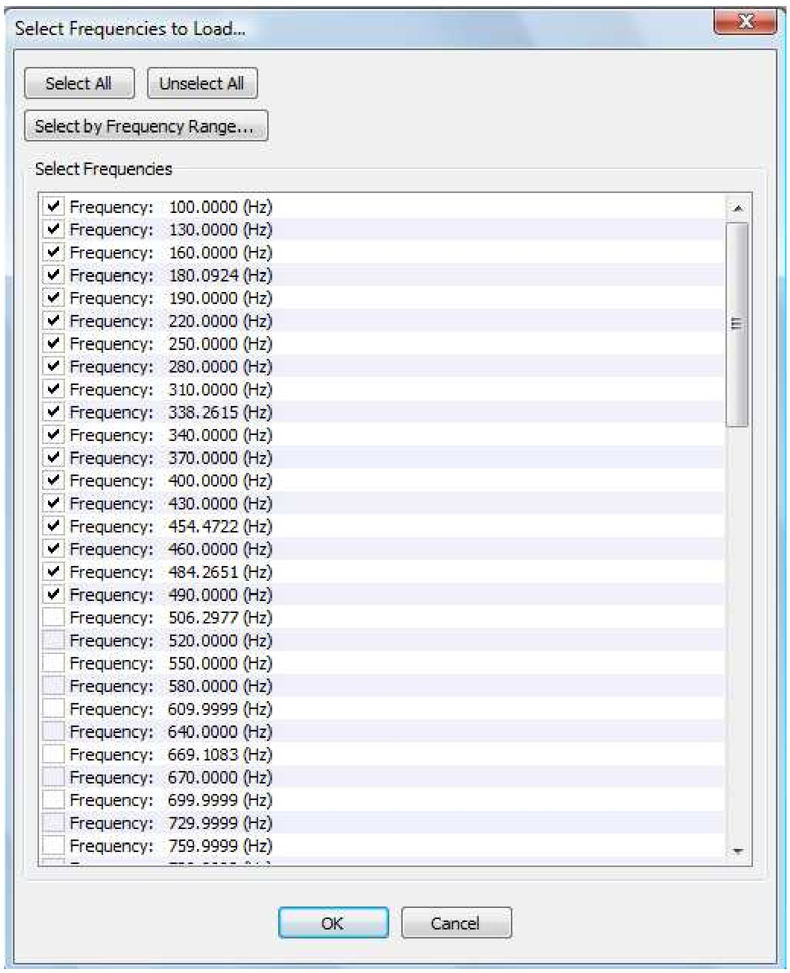


Figure 5.3: Select frequencies to load dialog box.

5.1.2.3. Ansys rst File

Coustyx can retrieve the frequency response data from ANSYS results file in **.rst* format.

5.1.2.4. Ansys rfrq File

Coustyx can retrieve the frequency response data from ANSYS results file in **.rfrq* format.

5.1.2.5. I-DEAS Universal File

Coustyx provides translator to retrieve the frequency response data from files in I-DEAS Universal file format.

5.1.2.6. Abaqus Output File

The ABAQUS **.odb* file is the output database file from the ABAQUS FE analysis. Currently, *Coustyx* does not provide a translator for this data format. However, the user can read the data by converting the ABAQUS output (**.odb*) file to the NASTRAN Output2 (*OP2*) file format using the ABAQUS command (refer [1]):

abaqus toOutput2 job=*job-name*

where *job-name* is the name of the ABAQUS output database file and also the name of the newly created NASTRAN OP2 file.

Note that the results from ABAQUS are written to the NASTRAN *OP2* file only when applicable records exist in the ABAQUS output database (**.odb*) file. To ensure the results to be translated are available in an ABAQUS output database file, include the following *OUTPUT options in the ABAQUS input file (refer [1]):

```
*OUTPUT, FIELD
*NODE OUTPUT
U,
RF,
CF,
*ELEMENT OUTPUT
S,
E,
SF,
NFORC,
```

Other ABAQUS results not specified above are skipped during translation.

5.1.3. Loading Natural Mode Data

Select: **Model** → **Structures** → <*Structure Mesh Name*>.

Right-click on the desired structure mesh <*Structure Mesh Name*> and select **Load Natural Mode Data** to load the natural modes data from a finite element analysis (refer to Figure 5.4).

If the data is read successfully, a new dialog box with a list of natural frequencies available in the data appears. See Figure 5.5. The user can select what natural frequencies to load into the model. Use **Select All**, **Unselect All**, or **Select by Frequency Range...** buttons to choose the natural frequencies of interest. The imported natural mode data includes the natural frequencies and the corresponding mode shapes. These values could later be used to define acoustic excitation (as boundary conditions) on the BE mesh (which is generated from the structure FE mesh). The supported file formats are: *Nastran OP2 file*, *Nastran Punch File*, *Ansys Results File*, and *I-DEAS Universal File*. The natural mode data can be cleared by selecting **Clear Natural Mode Data** on the right-click menu.

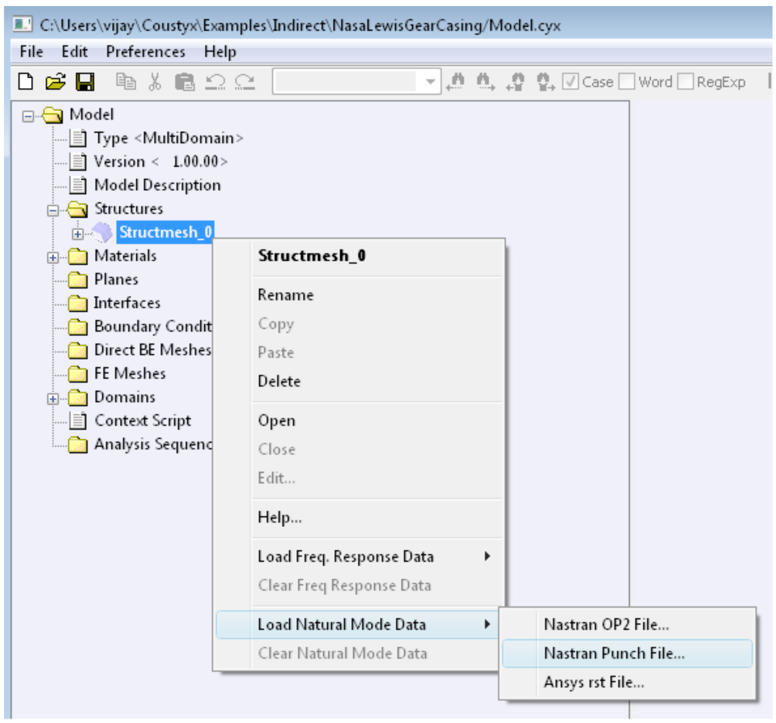


Figure 5.4: Loading natural mode data into *Coustyx* model.

5.1.3.1. Copy to Frequency Response Data

Select: **Model** → **Structures** → <*Structure Mesh Name*> → **Natural Mode Data** → <*Mode*>.

Right-click on the desired mode <*Mode*> and select **Copy to Freq Response Data** (refer to Figure 5.6). The mode information from the **Natural Mode Data** is copied to the **Freq Response Data**. This function is useful when the user wants to perform acoustic analysis over a frequency range with the current mode as the structural velocity excitation. A structure velocity boundary condition for the BE mesh is defined by specifying the *Structure Name* (refer

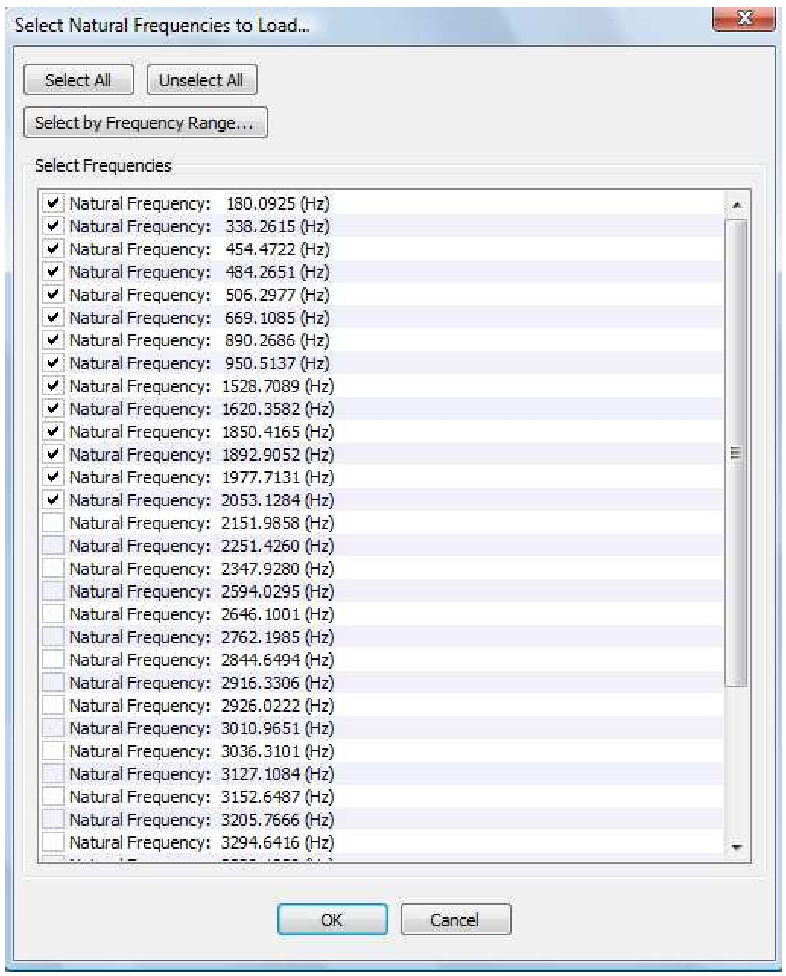


Figure 5.5: Select natural frequencies to load dialog box.

to Section 6.3.8 and Section 6.4.9). *Coustyx* uses the **Freq Response Data** of the specified *Structure Name* as the excitation. We can apply natural modes as the structural velocity excitation by copying the mode data to the frequency response data. Note that this action clears any existing frequency response data while copying the selected mode data (refer to Figure 5.7). Also note that only one mode could be copied to the frequency response data at a time. Once the model setup is completed, one can perform acoustic analysis to obtain relative responses at various frequencies due to the applied modal excitation. Absolute values are irrelevant. Radiation efficiency is one of the main results of interest in such an analysis.

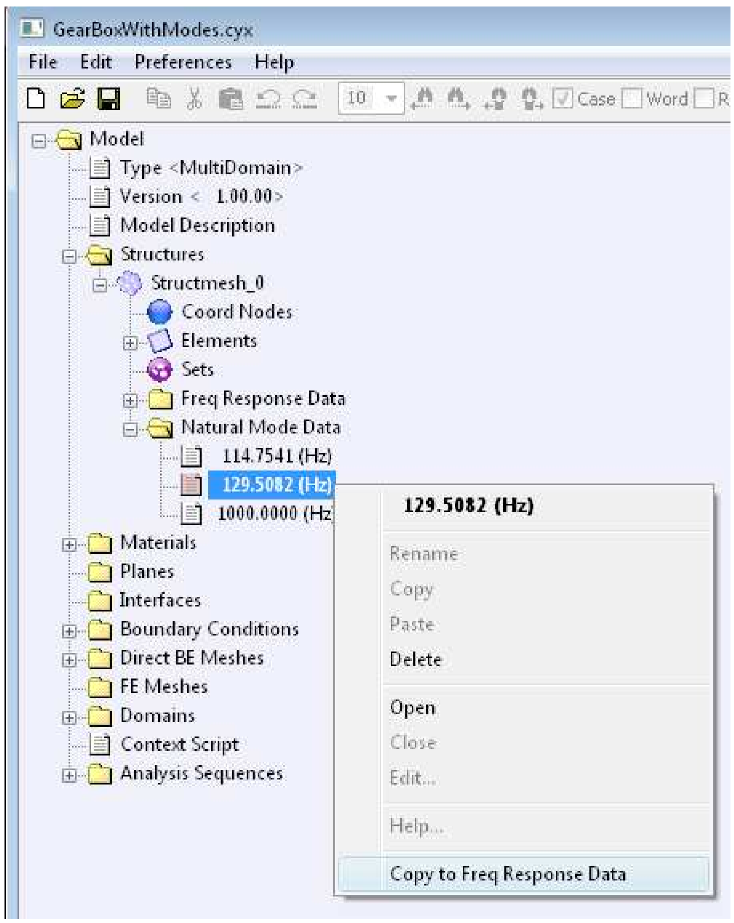


Figure 5.6: Copy the mode data to the frequency response data.

5.2. Exporting Mesh

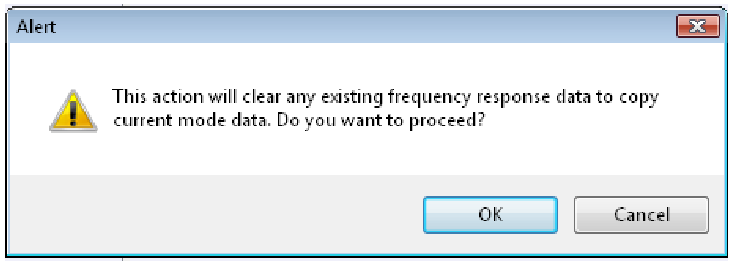


Figure 5.7: Alert message displayed before copying mode data to frequency response data.

Coustyx provides option to export a structure mesh or a boundary element mesh to a Nastran bulk data format. To export a mesh: Right-click on the *<Structure Mesh Name>* or *<Boundary Element Mesh Name>* and select **Export** → **Nastran Bulk Data (.bdf) File** (see Figure 5.8).

5.3. Forced Response Analysis using Modal Superposition

Select: **Model** → **Structures** → *<Structure Mesh Name>*.

The frequency response data (which is applied as the structure velocity boundary condition) could either be imported from an external finite element forced response analysis (see Section 5.1.2) or could be directly computed within Coustyx for the case when the natural modes of the structure are already available. Coustyx uses *modal superposition method* to compute the forced response analysis. Follow the steps below to compute the forced response using Coustyx:

- **Load natural mode data**. Load natural modes computed from any external finite element modal analysis. Note: Make sure the modes are ortho-normalized with respect to the finite element mass matrix.
- **Modify modal damping**. Specify non-zero damping to restrict infinite responses at natural frequencies.
- **Apply loads**. Apply nodal forces and/or moments prior to performing forced response analysis.
- **Perform modal superposition**. Select analysis frequencies and the participating modes to compute forced response.

5.3.1. Load Natural Mode Data

See Section 5.1.3 for details on how to load natural mode data. The natural modes are assumed to be ortho-normalized with respect to the mass matrix. That is, $\mathbf{A}^T\mathbf{M}\mathbf{A} = \mathbf{I}$ and $\mathbf{A}^T\mathbf{K}\mathbf{A} = \Omega^2$, where $\mathbf{A} = [\mathbf{a}_1\mathbf{a}_2...\mathbf{a}_n]$ is the modal matrix containing natural mode vectors $\mathbf{a}_i$, $\mathbf{M}$ and $\mathbf{K}$ are finite element mass and stiffness matrices, $\Omega^2 = \text{Diag}(\omega_1^2, ..., \omega_n^2)$, ω_i is the natural frequency, and $\mathbf{I}$ is a unit matrix.

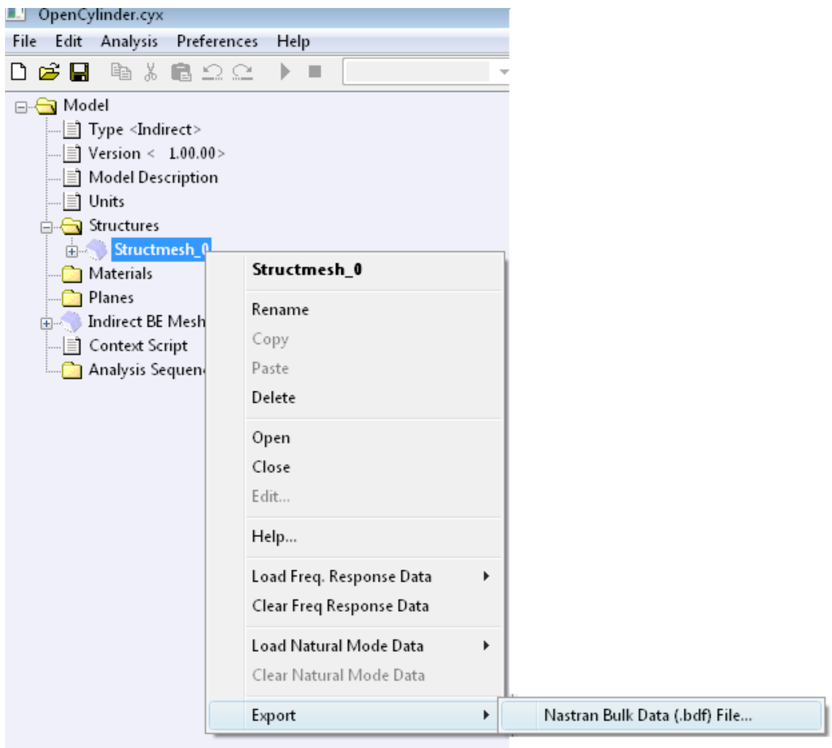


Figure 5.8: Export mesh.

5.3.2. Modify Modal Damping

Modal damping is the only damping model allowed in *Coustyx*. Modal damping is estimated from the damping ratios ζ. A damping ratio is a unitless ratio of the damping constant (c) to the critical damping constant (c_c), $\zeta = \frac{c}{c_c}$. Specify $\zeta = 1$ for critically damped systems, $\zeta < 1$ for underdamped systems, and $\zeta > 1$ for overdamped systems. A non-zero modal damping ratio is necessary to restrict infinite responses at natural frequencies. The relation between damping matrix $\mathbf{C}$ and damping ratios is, $\mathbf{A}^T\mathbf{C}\mathbf{A} = \text{Diag}(..., 2\zeta_i\omega_i, ...)$, where $\mathbf{A}$ is the modal matrix, ζ_i and ω_i are damping ratio and natural frequency at the i^{th} mode.

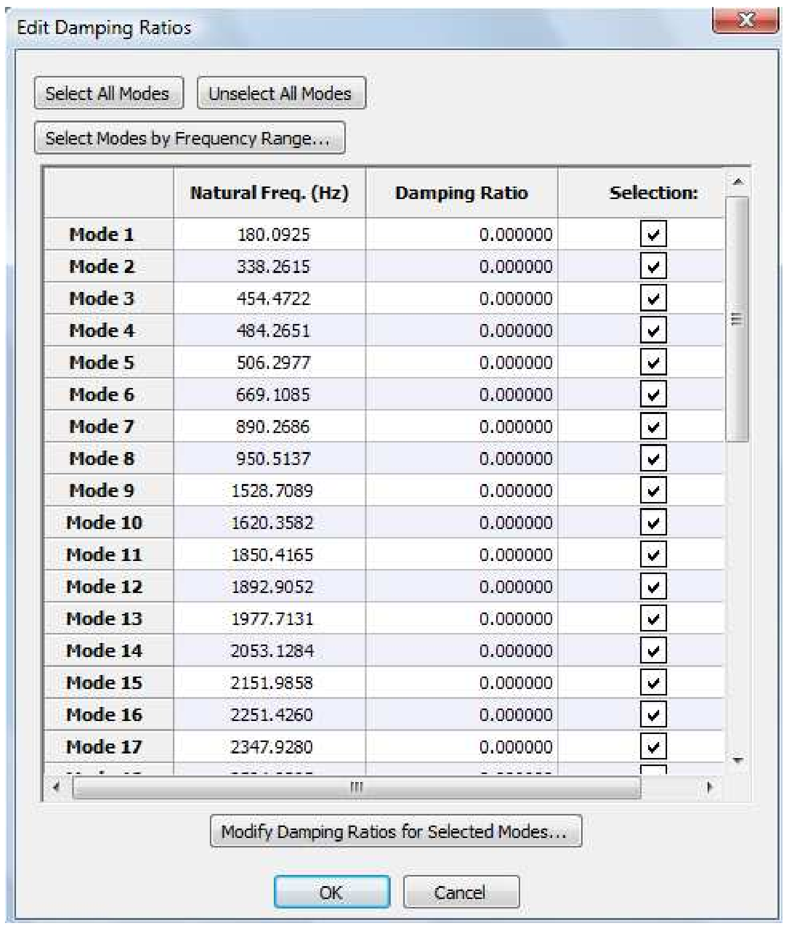


Figure 5.9: Damping ratios edit dialog box.

Follow the instructions below to edit modal damping ratios.
Select: **Model** → **Structures** → <*Structure Mesh Name*> → **Natural Mode Data**.
Right-click on **Natural Mode Data** and select **Edit Damping Ratios...**. A dialog window

shown in Figure 5.9 appears. It contains a table that lists modes, their natural frequencies, damping ratios, and selection (check) boxes. The user can modify the damping ratios individually by directly editing the table. The damping ratios for a group of selected modes can be edited by pressing the button **Modify Damping Ratios for Selected Modes...**.

There are multiple ways to select modes in the table for editing: (a) check or uncheck the selection box against a mode in the table, (b) use buttons **Select All Modes** or **Unselect All Modes** to select or unselect all modes in the table, (c) use button **Select Modes by Frequency Range...** to select modes lying within a frequency range (Figure 5.10). Set the upper and lower frequency limits and press OK to select modes within the frequency limits.

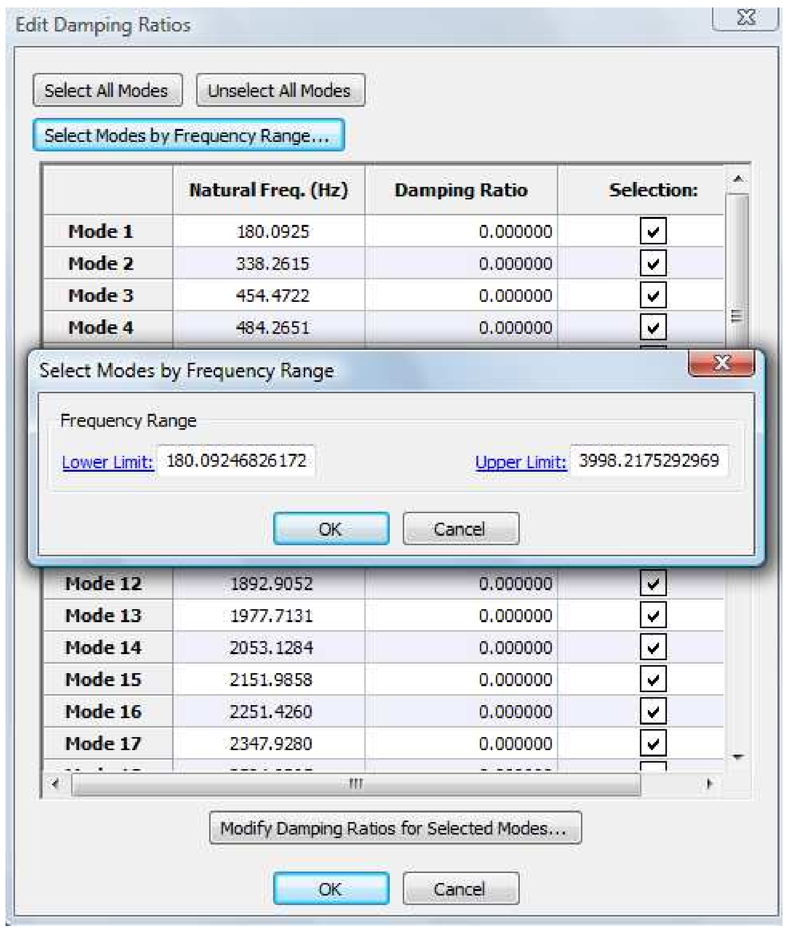


Figure 5.10: Select modes by frequency range.

The button **Modify Damping Ratios for Selected Modes...**, when pressed, opens up a dialog box as shown in Figure 5.11. Select a method from the **Specification** drop-down menu to compute damping ratios for selected modes:

Frequency Dependent Damping Ratio Specify the **Frequency Dependent Damping Ratio** through any of the frequency dependent types: *Constant*, *Table*, or *Script*. See Figure 5.11. For example, to define a 5% damping ratio for selected modes, select the frequency dependence type to be *Constant* and enter a value of 0.05. You can also specify a *Table* or a *Script* to define damping ratios varying with natural frequency. For example, a natural frequency dependent damping ratio can be defined by a script that returns the following: `return(Eval(0.05*Frequency))`. Note: the predefined variables **Frequency** or **AngularFreq** in such scripts or tables represent *natural frequencies*. Press OK to compute damping ratios and edit the table for selected modes.

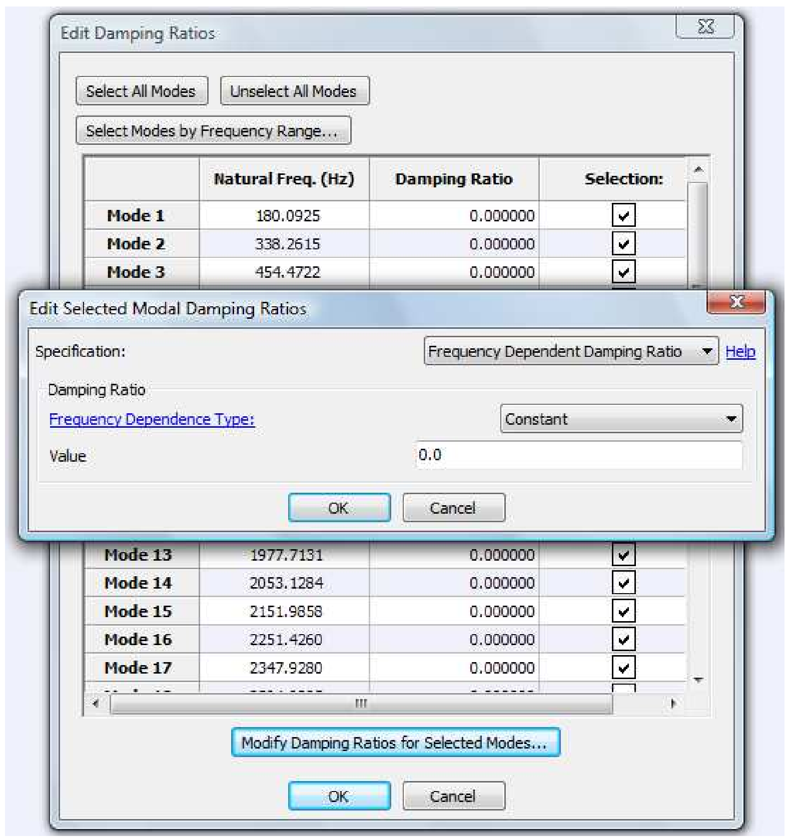


Figure 5.11: Edit selected modal damping ratios using Frequency dependent values.

Rayleigh Damping Rayleigh damping coefficents, **Alpha** α and **Beta** β are defined through this option. See Figure 5.12. Damping ratio for each selected mode is computed from α and β as follows:

$$\zeta_i = \frac{\alpha}{2\omega_i} + \frac{\beta\omega_i}{2}$$

where ζ_i is damping ratio and ω_i is the natural frequency at i^{th} mode. The above equation is derived from the definition of modal damping and the relation between rayleigh damping coefficients and the damping matrix: $\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$, where $\mathbf{C}$ is the damping matrix, $\mathbf{M}$ and $\mathbf{K}$ are mass and stiffness matrices.

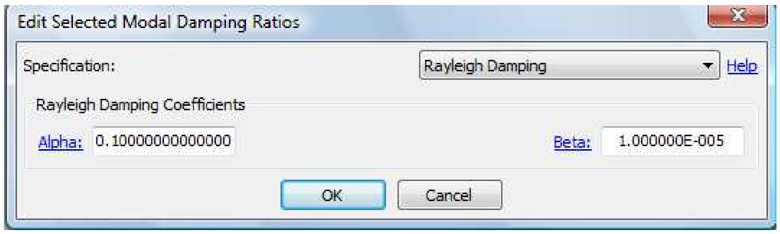


Figure 5.12: Edit selected modal damping ratios using Rayleigh damping coefficients.

5.3.3. Apply Loads

Select: **Model** → **Structures** → <*Structure Mesh Name*> → **Loads**.

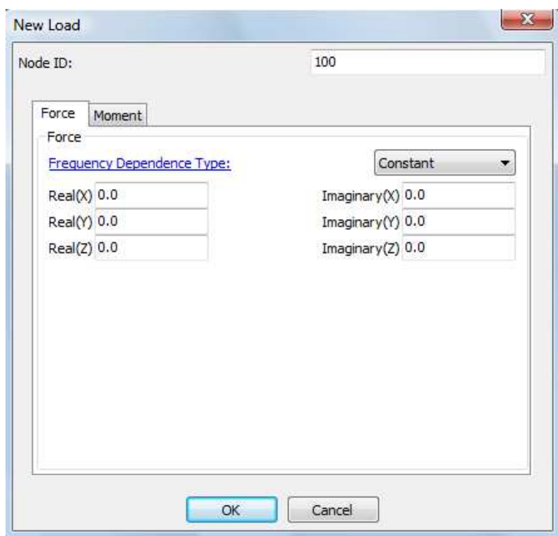


Figure 5.13: Apply nodal force and moment.

Apply non-zero forces and/or moments to perform forced response analysis. Note that *Coustyx* presently allows to only specify loads on the nodes. A nodal load is identified by the node id and the force and/or moment applied at that node. To specify a new load, right-click on **Loads** and select **New**. A dialog box as shown in Figure 5.13 appears. Enter a valid node id and edit the frequency dependent variables: **Force** and **Moment**. Press OK to create a new load. If the node id is not present in the structure mesh or has already been used to define a load *Coustyx* prompts the user to change the node id. Specify the force and moment through any of the frequency dependence types: *Constant*, *Table*, or *Script*. Note that the predefined variables **Frequency** or **AngularFreq** in the script or table represent *analysis frequency*.

To edit an existing load, right-click on the load in the model tree and select **Edit**. To delete existing loads, right-click on a load and select **Delete**, or right-click on **Loads** and select **Delete All Loads**.

5.3.4. Perform Modal Superposition

Modal superposition method decouples the equations of motion using mode shapes and natural frequencies. The decoupled system of equations that are solved in *Coustyx* are given below (Note: $e^{-j\omega t}$ convention is used):

$$-\omega^2\mathbf{A}^T\mathbf{M}\mathbf{A}\mathbf{u} - j\omega\mathbf{A}^T\mathbf{C}\mathbf{A}\mathbf{u} + \mathbf{A}^T\mathbf{K}\mathbf{A}\mathbf{u} = \mathbf{A}^T\mathbf{f}$$

or,

$$-\omega^2\mathbf{I}\mathbf{u} - j\omega Diag(..., 2\zeta_i\omega_i, ...)\mathbf{u} + \Omega^2\mathbf{u} = \mathbf{A}^T\mathbf{f}$$

where ω is the analysis frequency, $\mathbf{A} = [\mathbf{a}_1\mathbf{a}_2...\mathbf{a}_n]$ is the modal matrix with natural modes $\mathbf{a}_i$, $\mathbf{M}$ is finite element mass matrix, $\mathbf{C}$ is the damping matrix, $\mathbf{K}$ is the stiffness matrix, $\Omega^2 = \text{Diag}(\omega_1^2, \omega_2^2, ..., \omega_n^2)$, ω_i is the natural frequency, $\mathbf{u}$ is the generalized displacement vector, $\mathbf{f}$ is the force vector, ζ_i is the modal damping ratio, and $\mathbf{I}$ is a unit matrix.

The above decoupled system of equations is solved for the generalized displacements $\mathbf{u}$. The physical response (displacement $\mathbf{x}$ or velocity $\dot{\mathbf{x}}$) is then computed from the generalized displacement $\mathbf{u}$ and modal matrix $\mathbf{A}$ as,

$$\mathbf{x} = \mathbf{A}\mathbf{u}$$

and,

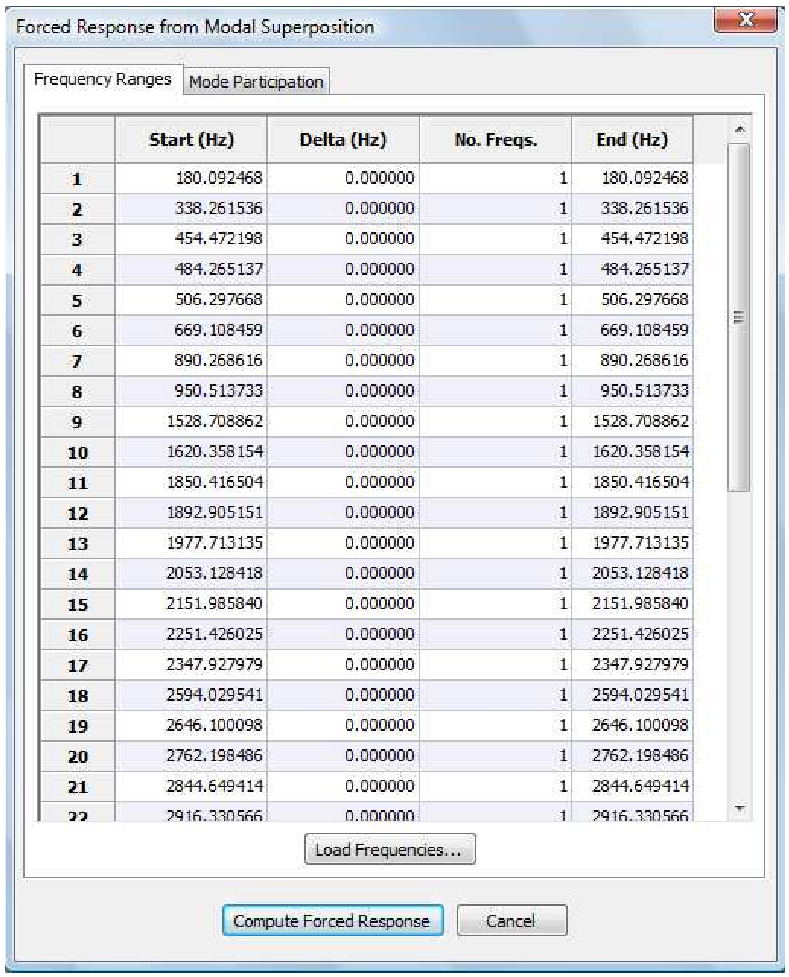
$$\dot{\mathbf{x}} = -j\omega\mathbf{A}\mathbf{u}$$

Follow the instructions below to perform *forced response harmonic analysis using modal superposition* in *Coustyx*.

Select: **Model** → **Structures** → <*Structure Mesh Name*> → **Freq Response Data**.

Right-click on the model tree member **Freq Response Data** and select **Compute using Modal Superposition...**. Note that this option is active only when **Natural Mode Data** and **Loads** are already specified. A dialog window as shown in Figure 5.14 appears. Enter analysis frequencies using the table in the **Frequency Range** tabbed window. Refer Section 7.1.2 for more details about the frequency range table. Select participating modes from the list in the **Mode Participation** tabbed window. See Figure 5.15. There are multiple ways to select modes: (a) check or uncheck the selection box against a mode in the list, (b) use buttons **Select All Modes** or **Unselect All Modes** to select or unselect all modes, (c) use the button **Select Modes by Frequency Range...** to select modes lying within a frequency range. Set the upper and lower frequency limits and press OK to select modes within the frequency limits.

Once the analysis frequencies and the participating modes are selected, press the button **Compute Forced Response** to start the forced response harmonic analysis using modal superposition. The analysis results are stored into the model tree member **Freq Response Data**. This data could later be applied as a structure velocity boundary condition for acoustic analysis.



	Start (Hz)	Delta (Hz)	No. Freqs.	End (Hz)
1	180.092468	0.000000	1	180.092468
2	338.261536	0.000000	1	338.261536
3	454.472198	0.000000	1	454.472198
4	484.265137	0.000000	1	484.265137
5	506.297668	0.000000	1	506.297668
6	669.108459	0.000000	1	669.108459
7	890.268616	0.000000	1	890.268616
8	950.513733	0.000000	1	950.513733
9	1528.708862	0.000000	1	1528.708862
10	1620.358154	0.000000	1	1620.358154
11	1850.416504	0.000000	1	1850.416504
12	1892.905151	0.000000	1	1892.905151
13	1977.713135	0.000000	1	1977.713135
14	2053.128418	0.000000	1	2053.128418
15	2151.985840	0.000000	1	2151.985840
16	2251.426025	0.000000	1	2251.426025
17	2347.927979	0.000000	1	2347.927979
18	2594.029541	0.000000	1	2594.029541
19	2646.100098	0.000000	1	2646.100098
20	2762.198486	0.000000	1	2762.198486
21	2844.649414	0.000000	1	2844.649414
22	2916.330566	0.000000	1	2916.330566

Figure 5.14: Perform modal superposition.

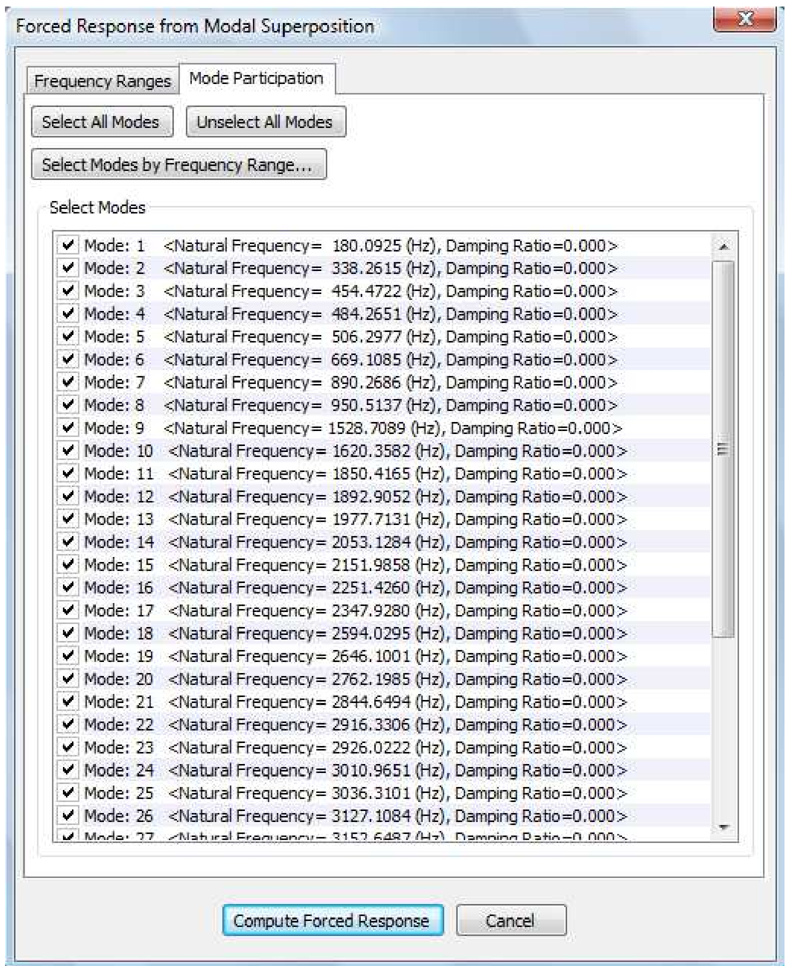


Figure 5.15: Participating modes selection.

5.4. Fill Hole

The **Fill Hole** *tabbed* window is located at the bottom pane of the *Mesh Viewer* window (Figure 5.16). *Coustyx* uses an *Optimized Delaunay Triangulation Method* to fill holes. The parameters used are shown in Figure 5.17. First, we will explain each of the parameters in the Figure 5.17, and then discuss the procedure to fill a hole. (Figure 5.16)

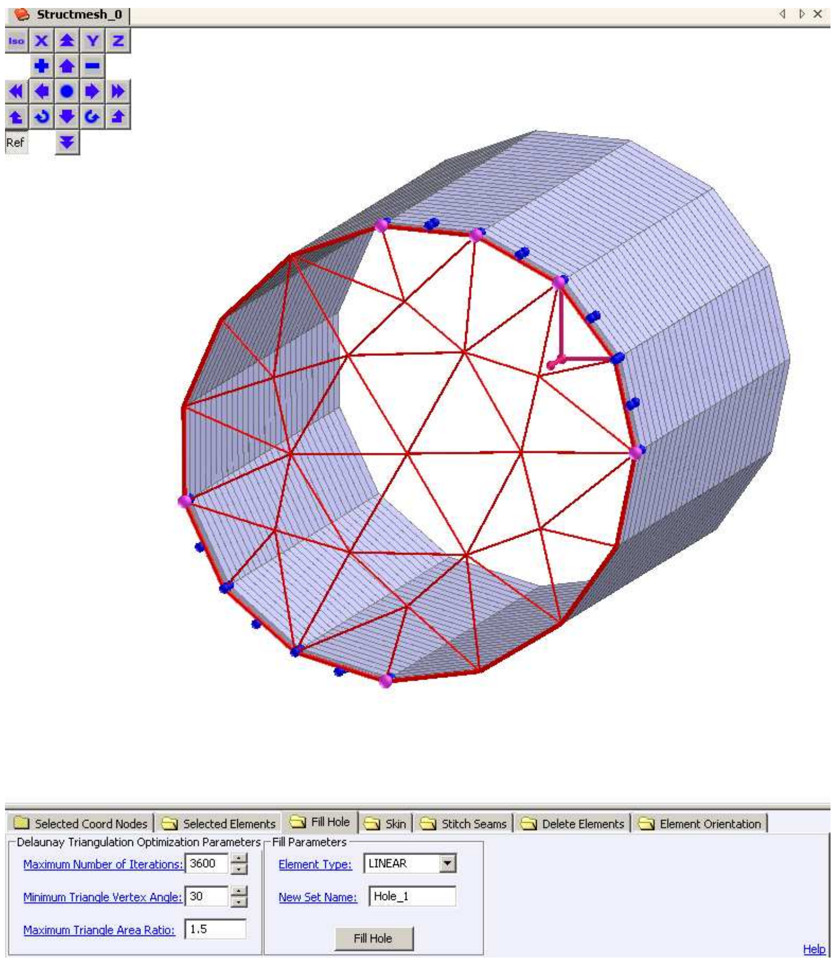


Figure 5.16: Coustyx GUI - Fill Hole.

5.4.1. Delaunay Triangulation Optimization Parameters

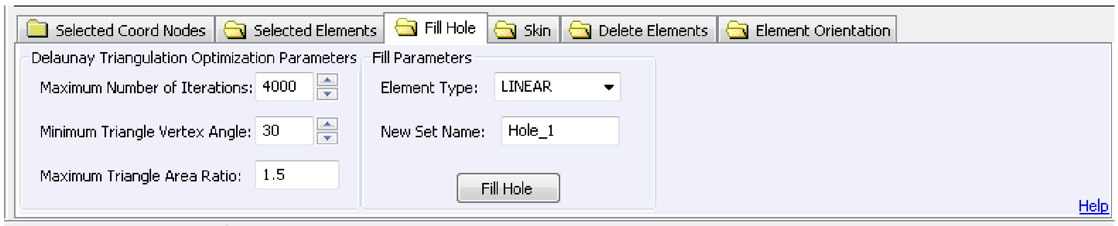


Figure 5.17: Fill Hole Parameters.

The parameters used by the *Delaunay Triangulation* method can be modified here. The default set of parameters provided by *Coustyx* are sufficient for most of the cases. *Coustyx* goes through number of iterations to optimize the shapes of triangles in the hole while satisfying the limits defined by the parameters **Minimum Triangle Vertex Angle**, and **Maximum Triangle Area Ratio**. (Figure 5.17)

Maximum Number of Iterations The maximum number of iterations allowed before terminating the *Delaunay Triangulation* method.

Minimum Triangle Vertex Angle The minimum vertex angle of the triangle allowed. If the triangle element formed by *Delaunay Triangulation* method has a vertex angle smaller than this, that is if the element is skinny, the algorithm calls for more iterations to optimize the mesh.

Maximum Triangle Area Ratio The maximum ratio of the largest to the smallest triangle areas allowed. If the triangle elements formed by *Delaunay Triangulation* method have the maximum triangle area ratio higher than the specified parameter, the algorithm calls for more iterations to optimize the mesh.

5.4.2. Fill Parameters

Element Type This option specifies the type of triangle elements to be created to fill the hole. *Linear* or *Quadratic* triangle elements can be created.

New Set Name All the triangle elements created to fill the hole can be conveniently added to a Set during the process. This option specifies the name of the new Set.

Baffle Plane Option This option is available only for *Indirect* BE meshes in *Indirect* BE models. When selected, the elements created to fill the hole are considered to be on a baffle plane.

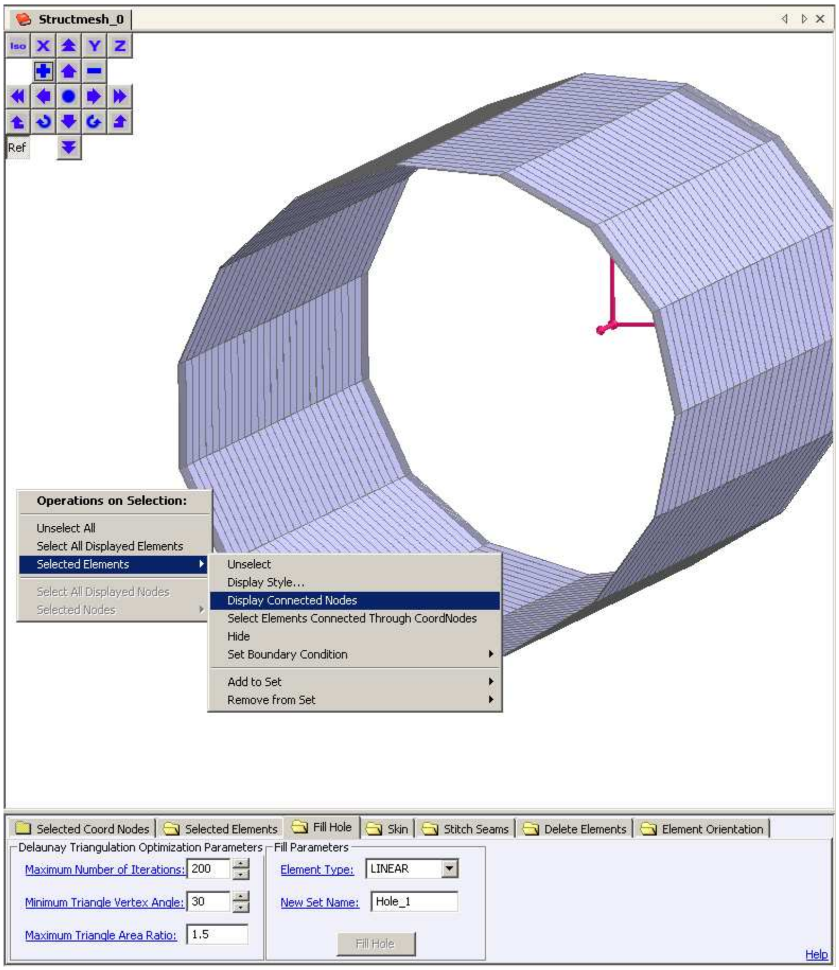


Figure 5.18: Display Connected Nodes

5.4.3. Procedure to Fill Hole

Follow these steps to fill a hole:

- Open any mesh in the GUI by right-clicking on the mesh and selecting **Open**.
- Set the parameters for hole filling in the **Fill Hole** *tabbed* window. Refer to the Section 5.4.1 and Section 5.4.2 to set *Delaunay Triangulation Optimization Parameters* and *Fill Parameters* (Refer to Figure 5.17).
- Select few elements whose edges coincide with the hole edge. To select an element in the GUI: Left-click on the element with the *shift-key* held down.
- Display connected nodes for the selected elements. Right-click on the selected elements with the *shift-key* held down and select: **Operations on Selection** → **Selected Elements** → **Display Connected Nodes**. (Figure 5.18)
- The *Delaunay Triangulation* method requires at least three nodes to be identified on the edge of the hole to auto-fill the hole with triangle elements. (Refer to Figure 5.16)
- Choose three or more nodes to form a unique closed loop on the edge of the hole. To select a node: Left-click on the node with the *shift-key* held down. Note: You should only select nodes on the corners of the elements, not mid-side nodes. Once a closed loop is identified *Coustyx* auto fills the hole with outlines of triangle elements for preview.
- Select the type of triangle elements to be filled in the hole: **Fill Hole** → **Element Type** → **LINEAR** or **QUADRATIC**.
- Rename the Set: **Fill Hole** → **New Set Name** → <*Set Name*>.
- Accept the triangulation of the hole by pressing **Fill Hole** → **Fill Hole** button at the bottom of the window. (Refer to Figure 5.17)

5.5. Skin

Coustyx has the ability to skin the 3-D FE structure mesh to create a 2-D surface mesh used for BEA. The *Manipulation Task Function* **Skin** is located in a *tabbed* window at the bottom of the *Mesh Viewer* window (refer to Figure 5.26).

Coustyx offers the option to specify seams to avoid skinning the unwanted regions of a FE Mesh. More details on how to create a seam are given in Section 5.5.3. Once the seams are created, the structure mesh can be skinned to obtain surface mesh. For structure components with zero thickness, such as ribs, there is no need to create seams on the edges. *Coustyx* identifies these 2-D elements and creates surface mesh without duplication in the *Indirect* BE model. However, ribs cannot be skinned in a *MultiDomain* model. For a *MultiDomain* model if the skin is allowed to propagate to both sides of a 2-D element no surface mesh is generated.

Use any of the other *Manipulation Task Functions* to visually inspect the FE mesh, fix any inconsistencies, etc., before skinning.

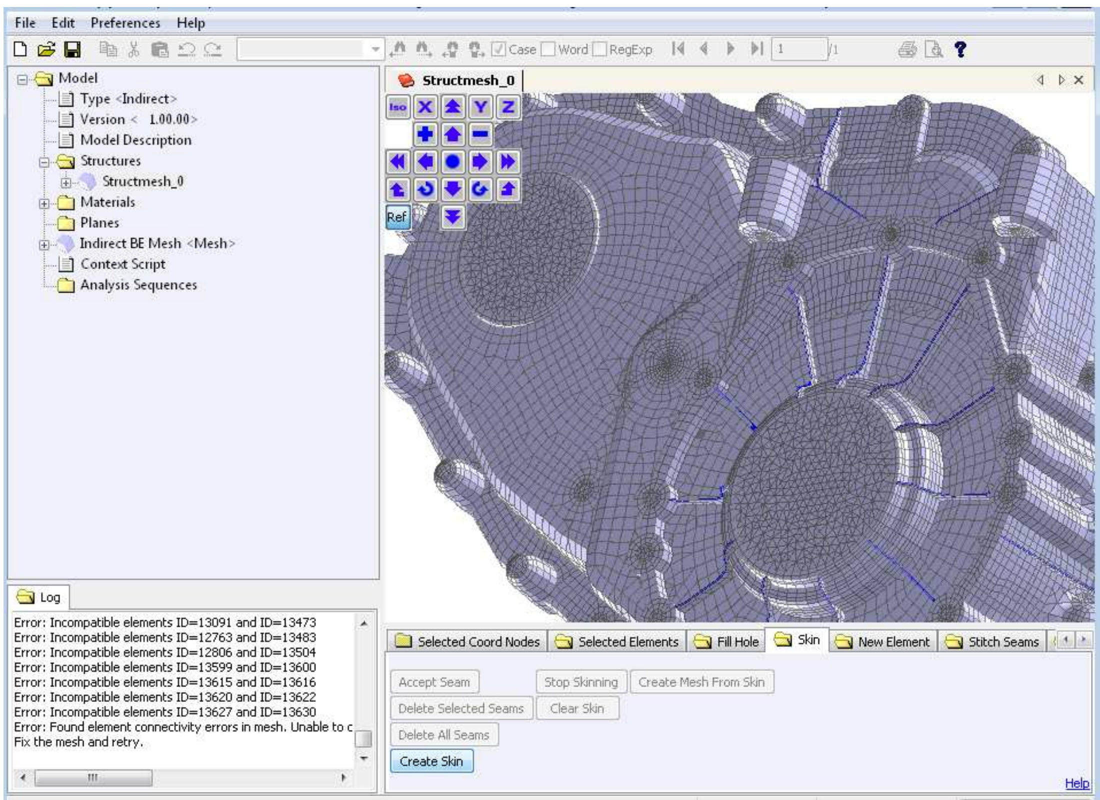


Figure 5.19: Press **Skin** → **Create Skin** to check for incompatible elements.

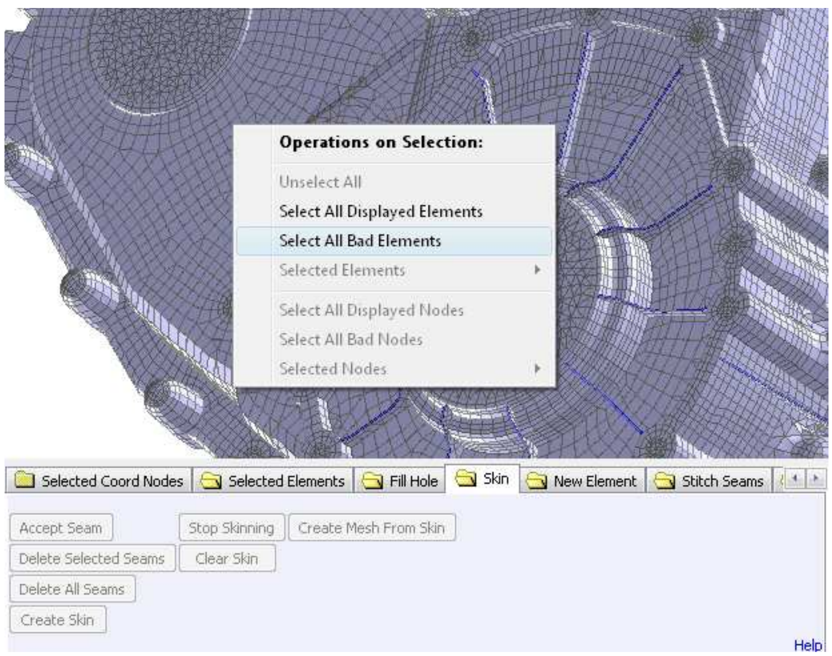


Figure 5.20: Select all Bad Elements.

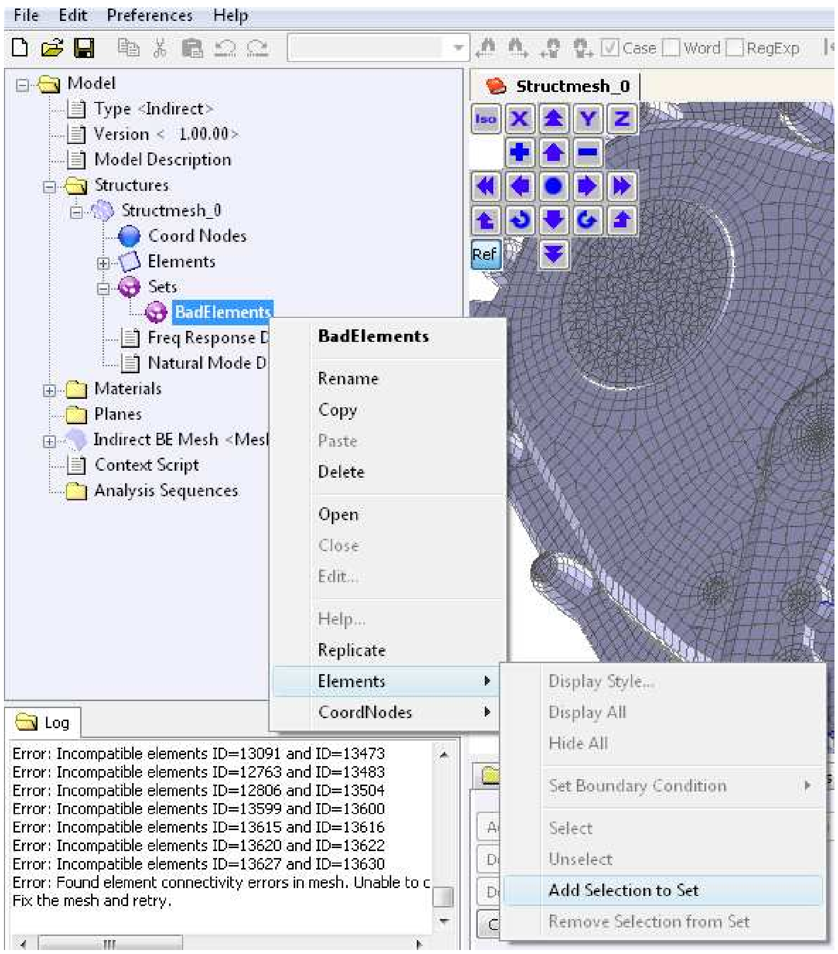


Figure 5.21: Add Bad Elements to a Set.

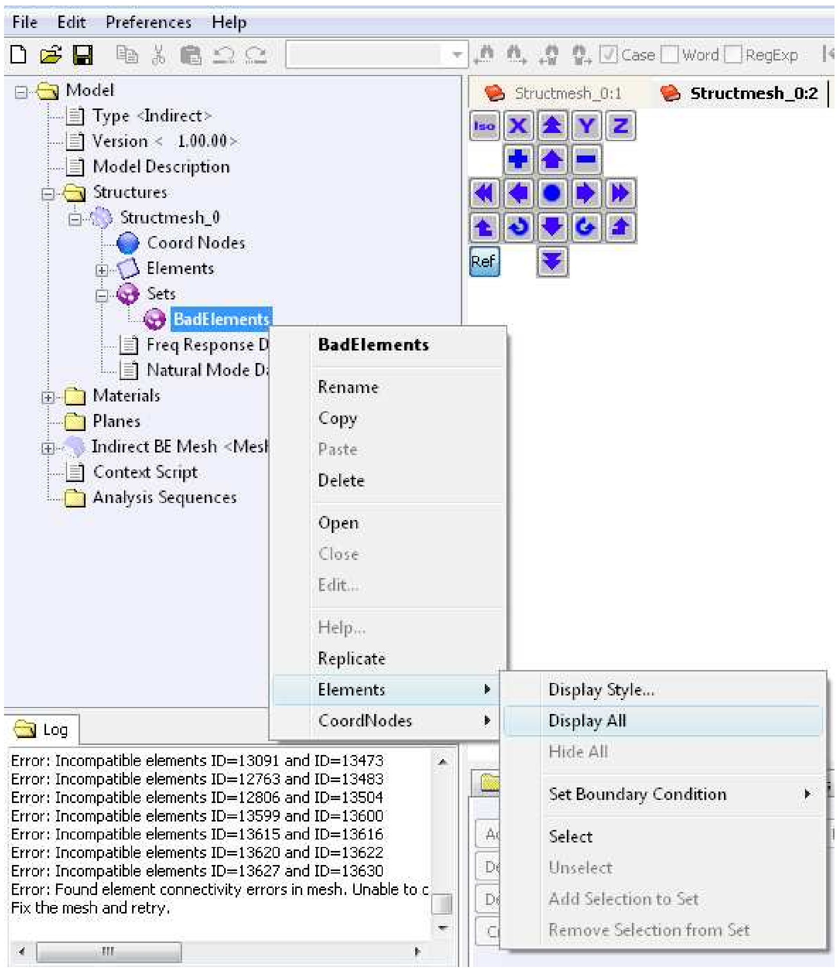


Figure 5.22: Display Set of Bad Elements.

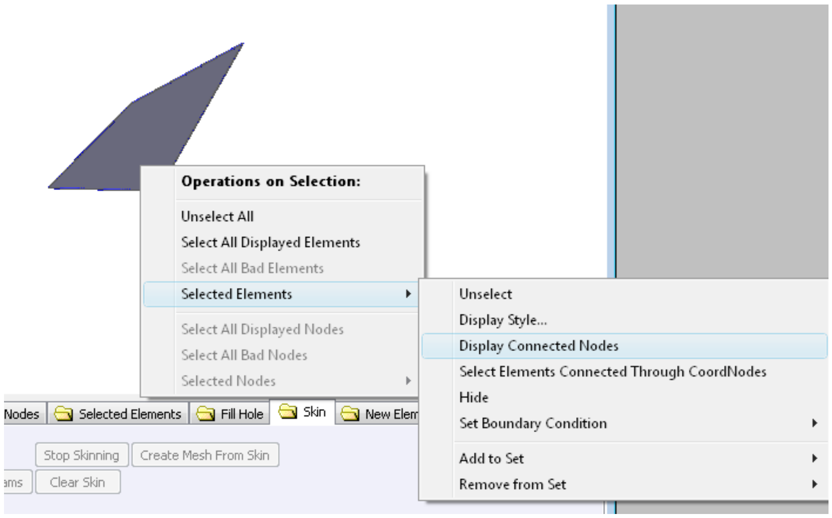


Figure 5.23: Display Connected Nodes of Selected Element.

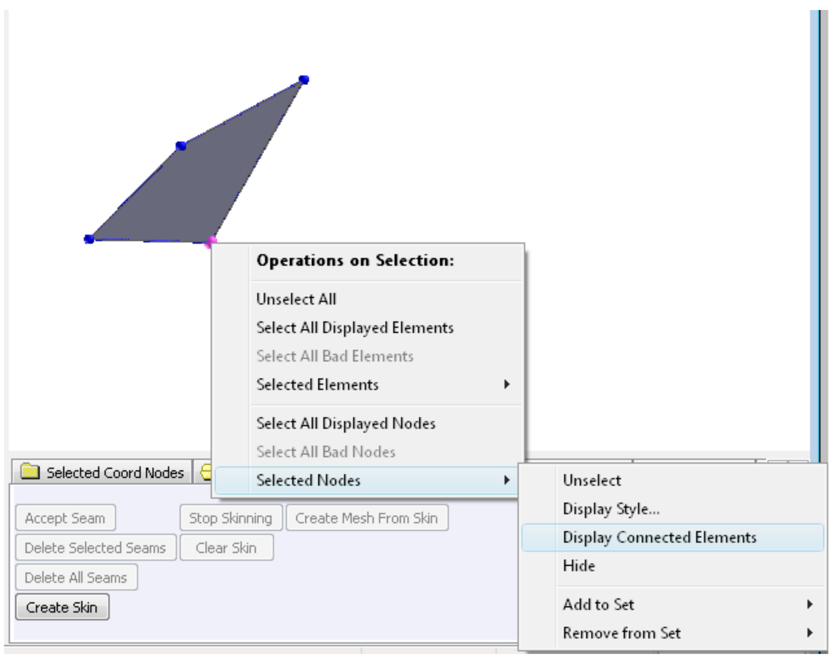


Figure 5.24: Display Connected Elements of Selected Node.

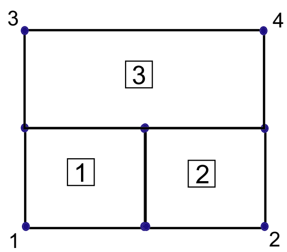
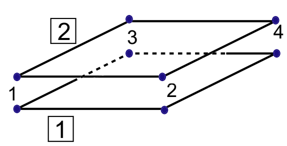
5.5.1. Treatment of Bad Elements

Coustyx performs consistency checks before skinning the FE mesh. Press **Skin → Create Skin** button to start these checks. If there are no inconsistencies in the FE structure mesh then *Coustyx* proceeds with the skinning process. However, when the FE mesh has elements with bad coordinate connectivity, *Coustyx* throws error messages in the log window. The error messages show the IDs of the incompatible elements. These Bad Elements have inconsistent coordinate connectivity with respect to their neighboring elements. The user needs to manually fix the mesh by treating the Bad elements appropriately.

The steps to be followed to treat Bad elements in a FE mesh are:

- Press **Skin → Create Skin** button. If the FE structure mesh has elements with bad coordinate connectivity then *Coustyx* throws error messages in the log window and asks the user to fix these Bad Elements before attempting to skin again. (Figure 5.19)
- To unselect all displayed elements in the GUI, right-click with *shift-key* held down and select: **Operations on Selection → Unselect All**.
- Select all Bad Elements in the FE mesh by right-clicking on the GUI while holding down the *shift-key* and choosing **Operations on Selection → Select all Bad Elements**. (Figure 5.20)
- Create a new set by choosing **Model → Structures →** <*Structure Mesh Name*> **→ Sets**. Right-click on **Sets** and select **New**. Rename the new set to <*Bad Elements Set*>. To add Bad Elements right-click on <*Bad Elements Set*> and select **Elements → Add Selection to Set**. (Figure 5.21)
- Go to **Model → Structures →** <*Struct Mesh Name*> **→ Elements**. Right-click and select **Hide All** to hide all the elements in the GUI.
- Now display only Bad Elements to start fixing them. Go to **Structures →** <*Structure Mesh Name*> **→ Sets →** <*Bad Elements Set*> and right-click to select **Elements → Display All**. (Figure 5.22)
- Select a Bad Element by left-clicking on it while holding down the *shift-key*. To display coordinate nodes of this element, right-click with *shift-key* held down and select: **Operations on Selection → Selected Elements → Display Connected Nodes**. (Figure 5.23)
- Select any one of the coordinate nodes of the Bad Element by left-clicking on the node while holding down the *shift-key*. Then right-click and select: **Operations on Selection → Selected Nodes → Display Connected Elements** to display all the elements connected to that node (neighboring elements). (Figure 5.24)
- Check how the coordinate connectivity of the Bad Element is not compatible with its neighbors. Treat the Bad Element appropriately based on the kind of incompatibility. Some of the most common types of Bad Elements found in FE meshes are listed in the Table 5.1 along with valid treatments. If required delete the Bad Element (or its neighboring elements) using the *Manipulation Task Function* **Delete Elements**. New elements can be created using the *Manipulation Task Function* **New Element**.

Table 5.1: Some of the common types of Bad Elements and their Treatment.

Bad Elements	Treatment
	• Incompatible coordinate connectivity between elements 3 and 1(2). • Delete elements 1, 2 and 3. • Create a new element using the coordinates 1, 2, 3 and 4.
	• Overlapping elements 1 and 2. These elements have the same coordinate nodes 1, 2, 3 and 4. • Delete any one of the overlapping elements.

5.5.2. Free Edges in a FE Mesh

While in the **Skin** *tabbed* window, *Coustyx* displays all the *Free Edges* in the FE mesh in color *Blue*. (Figure 5.25)

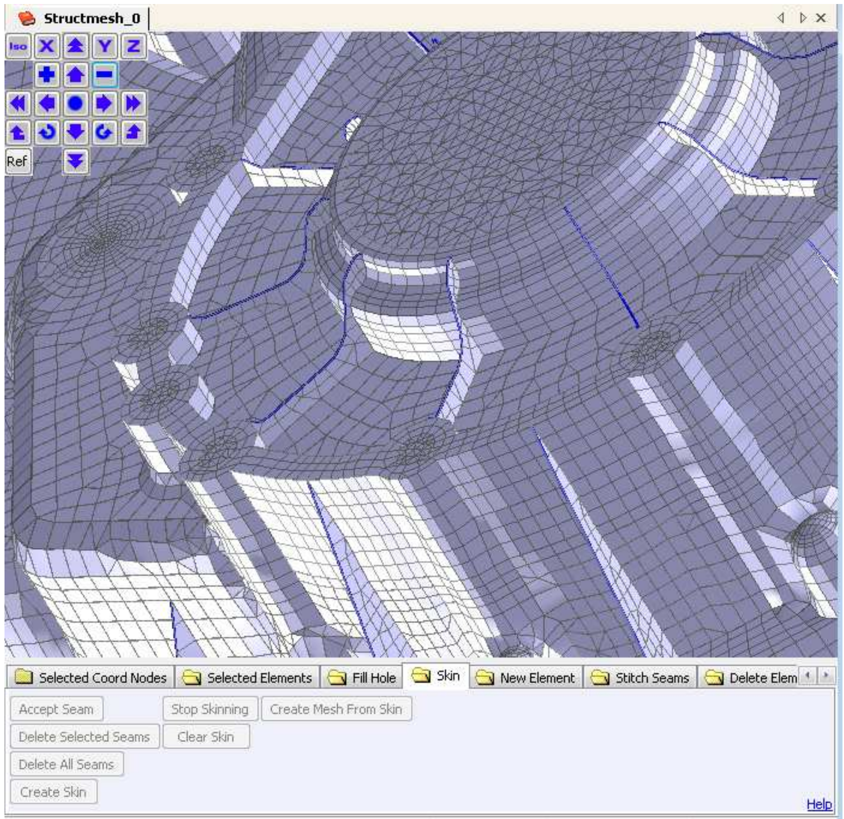


Figure 5.25: Mesh with free edges shown in *blue* (color).

5.5.3. Create Seam

Seams are demarcation lines which stop the propagation of skin beyond them. Listed below are some examples where seams are created to skin only the desired side of a mesh.

- A seam around a hole edge will avoid skinning through the hole averting the propagation of the skin to the other side of the mesh.
- Another example where seams are useful is in case of a FE mesh representing a symmetrical geometry. In that case the FE mesh doesn't enclose a closed volume and to skin only one side of the mesh we have to specify seams at the mesh edges on the symmetry planes.

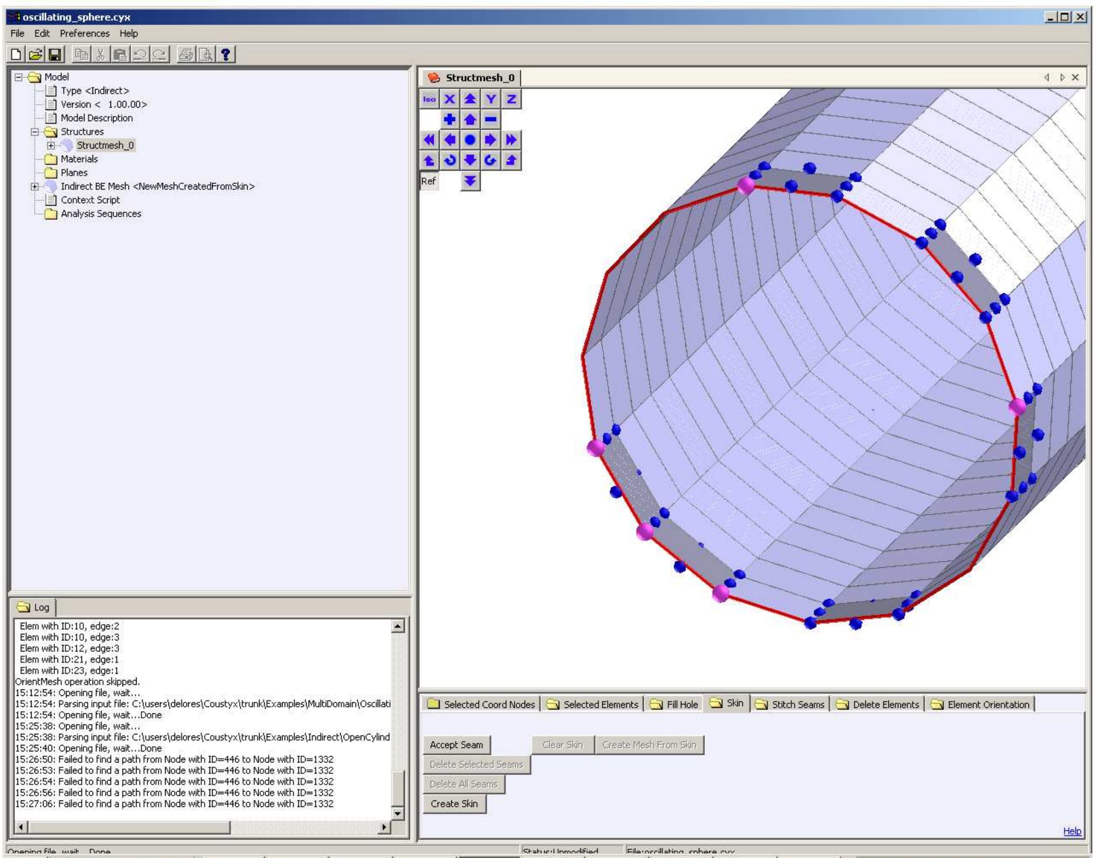


Figure 5.26: Create Seams.

5.5.3.1. Procedure to Create Seams

Follow the steps below to create a seam:

- Select a few of the elements through which the seam is supposed to pass through. To select an element in the GUI: Left-click on the element with the *shift-key* held down. (Figure 5.27)

- To display coordinate nodes of selected elements, right-click with *shift-key* held down and select: **Operations on Selection → Selected Elements → Display Connected Nodes**. (Figure 5.28)

- Choose the nodes in the path of the seam to be created. To select a node: Left-click on the node with the *shift-key* held down. Note: You should only select nodes on the corners of the elements, not mid-side nodes. Once three or more nodes are selected the seam connecting these nodes is drawn if *Coustyx* finds a valid path. Seams are constructed based on the shortest path identified by *Coustyx* while connecting all the selected nodes. It may appear that a seam is not being drawn, if *Coustyx* cannot find a closed path. However, as more nodes are selected in a valid closed path, the seam will be drawn. Only the last selected node can be unselected without disturbing the rest of the node list in the seam. If a node from the seam is arbitrarily unselected, there will be problem in finding the correct closed path. (Figure 5.29)

- Accept the seam by pressing: **Skin → Accept Seam**. See Figure 5.30. **Delete Selected Seams** and **Delete All Seams** could be used to remove seams already constructed. To select a seam for deletion, left-click on the seam while holding down the *shift-key*.

5.5.4. Create Skin

- Unselect all elements by right-clicking on the mesh in the GUI and selecting **Operations on Selection → Unselect All**.

- If necessary, form the first seam as described in Section 5.5.3. If the seam is acceptable, click on **Skin → Accept Seam**. (Figure 5.26)

- Repeat the above process until all the number of necessary seams are created.

- Choose any one face of an element, preferably on the side of the acoustic domain you are interested in, to start skinning. To select an element in the GUI: Left-click on the element with the *shift-key* held down. To ensure skinning is done accurately, unselect all elements in the GUI by right-clicking with the *shift-key* held down and selecting **Operations on Selection → Unselect All**.

- To start the propagation of skin across the mesh select: **Skin → Create Skin**. The skin propagation starts from the selected element and covers the entire mesh defined within seams. The skin that is going to be used to create the BE mesh is shown in color *Red* for inspection. Check if entire mesh you are interested is covered. See Figure 5.31.

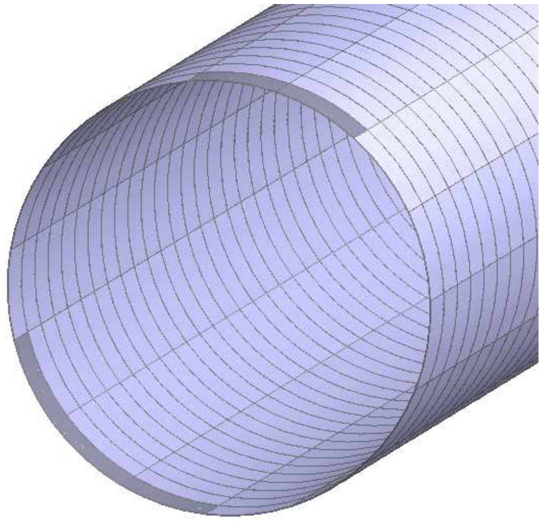


Figure 5.27: Select Elements for Creating a Seam.

- The propagation of the skin could be stopped at any time by pressing: **Skin → Stop Skinning**.
- The created skin can be cleared by pressing: **Skin → Clear Skin**.
- Once you are satisfied with the skin, a new BE mesh can be created using this skin by pressing: **Skin → Create Mesh From Skin** button.

5.6. New Element

This *tabbed* window is located at the bottom pane of the *Mesh Viewer* window. It is used to fill gaps in the mesh by creating new surface (2D) elements. This function is especially useful in creating new elements for cases where the functions Fill Hole and Stitch Seams can not be used. A new element is added by specifying the coordinate node ids and the type of the element to be created. Figure 5.32 shows the window to create a new element in a BE mesh.

- Left-click with *shift-key* held down to select elements.
- Right-click with *shift-key* held down and select: **Operations on Selection: Selected Elements → Display Connected Nodes**.
- Go to the **New Element** tab and select the shape and type of the new element.
- Begin building the new element by selecting nodes. Element **Coord Node ID** show the IDs of the nodes selected in sequence.

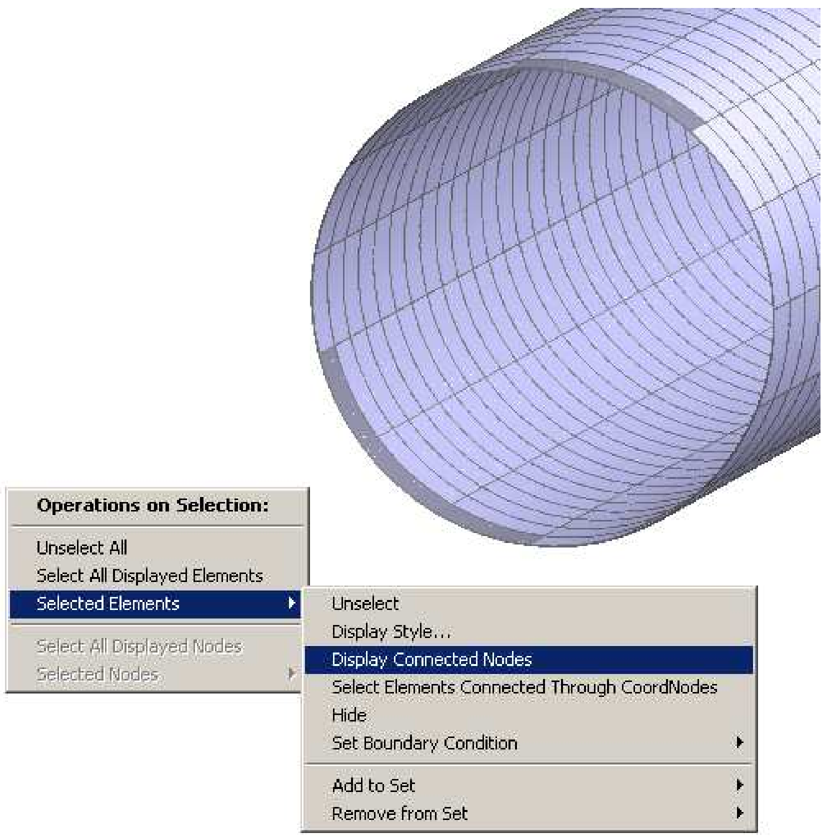


Figure 5.28: Display Connected Nodes for Selected Elements.

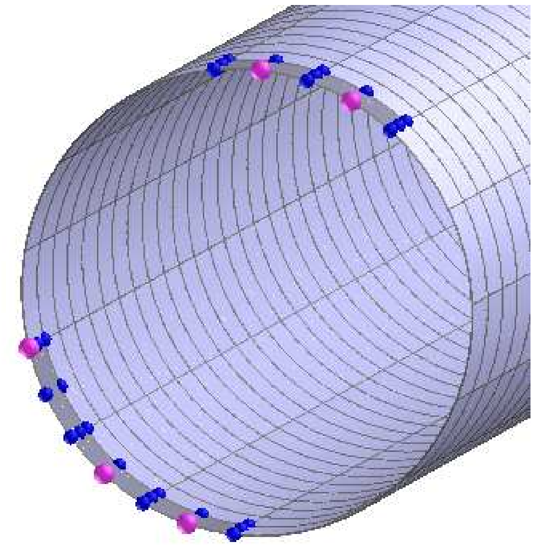


Figure 5.29: Choose Nodes for Creating Seams.

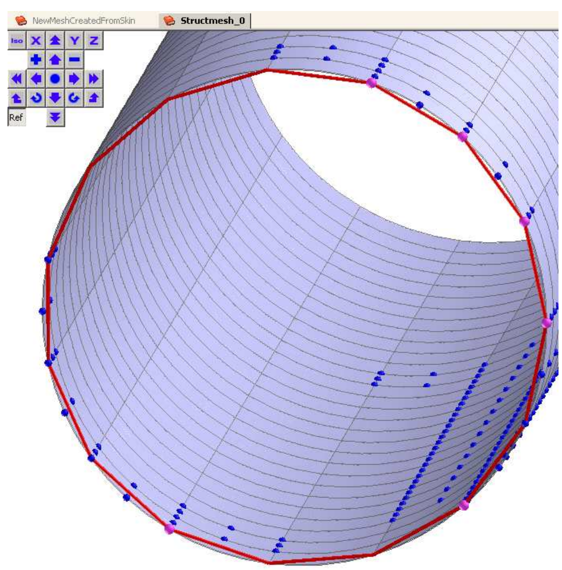


Figure 5.30: Accept Seam.

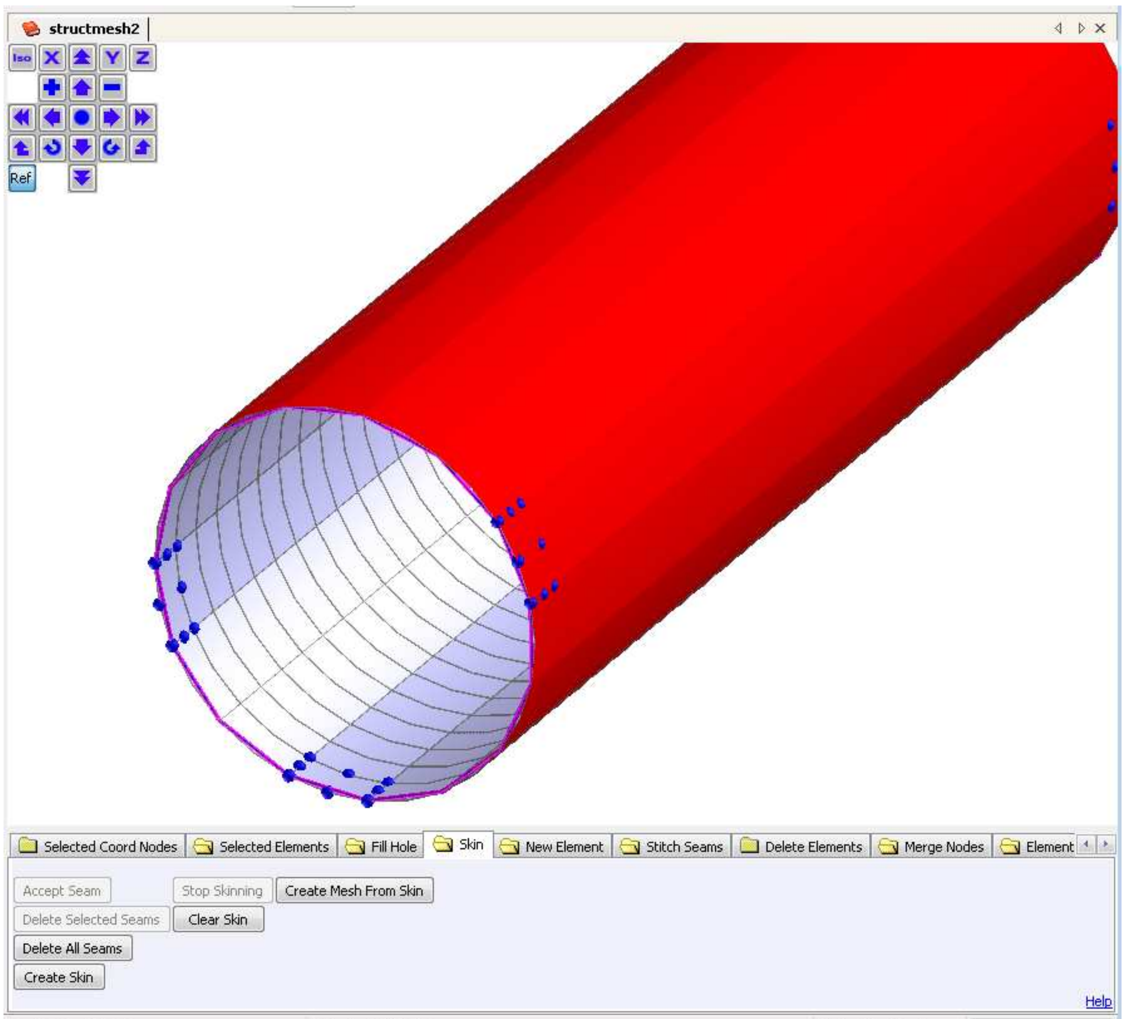


Figure 5.31: Create Skin.

- To select a node in the GUI: left-click on it with the *shift-key* held down.

Coustyx provides options to create new *Triangle* or *Quadrilateral* elements with *linear*, *quadratic*, or *cubic* interpolation schemes. The coordinate nodes for the new element are read by selecting nodes in the GUI. For the formation of a new element, *Coustyx* needs the coordinate nodes to be assigned in the proper sequence. Figure 5.33 shows the coordinate node connectivities for all the element shapes provided by *Coustyx*. The coordinate nodes *should* be chosen in the same order as required by the node numbering shown in the Figure 5.33. The center nodes in higher order elements are optional and are absent for elements created through this function. Once all the coordinate nodes are selected properly, an outline of the new element is shown as a preview in the GUI. If the element looks OK, then create the new element by clicking on the **Accept** button. At any time the selected nodes are deleted by clicking on **Clear Node** which deletes only the selected node, or by clicking on **Clear All Nodes** which deletes all the nodes.

Shape The shape of the new element can be chosen to be either *Triangle* or *Quadrilateral* from the drop down menu.

Order The order of the coordinate interpolation used on the new element. It can be either of the following three options: *Linear* - new element with linear interpolation; *Quadratic* - new element with quadratic interpolation; *Cubic* - new element with cubic interpolation.

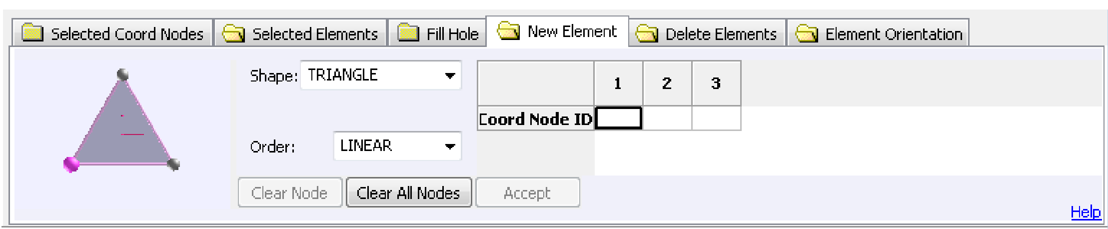


Figure 5.32: Create new element window.

5.7. Stitch Seams

The **Stitch Seams** *tabbed* window is located at the bottom pane the *Mesh Viewer* window. This function could be used to fill gaps between disjointed parts of a structure mesh by generating new triangle elements between the seams defined. Refer to Figure 5.34.

5.7.1. Set Parameters

Seams form closed loops When this option is enabled, *Coustyx* assumes that the seam forms a closed loop and tries to find a closed path by connecting the selected coordinate nodes.

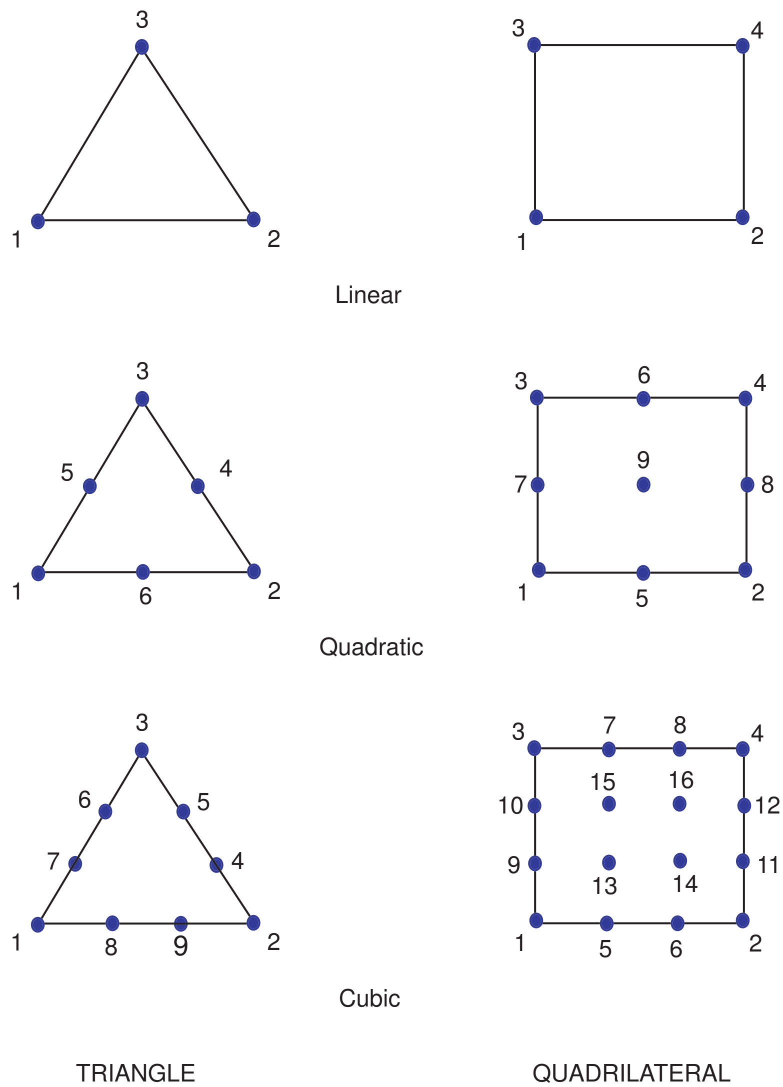


Figure 5.33: New elements coordinate node connectivity.

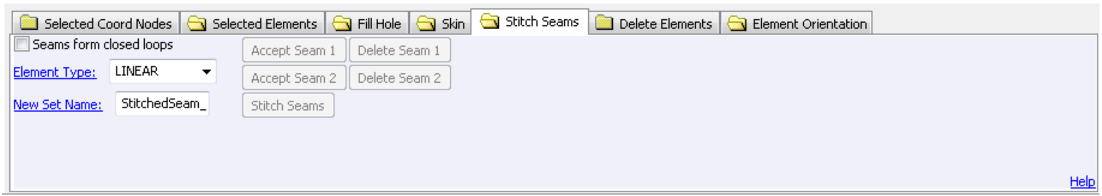


Figure 5.34: Stitch Seam Dialog Panel

If disabled, *Coustyx* assumes that the seam does not form a closed loop and just tries to connect the selected coordinate nodes to form a curve.

Element Type This option specifies the type of triangle elements to be filled in the gap between disjoint seams. *Linear* or *Quadratic* triangle elements can be created.

New Set Name All the triangle elements created to fill the gap can be conveniently added to a Set during the process. This option specifies the name of the new Set.

5.7.2. Procedure to Stitch Seams

Follow the steps below to stitch a seam:

- Open any mesh in the GUI by right-clicking on the mesh and selecting **Open**.
- Enable **Stitch Seams** → **Seams form closed loops** if the seams are expected to form a closed loop. If not disable this.
- Select elements that coincide with the seam edges. To select an element in the GUI: Left-click on the element with the *shift-key* held down. (Refer to Figure 5.35)
- Display connected nodes for the selected elements. Right-click on the selected elements with the *shift-key* held down and select: **Operations on Selection** → **Selected Elements** → **Display Connected Nodes**. See Figure 5.36.
- Choose the nodes in the path of the seam to be created. To select a node: Left-click on the node with the *shift-key* held down. Note: You should only select nodes on the corners of the elements, not mid-side nodes. Once three or more nodes are selected the seam connecting these nodes is drawn if *Coustyx* finds a valid path. Seams are constructed based on the shortest path identified by *Coustyx* while connecting all the selected nodes. See Figure 5.37.
- Accept the first seam by pressing: **Stitch Seams** → **Accept Seam 1**.
- Repeat steps mentioned above to create Seam 2. See Figure 5.38.
- To delete a seam, select: **Stitch Seams** → **Delete Seam 1** or **Delete Seam 2**.

- Once two valid seams are created, *Coustyx* fills up the gap between the seams with transparent 2-D triangle elements for inspection. See Figure 5.39.
- Select the type of triangle elements to be filled in the gap: **Stitch Seams** → **Element Type** → **LINEAR** or **QUADRATIC**.
- Rename the Set: **Stitch Seams** → **New Set Name** → <*Set Name*>.
- To accept the stitch click on: **Stitch Seams** → **Stitch Seams**. See Figure 5.40.

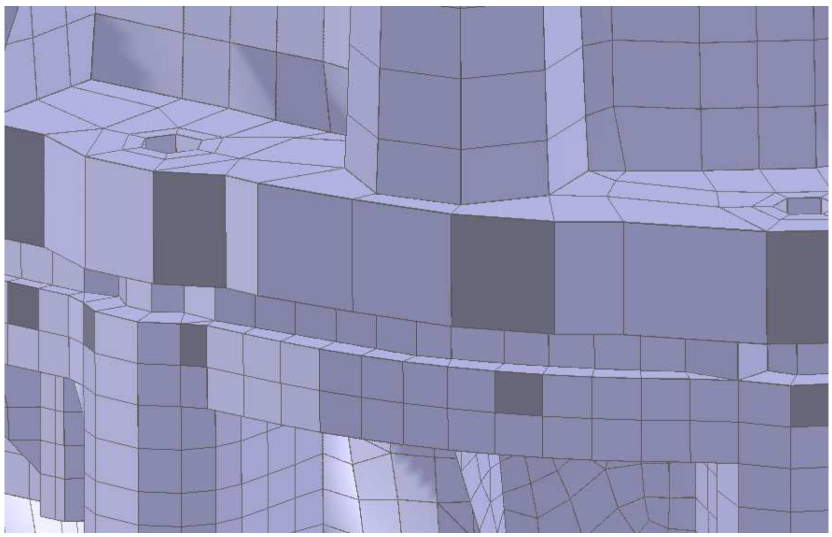


Figure 5.35: Select Elements.

5.8. Delete Elements

This *tabbed* window is located at the bottom pane of the *Mesh Viewer* window. It could be used to delete selected elements in the *Mesh Viewer* window (refer to Figure 5.41). *Coustyx* provides options to remove unshared coordinate nodes and unshared variable nodes of the deleted element.

Delete Unshared Coord Nodes When this option is enabled, the unshared coordinate nodes of selected elements are deleted along with the elements.

Delete Unshared Variable Nodes When this option is enabled, the unshared acoustic variable nodes of selected elements are deleted along with the elements. By default this option is enabled as it is necessary to delete unshared acoustic variable nodes to reduce the number of unknowns.

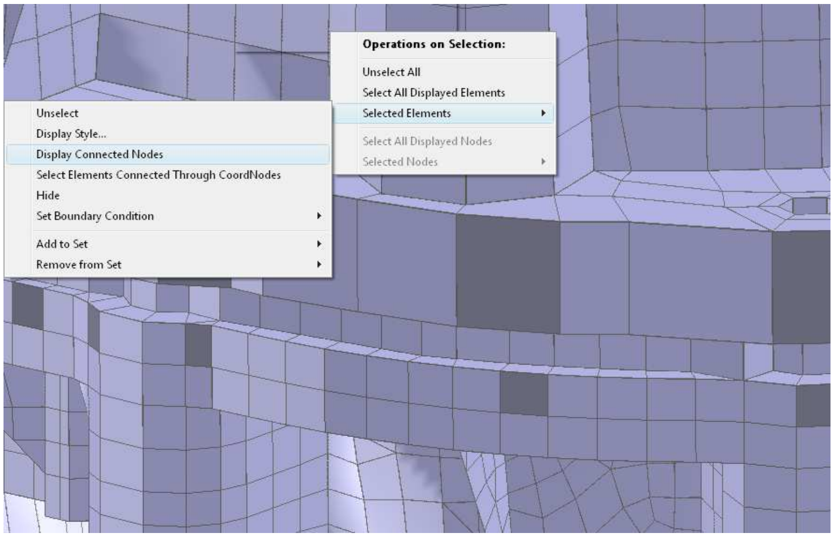


Figure 5.36: Display Connected Nodes for Elements.

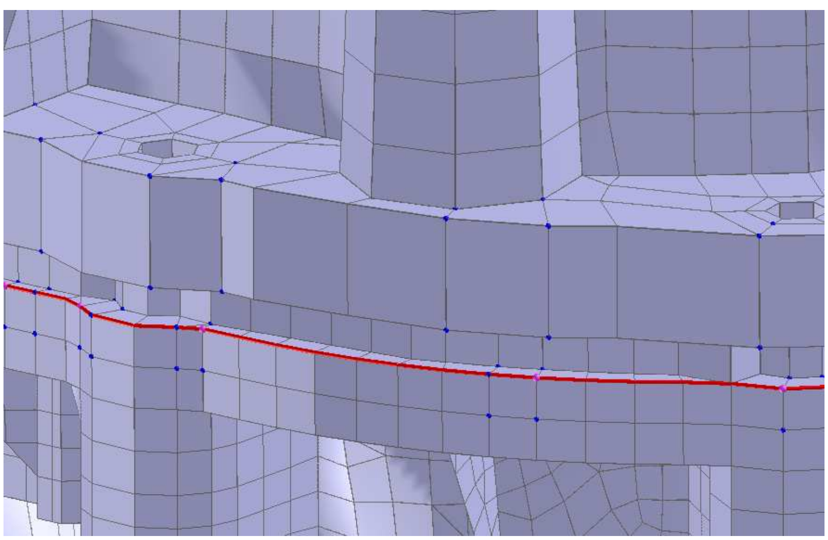


Figure 5.37: Seam 1.

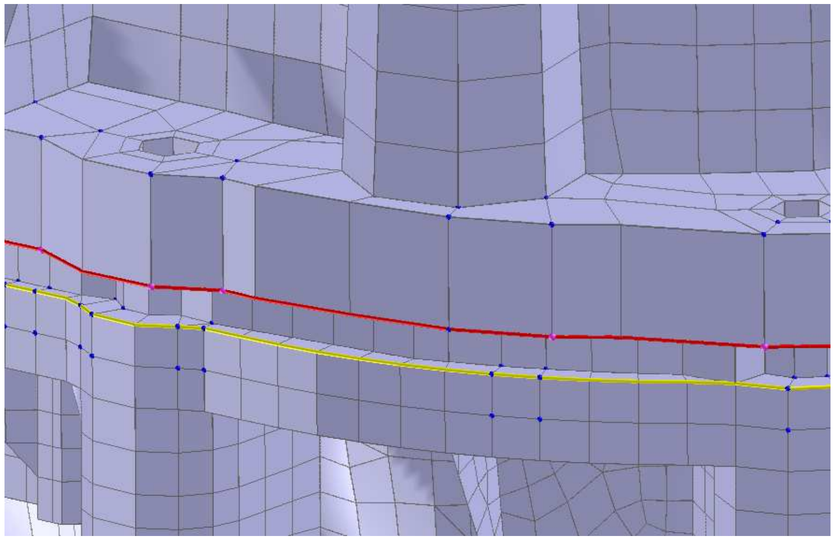


Figure 5.38: Seam 2.

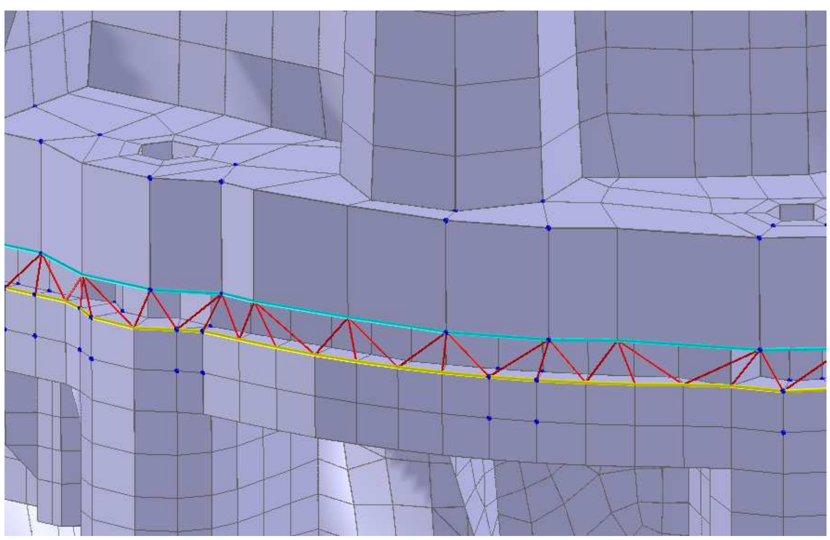


Figure 5.39: Transparent elements shown for inspection before Stitching Seams.

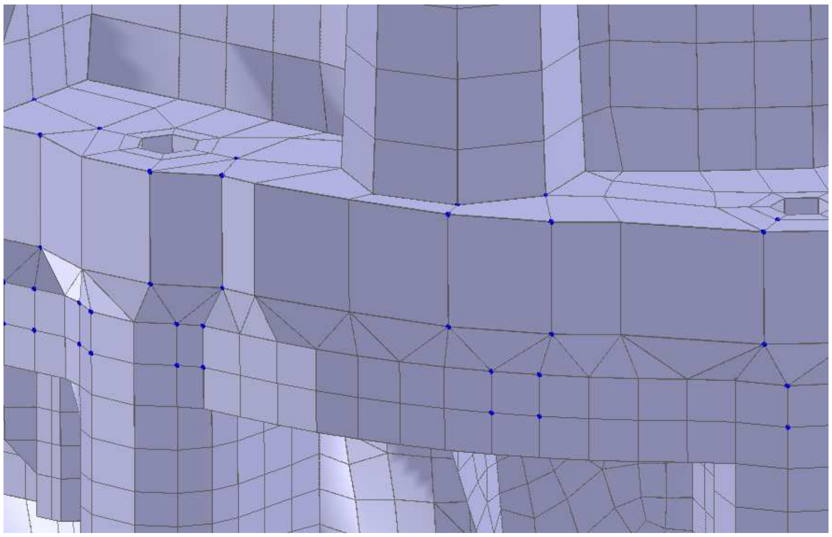


Figure 5.40: Complete Stitched Seam.

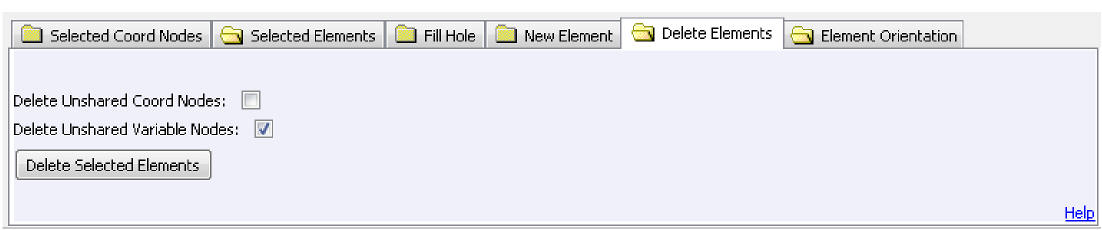


Figure 5.41: Delete element window.

5.9. Merge Nodes

This *tabbed* window is located at the bottom pane of the *Mesh Viewer* window. It is used to merge selected nodes with all coincident nodes in the *Mesh Viewer* window (refer to Figure 5.42). Press the button **Merge Selected Coord Nodes with All Coincident Nodes** to perform this task. Note that this button is active only when nodes are selected in the GUI. *Coustyx* also provides option to enter **Merge Distance**, which is used as the maximum distance within which coincident nodes lie. The option to remove unshared coordinate nodes is also provided. Follow the following instructions to merge coincident nodes.

- Enter the value for merge distance: **Merge Nodes** → **Merge Distance**.
- Select the element containing the desired node by left-clicking on it while holding down the *shift-key*.
- Right-click with *shift-key* held down and select: **Operations on Selection: Selected Elements** → **Display Connected Nodes**.
- Select the desired node by left-clicking on it while holding down the *shift-key*.
- Press **Merge Nodes** → **Merge Selected Coord Nodes with All Coincident Nodes** to merge the selected node with all coincident nodes.

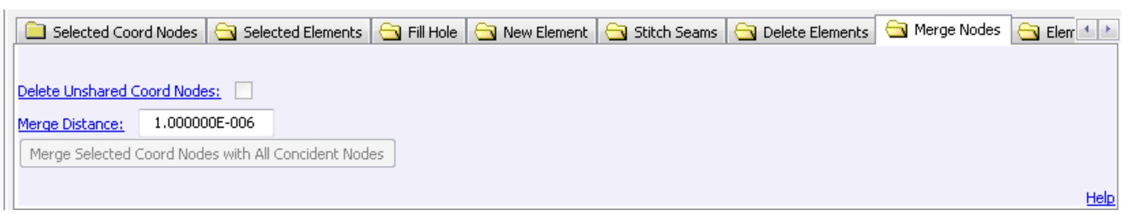


Figure 5.42: Merge Nodes window.

Delete Unshared Coord Nodes When this option is enabled, the unshared coordinate nodes left after the merge are deleted.

Merge Distance A node is considered coincident with the selected node if the distance between the two is less than the *merge distance*. Note that the *merge distance* units are the same as the FE mesh length units.

5.10. Split Pn Nodes / Split Sigma Nodes

This *tabbed* window is located at the bottom pane of the *Mesh Viewer* window. It helps user split **Pn Nodes** or **Sigma Nodes** in *MultiDomain* or *Indirect* models respectively. This feature is primarily useful for creating duplicate nodes along the common edge of adjacent elements with different velocity boundary conditions. To accurately model the *velocity discontinuity* along the common boundary of these adjacent elements, you need to have duplicate **Pn** or **Sigma** nodes. Please be aware that for most other cases, such as geometry discontinuities, edges, corners, junctions etc., *Coustyx* automatically creates appropriate duplicate nodes during skinning. Figure 5.43 shows the window for Split Pn Nodes or Split Sigma Nodes.

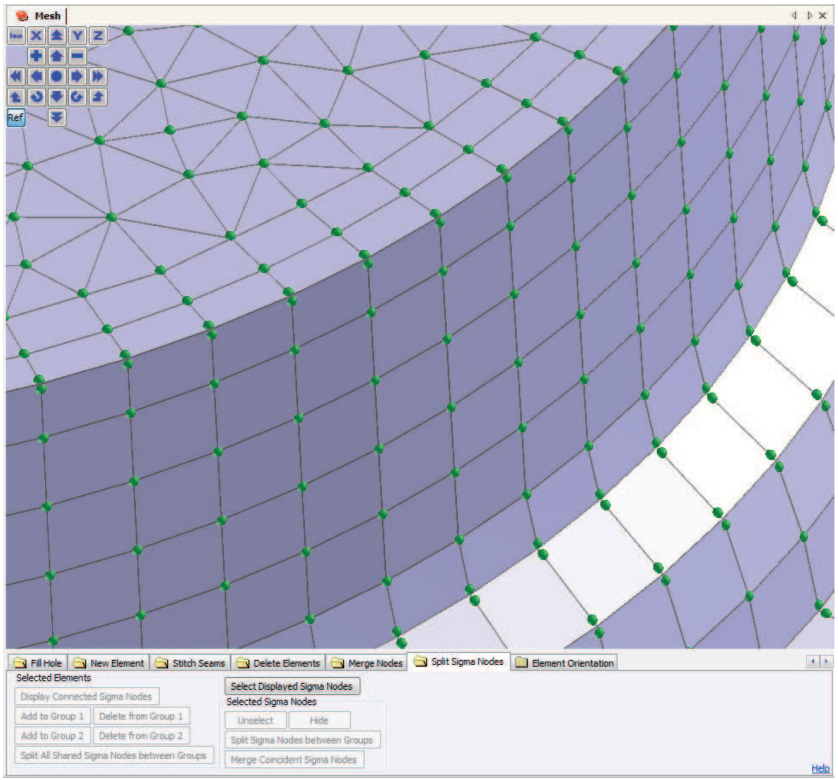


Figure 5.43: Split Pn Nodes (Split Sigma Nodes) window showing duplicate nodes along edges. Note: During skinning, *Coustyx* automatically creates duplicate nodes for most common cases, such as geometry discontinuities, edges, corners, junctions etc.

- Left-click with *shift-key* held down to select elements.
- Click on **Display Connected Pn Nodes** (**Display Connected Sigma Nodes**) to display **Pn Nodes** (**Sigma Nodes**) for all selected elements in a *MultiDomain* (*Indirect*) model .
- Left-click with *shift-key* held down to select a displayed node in the GUI. To select all displayed nodes, click on **Select Displayed Pn Nodes** (**Select Displayed Sigma Nodes**).
- To unselect selected nodes, click on **Unselect** or left-click on a selected node with the *shift-key* held down.

- To hide selected nodes, click on **Hide**.
- To merge coincident nodes among the selected nodes, click on **Merge Coincident Pn Nodes** (**Merge Coincident Sigma Nodes**). When a node is mistakenly split, this function would be helpful to get back to the original state.
- Left-click with *shift-key* held down to select elements and add them to Group 1 (Group 2) by clicking on **Add to Group 1** (**Add to Group 2**). See Figure 5.44 and Figure 5.45. The elements added to Group 1 are displayed in Red and the elements added to Group 2 are displayed in Blue. To delete elements from a Group, select the element and click on **Delete from Group 1** (**Delete from Group 2**). Note: We need to first create two different groups of elements between which common nodes are split.
- After adding elements to Group 1 and Group 2, click on **Split All Shared Pn Nodes between Groups** (**Split All Shared Sigma Nodes between Groups**) to split all shared nodes between Group 1 and Group 2. See Figure 5.46.
- To split only few selected nodes (not all shared nodes) between elements belonging to Group 1 and Group 2, first select nodes by left-clicking on them while holding down the *shift-key*, then click on **Split Pn Nodes between Groups** (**Split Sigma Nodes between Groups**). Only selected nodes are split between Group 1 and Group 2. See Figure 5.47 and Figure 5.48.
- Note: It is important to understand that if a node to be split is shared not only between elements belonging to Group 1 and Group 2, but also between some more elements belonging to neither group, then it is split among Group 1, Group 2 and the rest of elements. For example, let us assume a node is being shared by four elements - A, B, C, and D. See Figure 5.49. If A is added to Group 1, B to Group 2 (C, D are not added to any group), when the shared node is split, *Coustyx* splits the node into 3 different nodes: node 1 for Group 1 (Element A), node 2 for Group 2 (Element B), and node 3 for the rest of elements that belong to neither groups (Elements C and D). See Figure 5.50.

5.11. Element Orientation

This *tabbed* window is located at the bottom pane of the *Mesh Viewer* window. It is used to view and/or flip the normals of the selected elements in the GUI (refer to Figure 5.51). Left-click on any element in the GUI with *shift-key* held down to view its orientation. To view orientations of all the elements in the GUI, right-click with *shift-key* held down and select **Operation on Selection: Select All Displayed Elements**. Note that *green* arrow represents the positive side and the *red* arrow represents the negative side of the element normal.

Flip Selected Elements This is used to flip the normals of selected elements. When an element normal is flipped the coordinate connectivity and variable node connectivity are automatically modified to reflect these changes.

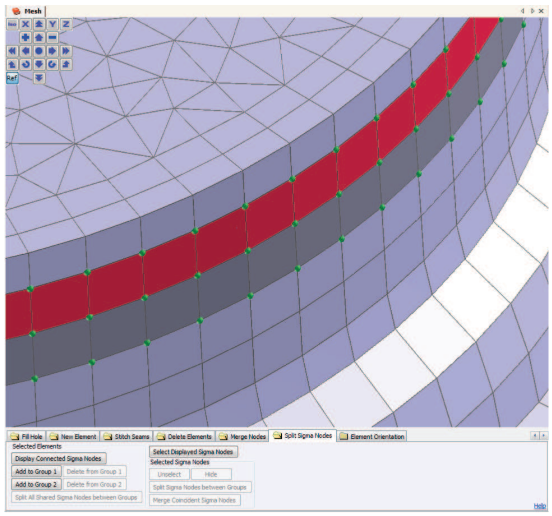


Figure 5.44: After adding elements to Group 1. Notice that the elements belonging to Group 1 are displayed in red.

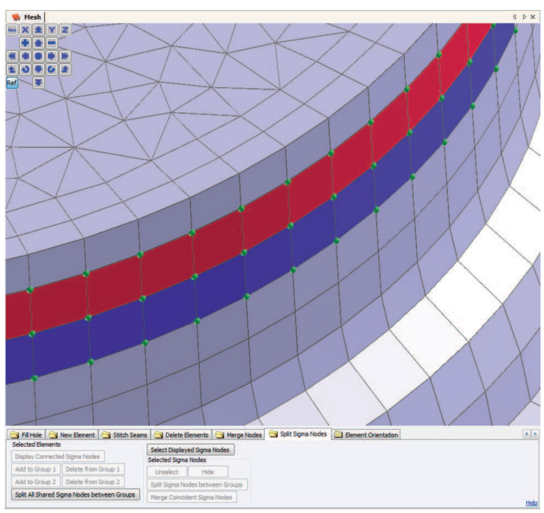


Figure 5.45: After adding elements to Group 2. Notice that the elements belonging to Group 2 are displayed in blue.

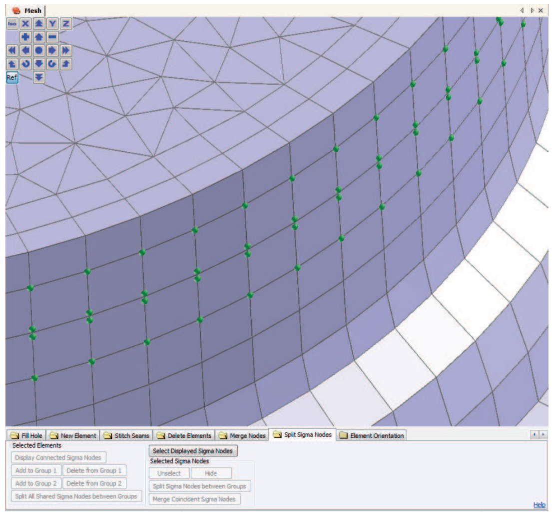


Figure 5.46: After splitting all shared nodes between Group 1 and Group 2. Notice that the nodes along common edges of elements in Group 1 and Group 2 are split into two.

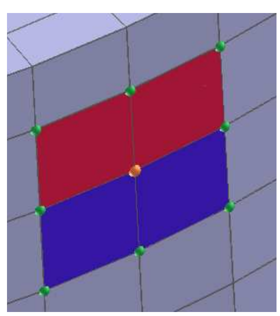


Figure 5.47: Before splitting a selected node between Group 1 and Group 2.

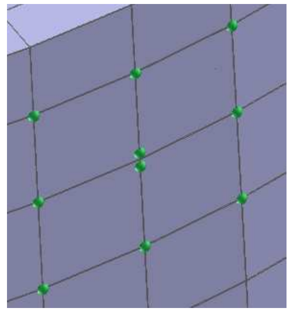


Figure 5.48: After splitting a selected node between Group 1 and Group 2.

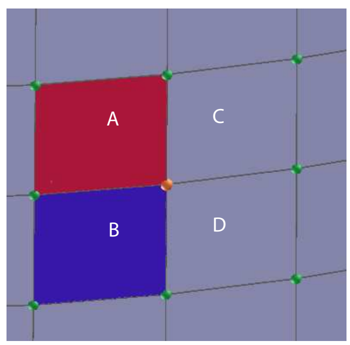


Figure 5.49: Before splitting a node shared by Group 1 (Element A) and Group 2 (Element B). Note that the node is also shared by Elements C and D (that belong to neither Group 1 nor Group 2).

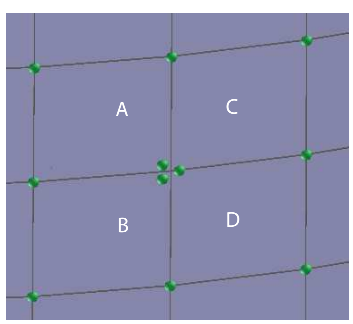


Figure 5.50: After splitting the shared node among Group 1 (Element A), Group 2 (Element B) and unselected elements (Elements C and D).

Make Mesh Consistent with Selected Element This makes all the element normals in the mesh consistent with the normal of the selected element. This option is active only when one element is selected. For *MultiDomain* models, it is important that all elements are consistently oriented.

Make Selected Elements Consistent with Each Other This makes selected element normals consistent with each other. This option is active only when more than one element is selected. Note that this operation is skipped when the group of selected elements have junctions in their midst or when not all the selected elements are connected to one another through coordinate nodes.

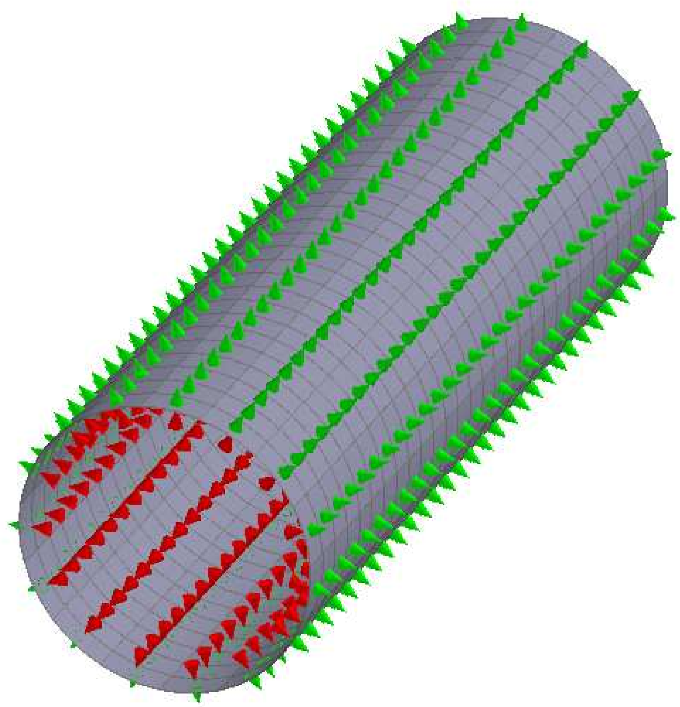


Figure 5.51: Element orientations window.

Bibliography

[1] *Abaqus Analysis User's Manual*. http://www.simulia.com.

Chapter 6

Boundary Conditions

Boundary conditions (BCs) are used in *Coustyx* to model reflection, absorption, or excitation of sound waves from the boundary surfaces.

Coustyx offers wide variety of boundary conditions for both *MultiDomain* and *Indirect* BE models. The user defines the boundary conditions as separate entities which are named uniquely. These boundary conditions can later be applied to the individual elements directly or collectively through sets (group of elements), before running the analysis.

6.1. Impedance Definition in Coustyx

Coustyx supports impedance boundary conditions to simulate the presence of sound absorbing materials.

Sound pressure (p) at the surface of the acoustic material (Figure 6.1) is related to particle normal velocity (v_{ni}) as

$$\frac{p}{v_{ni} - v_{n\_wall}} = Z \tag{6.1}$$

where $Z = R+jX$ is the impedance of the acoustic material, R and X are the real and imaginary parts of Z, and v_{n_wall} is the wall (or structure) velocity. The particle normal velocity v_{ni} at any point on the sample surface is the *inward* normal component of the fluid velocity ($\vec{v}$) at that point, that is, $v_{ni} = \mathbf{v}.\mathbf{n_i}$, where $\mathbf{n_i}$ is the inward normal direction pointing into the acoustic material and out of the fluid. Figure 6.1 shows the impedance definition followed in *Coustyx* for a rigid wall case ($v_{n\_wall} = 0$). This definition is consistent with the definition described in Pierce [1].

One of the popular methods used to measure impedance (Z) of a material is the *impedance-tube* method. The values of Z are deduced from the standing wave pattern developed when excited by a plane wave in a cylindrical tube (*impedance-tube*) with acoustic material sample at one end Pierce [1].

The time dependence of oscillating quantities in *Coustyx* follow $e^{-j\omega t}$ convention, where ω is frequency of fluctuation. If the user adopts $e^{j\omega t}$ convention during measurements then the impedance value should be appropriately modified before using in *Coustyx*. For example,

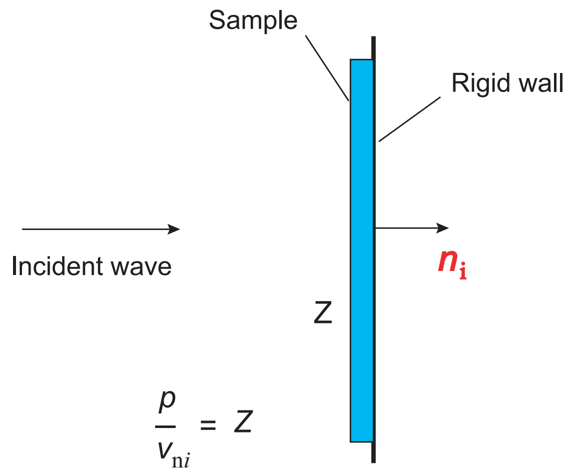


Figure 6.1: Definition of impedance.Note p is the pressure and v_{ni} is the particle normal velocity in $\mathbf{n_i}$ direction at the surface of the material.

impedance value, $Z = R + jX$, adopted from the experimental data with $e^{j\omega t}$ convention should be modified to $Z = R - jX$ to account for the difference in the convention.

Figure 6.2 shows the normalized impedance values measured for a foam of 1-inch thickness. The values are measured using the $e^{-j\omega t}$ convention. Observe that the imaginary part of impedance X is positive at sufficiently low frequencies, and the real part of impedance R is positive at all frequencies. Most acoustic materials and backing combinations are stiffness controlled at low frequency [1]. Hence the value of X in Figure 6.2 is large and positive for lower frequencies. Also, the impedance material **absorbs** acoustic energy. For the sound power to be absorbed by the impedance material, the real part of impedance R **should be positive**. This can derived as follows:

From the definition of impedance in Figure 6.1, the average sound power per unit area absorbed by the impedance material can be reduced to

$$Power = \frac{1}{2}\mathrm{Re}(pv_{ni}^*) = \frac{1}{2}\mathrm{Re}([R + iX]\, v_{ni}v_{ni}^*) = \frac{1}{2}R\,|v_{ni}|^2 \tag{6.2}$$

where Re stands for “real part of” and $*$ represents the complex conjugate.

For the sound power to be absorbed the value of R should be positive.

6.2. Interpolation Options for Mismatched Meshes

Interpolation options are used to apply (structure) velocities from a structure mesh to a mismatched boundary element mesh. Note that these options are available only for boundary conditions that use *Structure Velocity*. In this section, we will learn the kind of velocity interpolation performed for mismatched meshes and the options to control it.

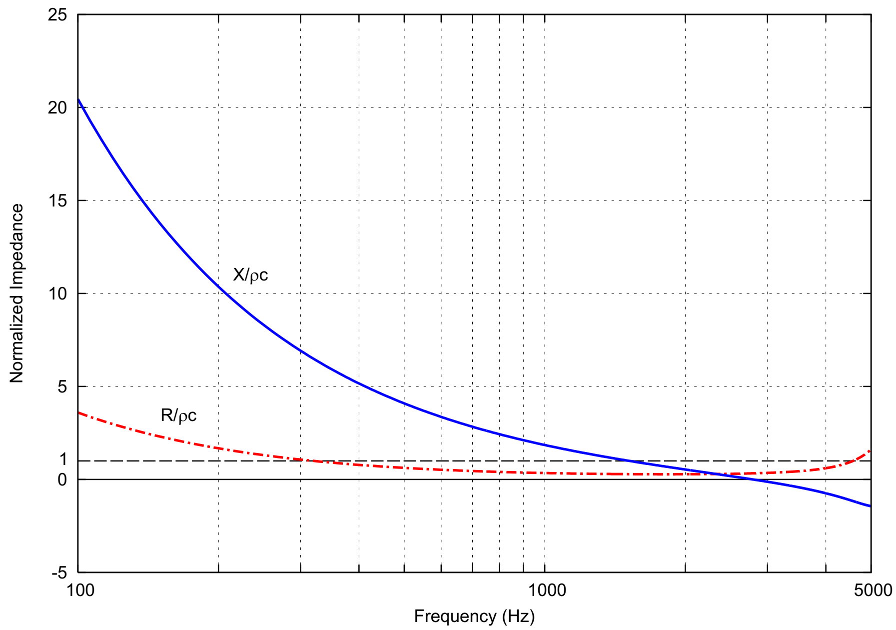


Figure 6.2: Specific acoustic impedance $Z/\rho_o c = R/\rho_o c + jX/\rho_o c$ for a foam of 1-inch thickness measured using $e^{-j\omega t}$ convention.

A boundary element (BE) mesh and a structure mesh are said to be *mismatched* when not all nodes in the BE mesh have a corresponding node in the structure mesh that is coincident (by position). In such cases, nodal velocities at mismatched BE nodes are estimated by interpolating velocities from the nearby structure nodes (also referred to as *interpolating points*). Refer to Figure 6.3. For two matching meshes the velocities at structure nodes are directly applied to BE nodes.

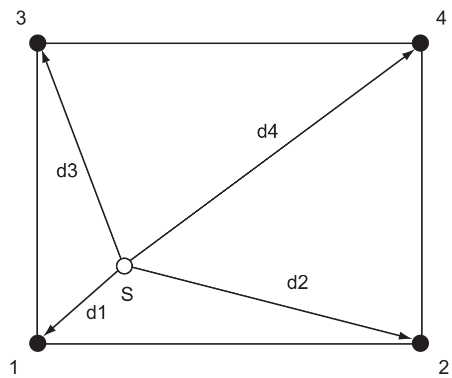


Figure 6.3: Interpolation for mismatched meshes.

Coustyx employs an inverse distance weighted interpolation method to estimate velocities at the mismatched nodes. The inverse of the distance between the BE node and the interpolating point is used as the weight. This approach diminishes the effect of far away interpolating points with respect to the nearby points. Employing this approach, the unknown velocity at a BE node is estimated by computing the weighted average of known velocities at 'N' nearby interpolating points collected from the structure mesh. Thus, the velocity at a mismatched BE node ($v(S)$) computed from the four interpolating points (shown in Figure 6.3) is given by

$$v(S) = \frac{\frac{1}{d_1}v_1 + \frac{1}{d_2}v_2 + \frac{1}{d_3}v_3 + \frac{1}{d_3}v_3}{\frac{1}{d_1} + \frac{1}{d_2} + \frac{1}{d_3} + \frac{1}{d_4}} \tag{6.3}$$

Various options that control the number of interpolation points and the search criteria are discussed below. Figure 6.13.

Choose Default Options Enable this to use default options. In the default options the number of interpolation points are automatically set to four. Coustyx searches for the four nearest interpolating points (or structure nodes) for a given BE node. Then, the weighted average of the velocities at these points is computed and applied as the nodal velocity to the BE node (Equation 6.3). Weights are obtained from the inverse of the distance between the BE node and the interpolating points.

The search for interpolating points is terminated only when all the four nearest structure nodes are found or the entire structure mesh is searched, which ever happens first. If no interpolation points are found during the search, a zero nodal velocity is assigned to the BE node.

If the user wants to terminate the search sooner (than the search over the entire structure mesh), or change the number of interpolating points (from four), then disable this option and set the user options manually.

Number of Interpolating Points This option sets the number of points (N) to be used for the velocity interpolation.

The search for interpolating points is terminated when any of the following criteria is met: (a) 'N' nearest structure nodes are found, (b) maximum search distance from the BE node position is reached, (c) (optional) when the relative weight (in percentage) at the farthest point is less than a user defined tolerance. If no interpolation points are found during the search, a zero nodal velocity is assigned to the BE node.

Maximum Search Distance This option sets the maximum distance away from a BE node within which the search for 'N' nearest interpolating points is performed.

The search for interpolating points is terminated when any of the following criteria is met: (a) 'N' nearest structure nodes are found, (b) maximum search distance from the BE node position is reached, (c) (optional) when the relative weight (in percentage) at the farthest point is less than a user defined tolerance. If no interpolation points are found during the search, a zero nodal velocity is assigned to the BE node.

Terminate if Percentage Weight of Farthest Point is less than This sets the option to terminate the search when the percentage weight of the farthest point is less than a user defined tolerance (tol). That is, the search is terminated if the following is true:

$$\frac{\frac{1}{d_k}}{\sum \frac{1}{d_i}} * 100 < tol$$

where d_k is the distance of the farthest interpolating point from the BE node, and $\sum \frac{1}{d_i}$ is the sum of the weights of all interpolating points.

This option is specifically useful if the user wants to discard interpolating points that contribute less than a specified tolerance value compared to others, even when there are less than 'N' interpolation points or the point lies with in the maximum search distance.

The search for interpolating points is terminated when any of the following criteria is met: (a) 'N' nearest structure nodes are found, (b) maximum search distance from the BE node position is reached, (c) when the percentage weight at the farthest point is less than a user defined tolerance. If no interpolation points are found during the search, a zero nodal velocity is assigned to the BE node.

6.3. Multi-Domain Model BCs

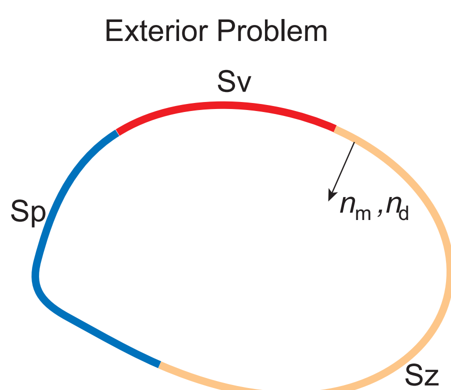
The primary variables in *MultiDomain* models are *pressure* (p) and *normal derivative of pressure* (p_n). And the boundary conditions are applied in terms of pressure, velocity, and impedances on the BE surfaces.

Before we discuss how to define different boundary conditions, the user should understand the distinction between *Mesh Normal and* Domain Normal in a *MultiDomain* model (Figure 6.4), in order to assign the boundary velocities correctly.

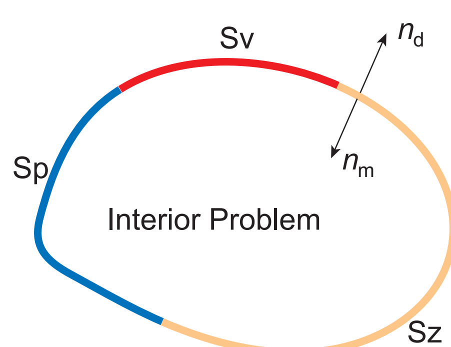
- **Domain Normal** is defined to be always pointing away from the domain of interest. Figure 6.4 shows opposite domain normals for interior and exterior problems but having the same mesh normal. For the exterior problem, the domain of interest is the exterior region and hence the domain normal is pointing inward; while the domain normal for the interior problem is pointing outward.

  Coustyx MultiDomain BE formulation is derived based on the assumption that the domain normal always points away from the domain of interest. All the derivatives in the formulation are with respect to the domain normal. Hence the **boundary conditions in a *MultiDomain* model must be defined with respect to the domain normal**.

- **Mesh Normal** at a specified point is the surface normal at that point based on the element orientation. The element orientation depends on the element coordinate connectivity. In Figure 6.4, the element normal is arbitrarily assumed to be pointing inward. The direction of mesh normal is irrelevant while defining boundary condition.



(a) Exterior problem



(b) Interior problem

Figure 6.4: Definition of domain (n_d) and mesh normals (n_m). Note that domain normal always points away from the domain of interest. Mesh normal, however, can point towards or away from the fluid. All boundary conditions in a *MultiDomain* model are defined with respect to the domain normal.

For a *MultiDomain* model, the Boundary Conditions are defined at: **Model → Boundary Conditions**.

To create a new Boundary Condition right-click on **Boundary Conditions** and select **New** (Figure 6.5). Below is the description of each of the boundary condition options provided in *MultiDomain* model.

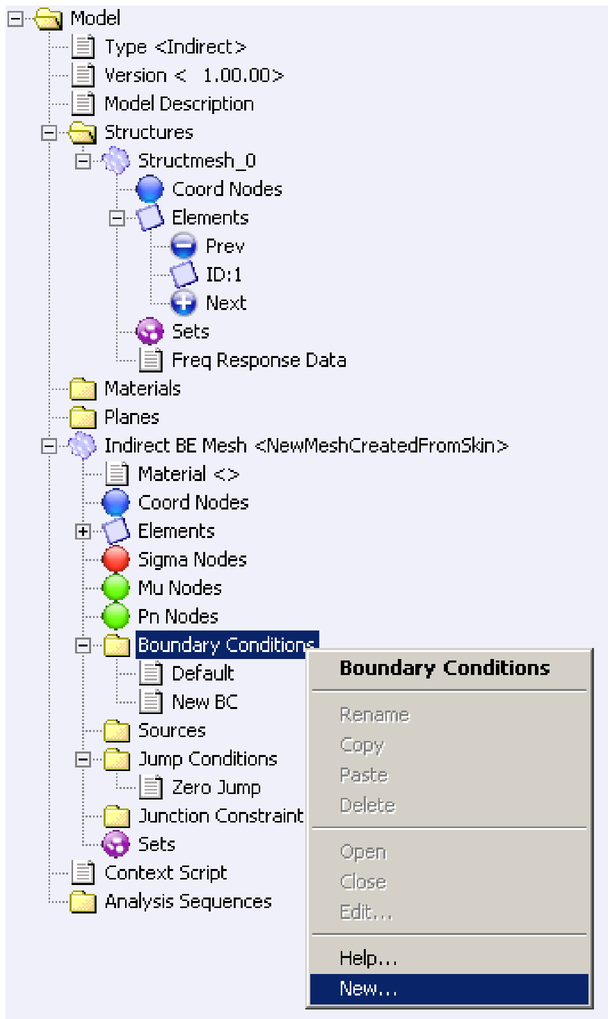


Figure 6.5: New Boundary Condition.

6.3.1. Dummy BC

This Boundary Condition is applied on elements which have no interaction with the fluid medium. They don't contribute to the acoustic radiation. (Figure 6.6.) For example, *Dummy BC* could be used on elements lying on a ground plane, as these elements don't interact with the fluid medium.

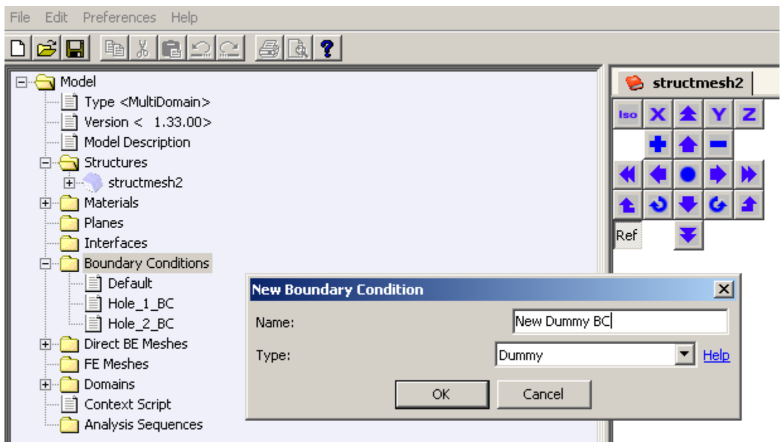


Figure 6.6: Dummy Boundary Condition.

6.3.2. Interface BC

This Boundary Condition is applied on the boundary of an interface between two domains in a *MultiDomain* model. Physical details of the interface are provided in the **Model** → **Interfaces**. (Figure 6.7.)

6.3.2.1. Interface Name

Defines the name of the acoustic interface. (Figure 6.7.)

6.3.2.2. Domain is on the Positive Side of this Interface

Defines whether the acoustic domain being solved is on the positive side of the interface or not. (Figure 6.7.)

6.3.3. Uniform Pressure BC

This Boundary Condition is applied on the element where pressure is uniformly distributed. There is no variation of pressure with position over the element. However, the pressure can

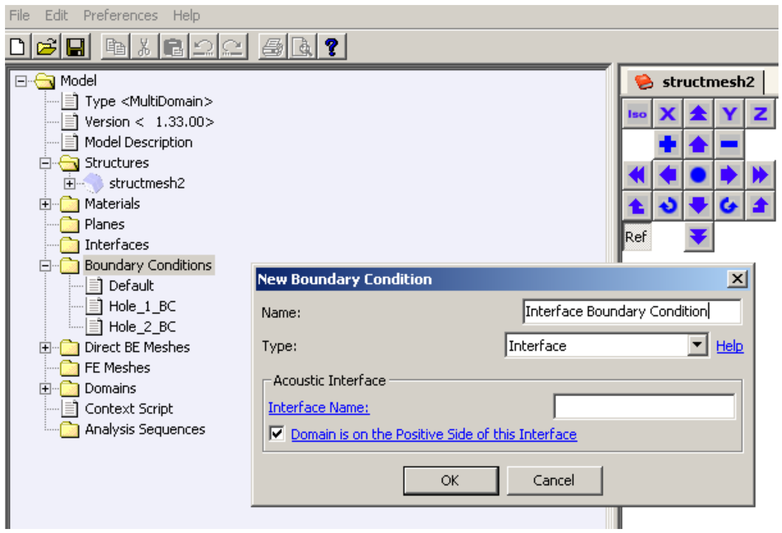


Figure 6.7: Interface Boundary Condition.

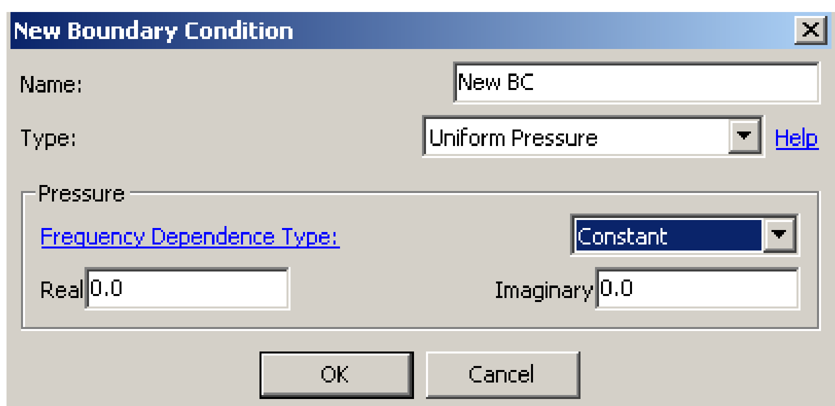


Figure 6.8: Uniform Pressure BC.

be dependent on the frequency. The pressure values can be specified by selecting any of the frequency dependence types: *Constant*, *Table*, or *Script* (Figure 6.8.)

6.3.4. Non-uniform Pressure BC

This Boundary Condition is applied on the element where pressure varies with position. The pressure is defined using a script function *GetPressure*. The input argument to this function is a predefined position variable: *PosnVec*. The variable *PosnVec* reads the coordinates of a point on the element, that is, *PosnVec*=(x,y,z). Other predefined variables that can be used in the script are *AngularFreq* (ω) – frequency in radians/sec, *SoundSpeed* (c) – speed of sound in the medium with the same units as those defined in materials, *WaveNumber* ($k = \frac{\omega}{c}$), and *AmbientDensity* (ρ) – density of the medium with the same units as those defined in materials. (Figure 6.9.)

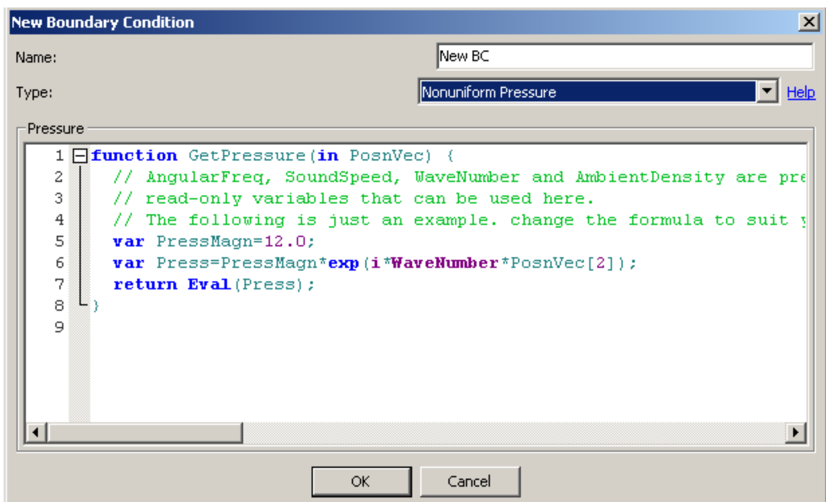


Figure 6.9: Non-Uniform Pressure BC.

6.3.5. Uniform Normal Velocity BC

This Boundary Condition is applied on the element where the normal velocity is uniformly distributed. There is no variation of normal velocity with position over an element. However, it can be defined to be dependent on frequency. (Figure 6.10.) The normal velocity values can be specified by selecting any of the frequency dependence types: *Constant*, *Table*, or *Script*. The user should be aware that the normal velocity defined here is in the direction of the domain normal (Figure 6.4). When *Use Impedance* option is disabled, the *Normal Velocity* defined in

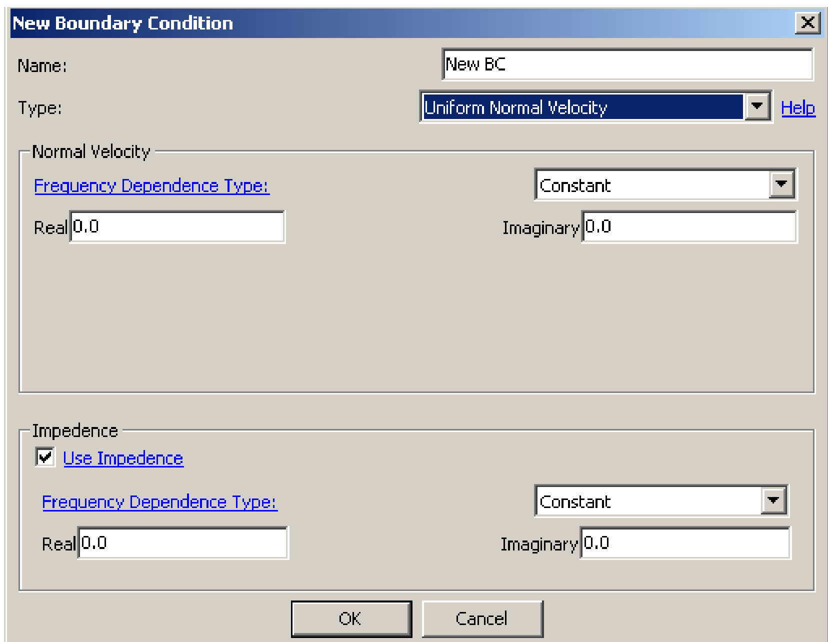


Figure 6.10: Uniform Normal Velocity BC.

this boundary condition (BC) is considered to be the particle normal velocity (v_{ni}). (Figure 6.10)

6.3.5.1. Use Impedance

When this option is enabled, the *Normal Velocity* defined in this Boundary Condition is considered to be the wall velocity (v_{n_wall}). The impedance (Z) relates the pressure (p) at the surface of the acoustic material to the particle normal velocity (v_{ni}) at the surface. Refer Section 6.1 for the definition of impedance implemented in *Coustyx*. The impedance value is defined by selecting any of the frequency dependent types: *Constant*, *Table*, or *Script*.

6.3.6. Uniform Velocity BC

This Boundary Condition is applied on the element where the velocity vector is uniformly distributed. There is no variation of velocity components (v_x, v_y, v_z) with position over an element. However, the velocity can be defined to be dependent on frequency. The components of the velocity at any point can be specified by selecting any of the frequency dependence types: *Constant*, *Table*, or *Script*. (Figure 6.11.)

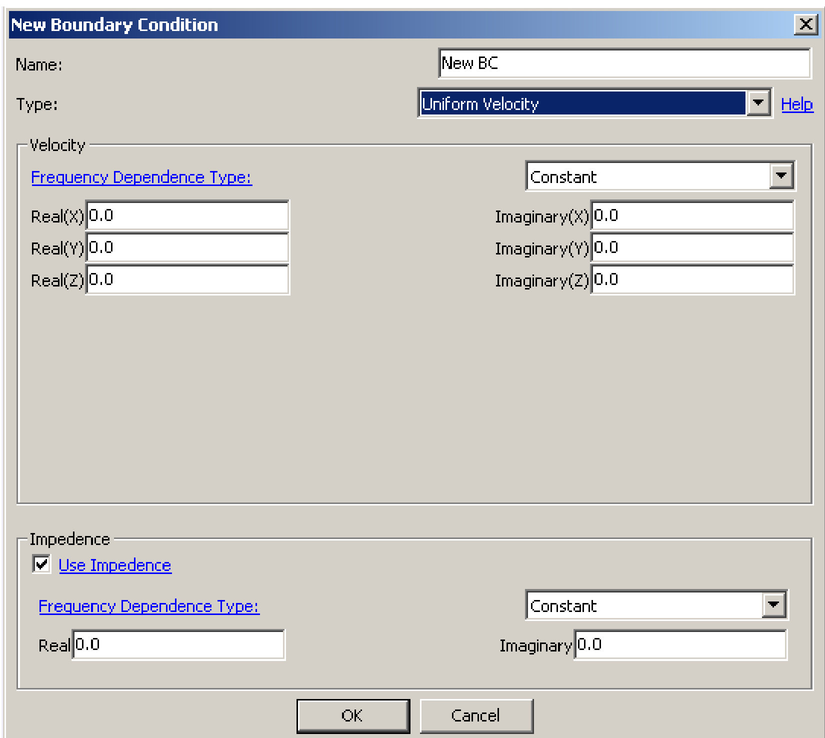


Figure 6.11: Uniform Velocity BC.

6.3.6.1. Use Impedance

When this option is enabled, the *velocity* defined in this Boundary Condition is considered to be the wall velocity. The normal component of the wall velocity (v_{n_wall}) at a point is computed from the dot product of velocity vector to the domain normal at that point (Figure 6.4). The impedance (Z) relates the pressure (p) at the surface of the acoustic material to the particle normal velocity (v_{ni}) at the surface. Refer Section 6.1 for the definition of impedance implemented in *Coustyx*. The impedance (Z) is defined by selecting any of the frequency dependent types: *Constant*, *Table*, or *Script*.

6.3.7. Non-uniform Normal Velocity BC

This Boundary Condition is applied on the element where normal velocity (Note: normal velocity is defined with respect to the domain normal, Figure 6.4) varies with position and normal. The normal velocity is defined in the script by the function *GetNormalVelocity*, which takes in the predefined variables *PosnVec* and *NormalVec* as the arguments. The variable *PosnVec* reads the coordinates of a point on the element, that is *PosnVec*=(x,y,z). The variable *NormalVec* reads the components of the domain normal vector at a point on the element, that is *NormalVec*=(n_x,n_y,n_z). The special vectors **e1**, **e2**, **e3** are predefined unit vectors in coordinate directions of the reference frame which can be used in the *Coustyx* scripts. When *Use Impedance* option is disabled, the *Normal Velocity* defined in this boundary condition (BC) is considered to be the particle normal velocity (v_{ni}). (Figure 6.12.)

Other predefined variables that can be used in the script are *AngularFreq* (ω) – frequency in radians/sec, *SoundSpeed* (c) – speed of sound in the medium with the same units as those defined in materials, *WaveNumber* ($k = \frac{\omega}{c}$), and *AmbientDensity* (ρ) – density of the medium with the same units as those defined in materials.

6.3.7.1. Use Impedance

When this option is enabled, the *Normal Velocity* defined in this Boundary Condition is considered to be the wall velocity (v_{n_wall}). The impedance (Z) relates the pressure (p) at the surface of the acoustic material to the particle normal velocity (v_{ni}) at the surface. Refer Section 6.1 for the definition of impedance implemented in *Coustyx*. The impedance is assigned through a script which uses the function call *GetImpedence*. The function takes in the predefined variables *PosnVec* and *NormalVec* as the input arguments. Other predefined variables such as *AngularFreq*, *SoundSpeed* etc., can also be used to compute impedance. (Figure 6.33)

6.3.8. Structure Velocity BC

Structure velocity Boundary Condition is applied when the frequency response data is known from the FEA analysis of the structure. Refer to Section 5.1.2 for details on how to load frequency response data. (Figure 6.13.)

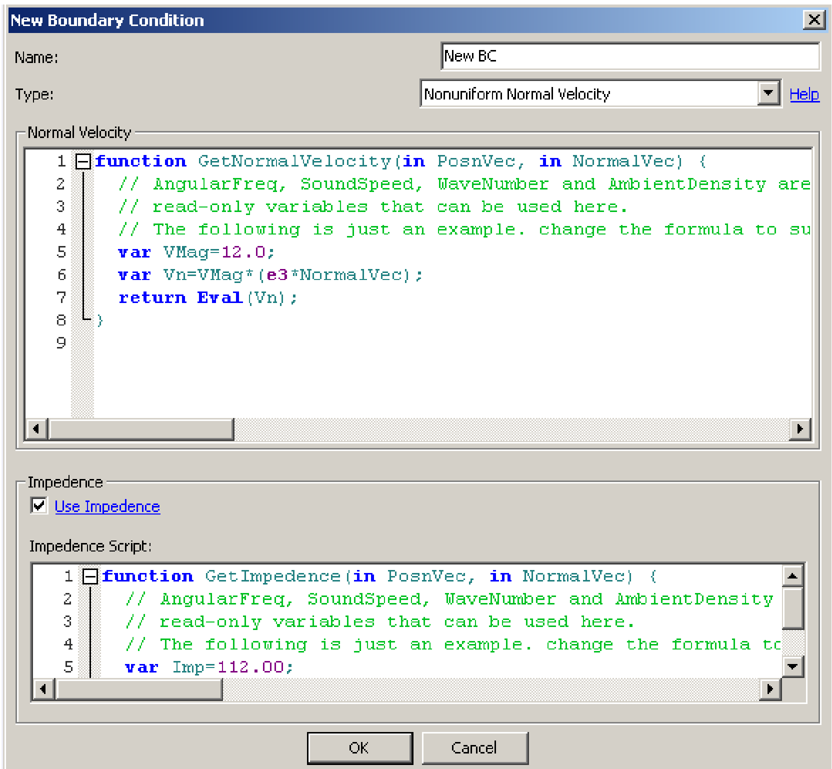


Figure 6.12: Non-Uniform Normal Velocity BC.

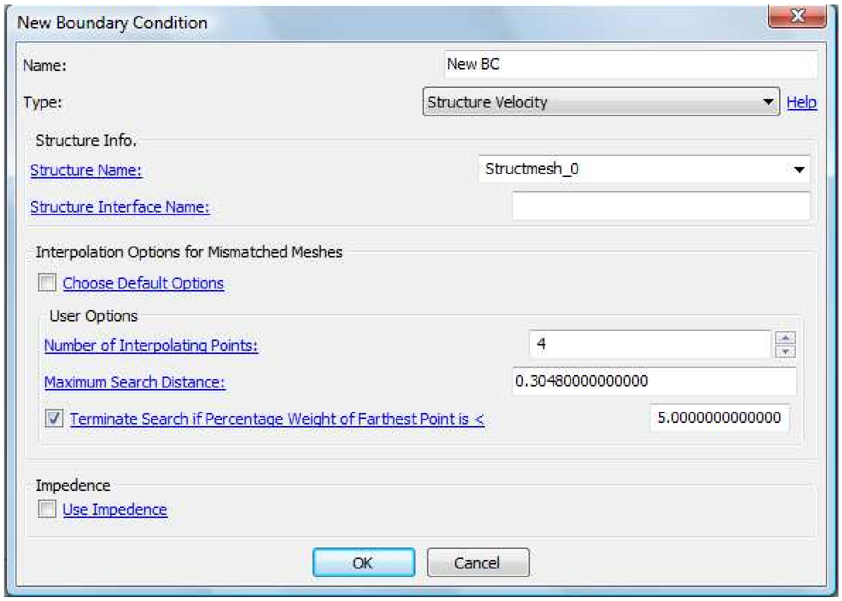


Figure 6.13: Structure Velocity BC.

6.3.8.1. Structure Name

Select the structure name from which the Boundary Conditions are applied. (Figure 6.13.)

6.3.8.2. Structure Interface Name

Select the structure interface from which the Boundary Conditions are applied. A structure interface is a group of faces (of FEA elements) in contact with the fluid medium. Each structure interface is uniquely identified by the structural mesh id, element id, and the local face number on the element. If this field is left blank, the structure velocity is applied directly from the structure. (Figure 6.13.)

6.3.8.3. Interpolation Options for Mismatched Meshes

Refer to Section 6.2.

6.3.8.4. Use Impedance

When this option is enabled, the impedance (Z) relates the pressure (p) at the surface of the acoustic material to the particle normal velocity (v_{ni}) and the structure velocity (v_{n_wall}). Refer Section 6.1 for the definition of impedance implemented in *Coustyx*. The impedance

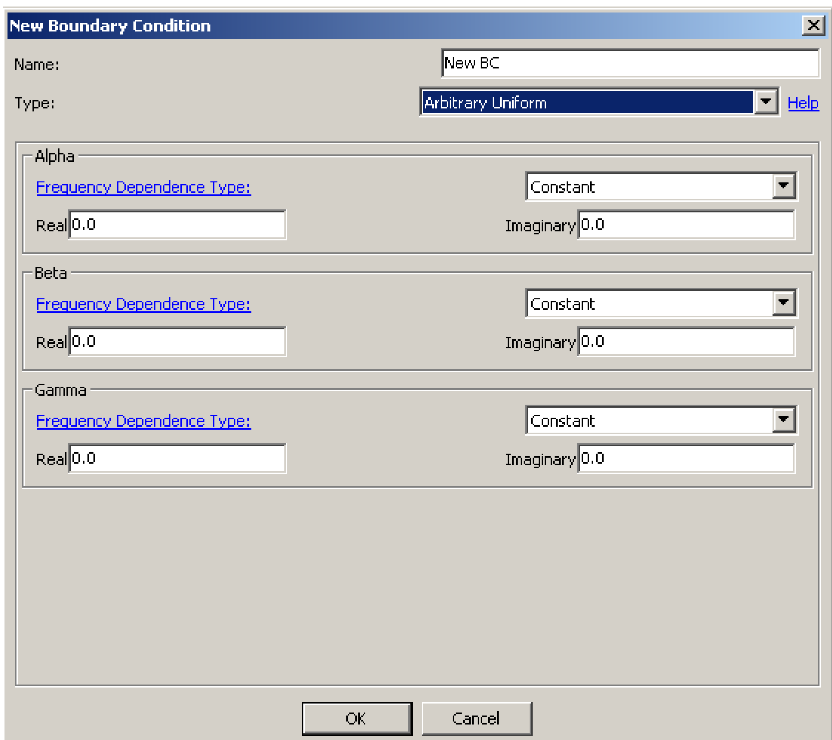


Figure 6.14: Arbitrary Uniform BC

(Z) is defined by selecting any of the frequency dependent types: *Constant*, *Table*, or *Script*. (Figure 6.13.)

6.3.9. Arbitrary Uniform BC

This option can be used to define a Uniform Boundary Condition (BC) that doesn't fall into any of the other BC types. Any uniform boundary condition could be expressed in this general form. The general equation for this BC is given by

$$\alpha p + \beta v_n = \gamma \tag{6.4}$$

where p is the pressure, v_n is the normal velocity at a point on the element, and α, β, γ are variables defined by any of the frequency dependent types: *Constant*, *Table*, or *Script*. The values of α, β and γ don't vary with position over the element. (Figure 6.14.)

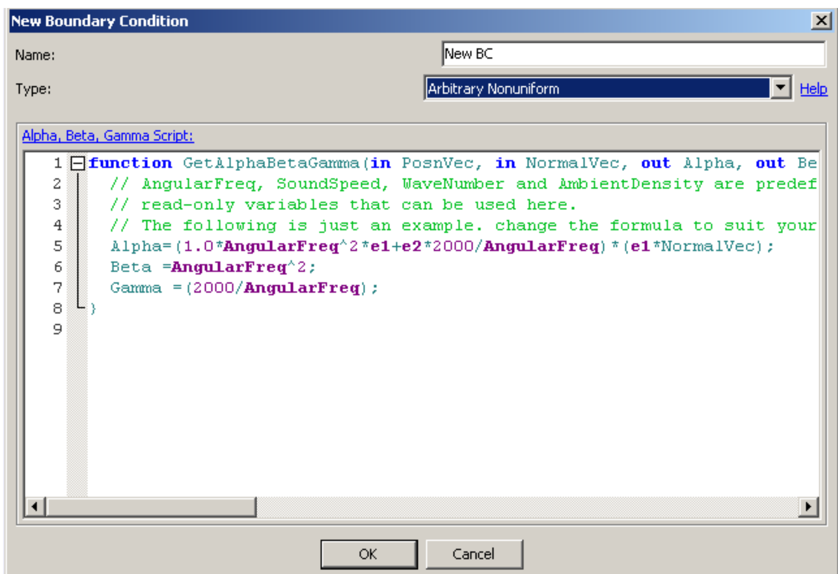


Figure 6.15: Arbitrary Non-Uniform BC

6.3.10. Arbitrary Non-uniform BC

This option can be used to define Non-uniform Boundary Condition that doesn't fall into any of the other BC categories. Any non-uniform boundary condition could be expressed in this general form. The general equation for this BC is given by

$$\alpha(\mathbf{x}, \mathbf{n_x})p + \beta(\mathbf{x}, \mathbf{n_x})v_n = \gamma(\mathbf{x}, \mathbf{n_x}) \tag{6.5}$$

where p is the pressure, v_n is the normal velocity at a point on the element, and α, β, γ are variables that vary with position ($\mathbf{x}$) and normal ($\mathbf{n_x}$) over the element.

The values for α, β, and γ could be computed from the script function *GetAlphaBetaGamma*. The input arguments for this function are: predefined position vector (*PosnVec*), and normal vector (*NormalVec*) at a point. It outputs the values for *Alpha*(α), *Beta*(β), and *Gamma*(γ) which are used in the BC. Other predefined variables that can be used in the script are *AngularFreq* (ω) – frequency in radians/sec, *SoundSpeed* (c) – speed of sound in the medium with the same units as those defined in materials, *WaveNumber* ($k = \frac{\omega}{c}$), and *AmbientDensity* (ρ) – density of the medium with the same units as those defined in materials. (Figure 6.15.)

6.4. Indirect BE Model BCs

The primary variables in *Indirect* BE models are *pressure jumps* or *double layer potential* (μ), and *velocity jumps* or *single layer potential* (σ). The Boundary Conditions, however, are applied in terms of acoustic physical quantities: pressure, velocity, and impedance.

Please note that all the derivatives in *Indirect* BE formulation are with respect to the *Mesh Normal* or *Element Normal*. Hence, all the boundary conditions in *Indirect* BE models are defined with respect to *Element Normals*. This is in contrast with the *MultiDomain* model, where the boundary conditions are defined with respect to the *Domain Normal* (Figure 6.4).

For an *Indirect* model, *Coustyx* allows the user to define different boundary condition types on either side of the mesh. For any element Side 1 is always considered to be on the positive side of the element normal and Side 2 on the negative side. Refer to Figure 6.16 for the element sides definition.

The Boundary Conditions offered in *Indirect* models could be categorized into three broad groups for better understanding:

- **Continuous BCs** – A Continuous BC is one where the Boundary Condition type (known pressure, known velocity, or known impedance) and value on both sides of the element (boundary) are the same. Continuous BCs employ lesser number of variables compared to Discontinuous BCs (mentioned below). Thus, these BCs are very attractive from the problem size and analysis speed points of view. It is imperative that the user apply these BCs wherever possible to take advantage of smaller problem sizes. However, there are certain cases where usage of Discontinuous BCs are very much necessary (refer to Discontinuous BCs below).

- **Discontinuous BCs** – *Coustyx* allows discontinuous boundary conditions with the same boundary condition type (known pressure, known velocity or known impedance) with unequal values specified on both sides of the element (boundary). In addition to this, *Coustyx* also allows discontinuous boundary conditions with *different types* on each side of the element (boundary). For example, pressure can be specified on one side, while specifying velocity on the other side. The capability to completely decouple the two sides of a boundary using *different types* of BCs provide *Coustyx* users with greater modeling flexibility unknown in any of today's commercially available acoustic BE programs.

  The discontinuous boundary conditions could be effectively used to eliminate *Non-uniqueness* in the boundary integral solution at *Irregular frequencies* in meshes which enclose a volume. *Irregular frequencies* are related to the eigen-frequencies of the interior problem. In

Indirect BEM exterior and interior problems are connected and solved at the same time. Hence at *Irregular frequencies*, exterior solutions are contaminated by unbounded interior solutions. The user can employ discontinuous boundary conditions to decouple the exterior and interior problems. As in most cases, if the user is interested in the radiation problem alone, the interior side of the boundary could be assigned a *zero* boundary excitation to suppress the interior excitation. Refer to Ambarisha et al. [2] for more details.

For example, consider a pulsating sphere mesh which encloses a volume and one is interested in the acoustic radiation problem. At the eigen-frequencies of the interior, the unbounded interior solution dominate the value of primary variable μ on the surface. This contaminates the solution for the exterior field. Accurate results for the exterior problem could be obtained by specifying the structure velocity boundary condition on the exterior and a *zero velocity* or *zero pressure* boundary condition on the interior side of the boundary.

- The Boundary Conditions which relate pressures and normal velocities on one side of the boundary to the pressures and velocities on the other side are grouped together in this category. **Anechoic Termination BC**, **Uniform Perforated BC**, **Non-uniform Perforated BC**, **Uniform Arbitrary BC**, **Non-uniform Arbitrary BC** fall under this category.

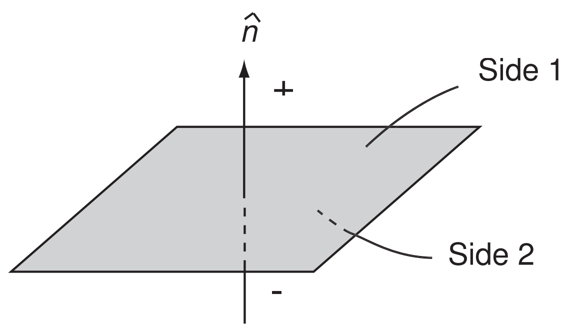


Figure 6.16: Description of element normal and its sides.

For an *Indirect* model, the Boundary Conditions are defined at: **Model** → **Indirect BE Mesh** → **Boundary Conditions**.

To create a new Boundary Condition right-click on **Boundary Conditions** and select **New** (Figure 6.5). Below is the description of each of the boundary condition options provided in *Indirect* model.

6.4.1. Transparent BC

Transparent Boundary Condition is applied on the boundary across which the pressure and normal velocity is continuous. Normally, the transparent BC is not required in the *Indirect* BE models, because this type of continuity can be achieved by simply removing the element from the model. The portion of the surface where continuity conditions are enforced need not be modeled at all because of both the potentials going to zero, that is, $\mu = 0$ and $\sigma = 0$. However, the only exception occurs when the element lies on a *Baffle Plane* where the normal derivative of pressure is the acoustic variable. In this case, the user has to keep the element on the baffle plane and apply Transparent BC. (Figure 6.17)

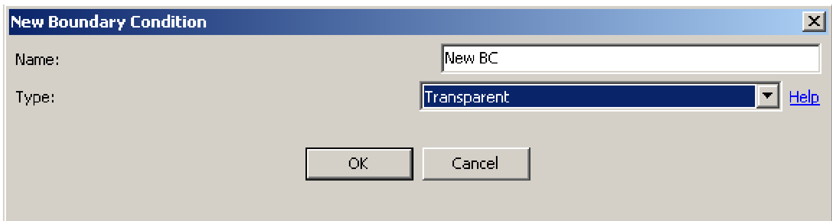


Figure 6.17: Transparent BC

6.4.2. Anechoic Termination BC

This Boundary Condition is applied on the elements where the pressure p (on either side of the boundary) and normal velocity v_n (on either side of the boundary) are related by the characteristic impedance (Z_o) of the fluid medium. Refer Section 6.1 for the definition of impedance implemented in *Coustyx*. The relation between pressure and velocity at anechoic termination is given by

$$\frac{p}{v_n} = Z_o = \rho_o c \tag{6.6}$$

where ρ_o is the mean density of the surrounding medium, and c is the speed of sound in the medium. (Figure 6.18.)

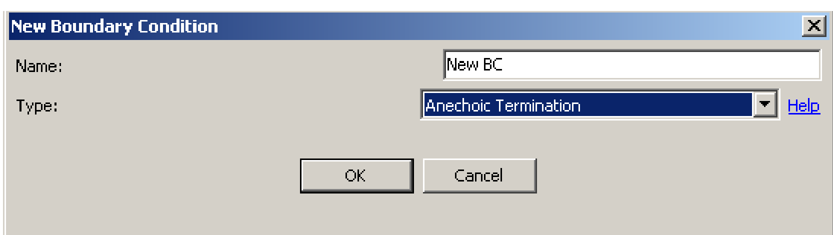


Figure 6.18: Anechoic Termination BC

6.4.3. Perforated BC

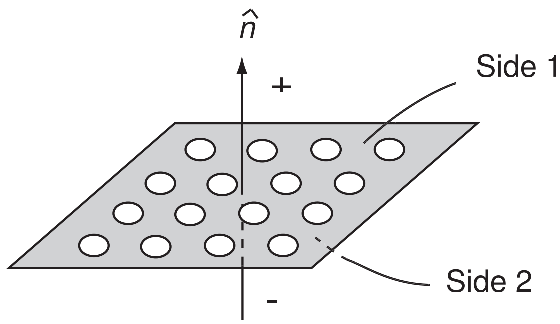


Figure 6.19: Perforated Plate

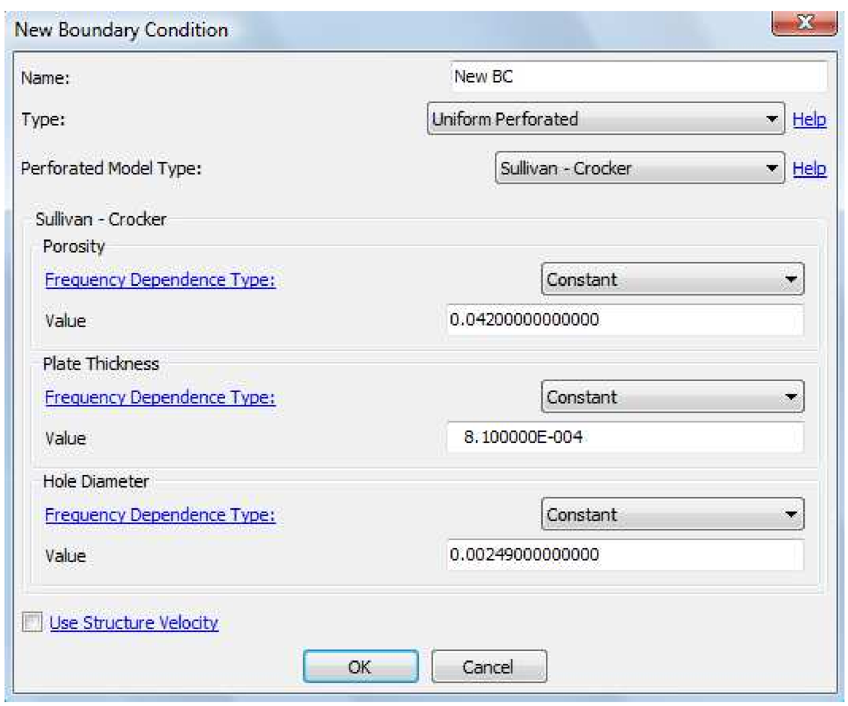


Figure 6.20: Uniform Perforated

This type of Boundary Condition will have applications in perforated mufflers. A perforated BC defines a special type of transfer relation between the pressure, normal velocity on either side of the surface. The transfer relation in a perforated BC is given by

$$p^+ - p^- = -\rho_o c\zeta\left[v_n - \bar{v}_{sn}\right] \tag{6.7}$$

where ζ is the non-dimensional transfer impedance of the perforated surface, ρ_o is the acoustic medium density, c is the speed of sound, p^+ and p^- are surface pressures on side 1 and side 2 respectively, v_n is the acoustic normal velocity, v_{sn} is the specified structure normal velocity.

The user can either select to use the transfer impedance relation derived from the Sullivan and Crocker model ([3]) or they can define their own transfer impedance from the following choices in the drop-down menu **Perforated Model Type**:

6.4.3.1. Sullivan and Crocker

The non-dimensional transfer impedance ζ used in the Sullivan and Crocker model ([3]) is,

$$\zeta = \frac{[0.006 - ik_o(t_w + 0.75d_h)]}{\chi} \tag{6.8}$$

where k_o is the wave number, t_w is the plate thickness, d_h is the hole diameter, and χ is the porosity of the plate (Figure 6.19) (Note: the actual relation from [3] is modified to accomodate $e^{-j\omega t}$ convention used in *Coustyx*). Note that the above relation would be valid only if the porosity is not too different from 0.042 (4.2%) used by Sullivan and Crocker in their experiments.

Porosity Define porosity χ of the plate or pipe using this option. Porosity is defined as the ratio between open surface area and total surface area,

$$\chi = \frac{\text{open surface area}}{\text{total surface area}}$$

When $\chi = 0$ (surface is completely closed), the acoustic normal velocity (v_n) on both sides of the perforated plate will be equal to the structure velocity (v_{sn}). On the other hand when $\chi = \infty$ (open surface), the pressure on both sides of an imaginary dividing surface will be equal. (Figure 6.20.)

Plate Thickness Define the thickness of the plate or pipe (t_w) using this option. Note the units should be consistent with the model length units.

Hole Diameter Define the diameter of holes (d_h) using this option. Note the units should be consistent with the model length units.

6.4.3.2. Transfer Impedance

Coustyx allows you to define your own transfer impedance models. New transfer impedance models are usually derived from experiments or other empirical models. Note that the transfer impedance ζ specified here should have no dimensions. Make sure the impedance definition is consistent with the $e^{-j\omega t}$ convention used in Coustyx (see Section 6.1).

$$\zeta = \tilde{R} + j\tilde{X} \tag{6.9}$$

6.4.3.3. Use Structure Velocity

Structure velocity can be used by enabling this option. Equation 6.7 shows the transfer relation between pressure on both sides of the element (p^+ and p^-) and normal velocity (v_n) when structure velocity (v_{sn}) is present. Refer to Section 6.4.9 for more details on how to define a structure velocity. If this option is disabled structure velocity in the Equation 6.7 is set to zero. (Figure 6.20.)

6.4.3.4. Uniform Perforated BC

This Boundary Condition is applied over perforated elements with position-independent parameters. Refer to Section 6.4.3 for the transfer relation used to model perforated boundary condition. The values for porosity, plate thickness, and hole diameter in **Sullivan and Crocker** model or transfer impedance in **Transfer Impedance** model are set by selecting any of the frequency dependent types: *Constant*, *Table*, or *Script*. (Figure 6.20)

The option **Use Structure Velocity** is enabled to define structure velocity used in Equation 6.7. Refer to Section 6.4.3.3 for further details.

6.4.3.5. Non-uniform Perforated BC

This Boundary Condition is applied over perforated elements with position-dependent parameters. Refer to Section 6.4.3 for the transfer relation used to model perforated boundary condition. The values for porosity, plate thickness, and hole diameter in **Sullivan and Crocker** model is set by a call to the function *GetSullivanCrockerModelParameters*. This function accepts the predefined position vector *PosnVec* as the input argument and outputs *Porosity*, *Plate Thickness*, and *Hole Diameter*. Similarly, the value for the transfer impedance in **Transfer Impedance** model is set by the function call *GetTransferImpedance*, which accepts *PosnVec* as the input and returns transfer impedance. Other predefined variables that can be used in the script are *AngularFreq* (ω) – frequency in radians/sec, *SoundSpeed* (c) – speed of sound in the medium with the same units as those defined in materials, *WaveNumber* ($k = \frac{\omega}{c}$), and *AmbientDensity* (ρ) – density of the medium with the same units as those defined in materials. (Figure 6.21.)

The option **Use Structure Velocity** is enabled to define structure velocity used in Equation 6.7. Refer to Section 6.4.3.3 on how to use this option.

6.4.4. Uniform Pressure (Continuous) BC

This Boundary Condition is applied on elements with uniform pressure on both sides of the boundary. The BC is continuous, which implies that the values of pressure at the same point on side 1 (p^+) and side 2 (p^-) are identical (refer to Figure 6.16 for side 1 and side 2 definitions).

$$p^+ = p^- = p_o$$

where p_o is the uniform pressure applied on both sides. (Figure 6.22.)

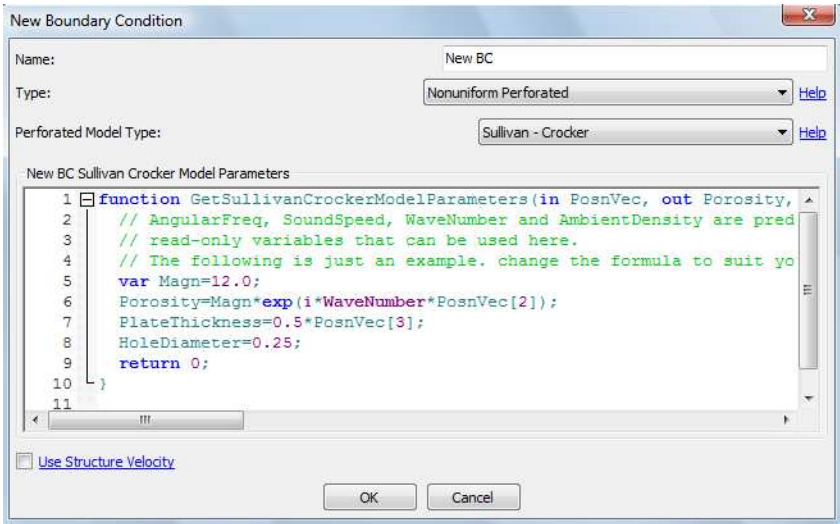


Figure 6.21: Non-Uniform Perforated

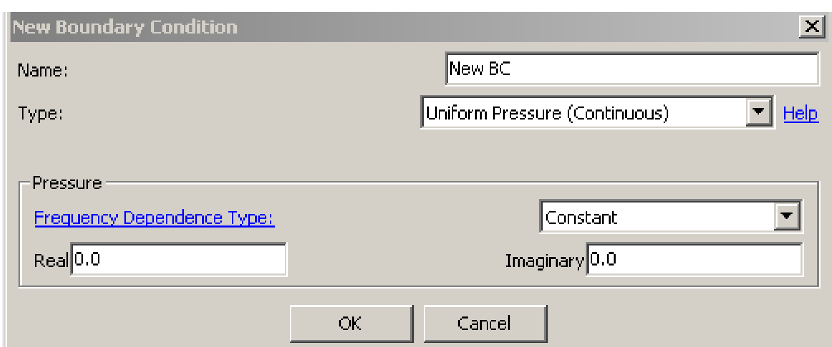


Figure 6.22: Uniform Pressure (Continous)

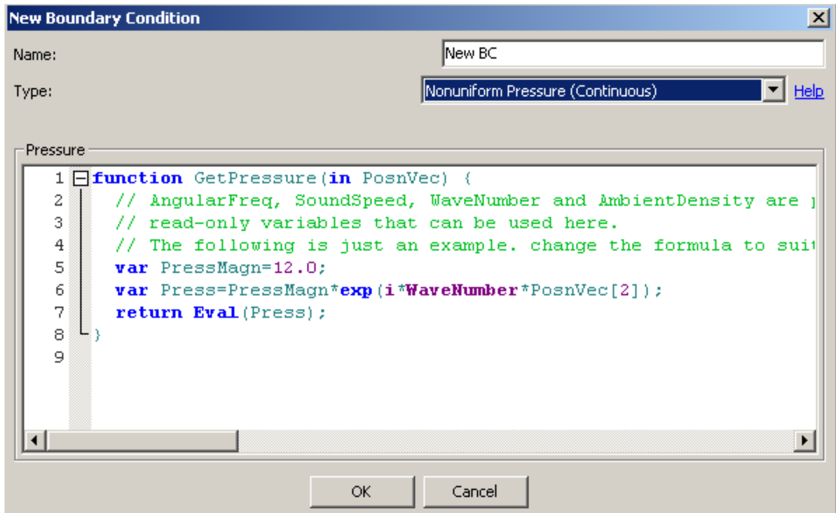


Figure 6.23: Non-Uniform Pressure (Continous)

The pressure value is set by selecting any of the frequency dependent types: *Constant*, *Table*, or *Script*.

6.4.5. Non-uniform Pressure (Continuous) BC

This Boundary Condition is applied on elements with non-uniform pressure on both sides of the boundary. The BC is continuous, which implies that the values of pressure at the same point on side 1 (p^+) and side 2 (p^-) are identical (refer to Figure 6.16 for side 1 and side 2 definitions).

$$p^+(\mathbf{x}) = p^-(\mathbf{x}) = p_o(\mathbf{x})$$

where $p_o(\mathbf{x})$ is the pressure that varies with position ($\mathbf{x}$) on the element.

The pressure is defined in the script by the function *GetPressure*, which takes in the predefined position vector, *PosnVec*, as the argument. Other predefined variables such as *AngularFreq* (ω), *SoundSpeed* (c), *WaveNumber* ($k = \frac{\omega}{c}$), and *AmbientDensity* (ρ) can also be used in the script. (Figure 6.23.)

6.4.6. Uniform Normal Velocity (Continuous) BC

This Boundary Condition is applied on elements with uniform normal velocity on both sides of the boundary. The BC is continuous, which implies that the values of normal velocity at the same point on side 1 (v_n^+) and side 2 (v_n^-) are identical (refer to Figure 6.16 for side 1 and side 2 definitions).

$$v_n^+ = v_n^- = v_{no}$$

where v_{no} is the uniform normal velocity applied on both sides.

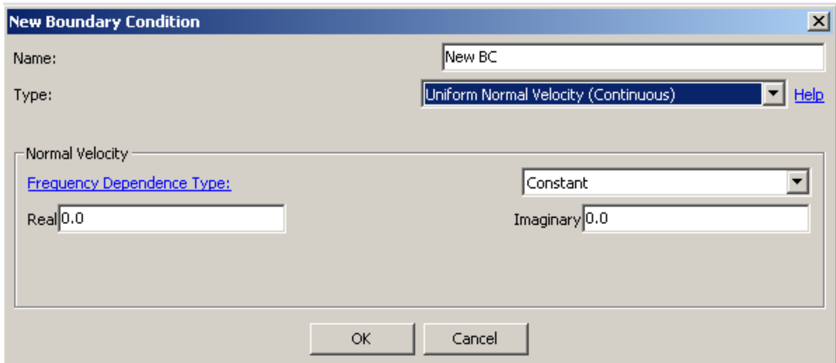


Figure 6.24: Uniform Normal Velocity (Continous)

The normal velocity value is set by selecting any of the frequency dependent types: *Constant*, *Table*, or *Script*. The user should recognize that the normal velocity defined here is with respect to the element normal. (Figure 6.24)

6.4.7. Uniform Velocity (Continuous) BC

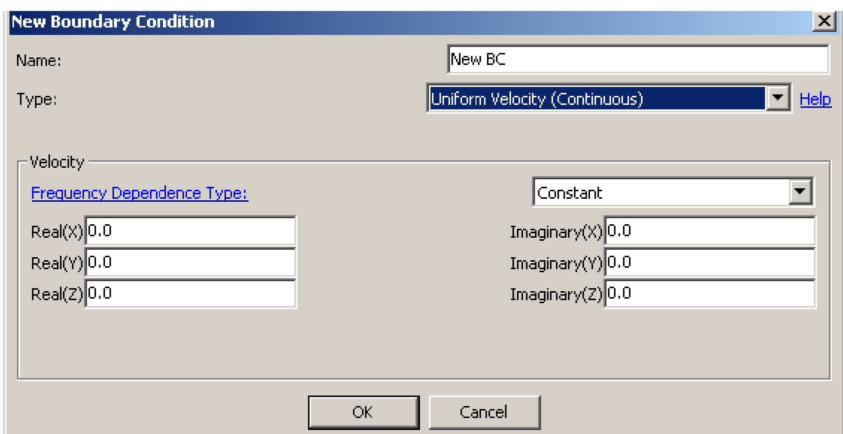


Figure 6.25: Uniform Velocity (Continous)

This Boundary Condition is applied on the element with uniform velocity vector on both sides of the boundary. The BC is continuous, which implies that the velocity at the same point on side 1 ($\mathbf{v}^+$) and side 2 ($\mathbf{v}^-$) are identical (refer to Figure 6.16 for side 1 and side 2 definitions).

$$\mathbf{v}^+ = \mathbf{v}^- = \mathbf{v_o}$$

where $\mathbf{v_o}$ is the uniform velocity applied on both sides.

The velocity vector components (v_x, v_y, v_z) are uniform over the element and are defined using any of the frequency dependence types: *Constant*, *Table*, or *Script*. (Figure 6.25)

6.4.8. Non-uniform Normal Velocity (Continuous) BC

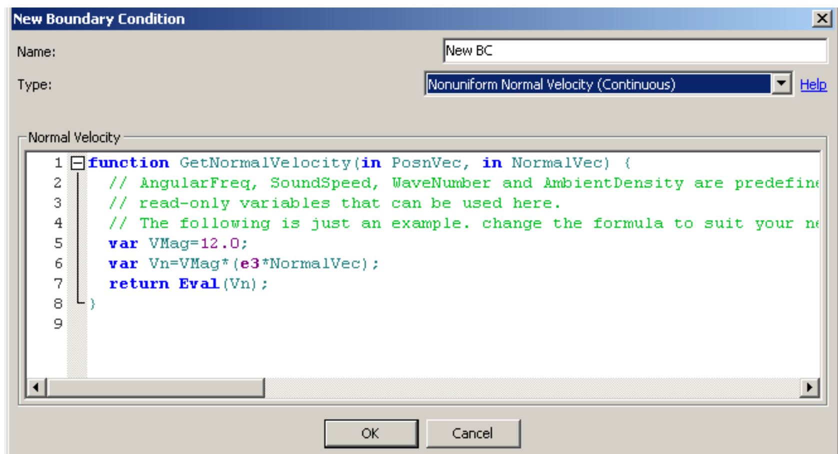


Figure 6.26: Non-Uniform Normal Velocity (Continous)

This Boundary Condition is applied on the element with non-uniform normal velocity on both sides of the boundary. The BC is continuous, which implies that the normal velocity at the same point on side 1 (v_n^+) and side 2 (v_n^-) are identical (refer to Figure 6.16 for side 1 and side 2 definitions). (Figure 6.26)

$$v_n^+(\mathbf{x},\mathbf{n}) = v_n^-(\mathbf{x},\mathbf{n}) = v_{no}(\mathbf{x},\mathbf{n})$$

where $v_{no}(\mathbf{x},\mathbf{n})$ is the normal velocity that varies with position ($\mathbf{x}$) and normal velocity ($\mathbf{n}$) on the element.

The normal velocity is defined in the script by the function *GetNormalVelocity*, which takes in the predefined position vector *PosnVec*, and normal vector *NormalVec* as the arguments. Other predefined variables such as *AngularFreq* (ω), *SoundSpeed* (c), *WaveNumber* ($k = \frac{\omega}{c}$), *AmbientDensity* (ρ), and reference frame unit vectors **e1**, **e2**, **e3** can also be used in the script. (Figure 6.26)

6.4.9. Structure Velocity (Continuous) BC

Structure Velocity BC is the most common Boundary Condition in the industry. This can be loaded from frequency response analysis of the structure FEA done externally. Refer to Section 5.1.2 for details on how to load frequency response data. (Figure 6.27)

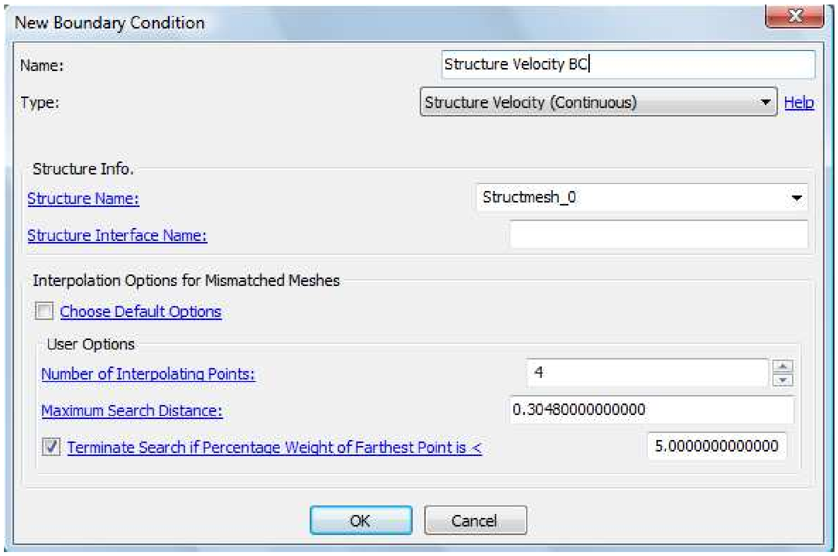


Figure 6.27: Structure Velocity (Continous)

6.4.9.1. Structure Name

Select the structure name from which the BCs are applied. (Figure 6.27)

6.4.9.2. Structure Interface Name

Select the structure interface from which the BCs are applied. A structure interface is a group of faces (of FEA elements) in contact with the fluid medium. Each structure interface is uniquely identified by the structural mesh ID, element ID, and the local face number on the element. If this field is left blank, the structure velocity is applied directly from the structure. (Figure 6.27)

6.4.9.3. Interpolation Options for Mismatched Meshes

Refer to Section 6.2.

6.4.10. Discontinuous BC

This Boundary Condition allows the user to apply different types of boundary conditions on each side of the boundary. Presently no other commercially available acoustic BEM software provides these options. The discontinuous BCs in conventional acoustic softwares are limited to the same boundary condition type (known pressure, known velocity or known impedance) to be specified on both sides of the surface. But *Coustyx* allows different BC types on either side. For example, pressure can be specified on one side, and velocity on the other side. This complete decoupling of the boundary conditions allows for greater modeling flexibility for the user.

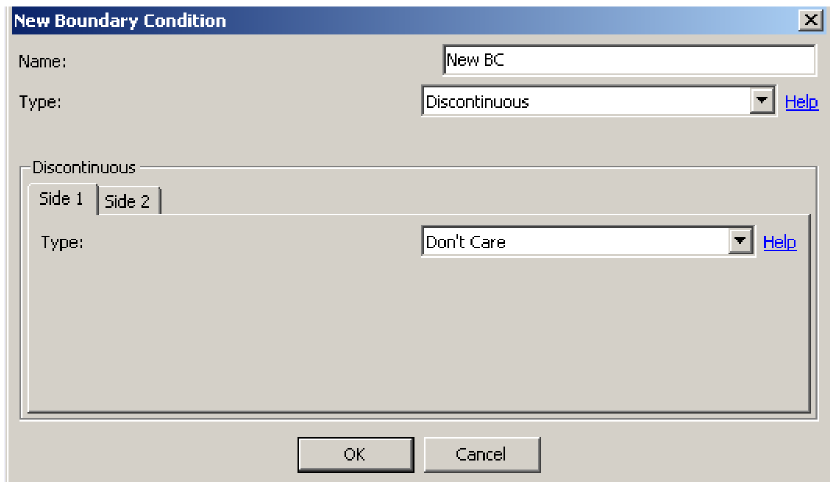


Figure 6.28: Discontinous BC (Don't Care)

Combinations of different types of BCs on each side of the boundary can be applied to overcome the non-uniqueness problems, which are very common in boundary element methods. An example of this would be a determining the exterior sound field on a vibrating surface which encloses a volume. At the resonance frequencies of the interior, the interior acoustic variables become very large and dominate the values of surface potentials. This leads to error in the computation of the exterior field. However, accurate results for this exterior problem can be obtained with *Coustyx* by specifying a velocity boundary condition on the exterior and a *zero velocity* or *zero pressure* boundary condition on the interior.

The discontinuous BC gives the option of specifiying different BCs on *Side 1* and *Side 2*. Refer Figure 6.16 for the definitions of *Side 1* and *Side 2*. Note that *Coustyx* always considers *Side 1* of the boundary to be on the positive side (+) of the normal and *Side 2* on the negative side (-). That is, always the element normal points from *Side 2* to *Side 1*. The possible side boundary conditions allowed in *Coustyx* are listed below. (Figure 6.28)

6.4.10.1. Don't care

This Boundary Condition is applied on the side of the boundary which doesn't come in contact with the fluid medium or on the side the user is not interested in. Again, consider the example of a vibrating surface which encloses a volume and the primary interest is in the exterior sound field. The boundary condition on the interior side of the mesh doesn't affect the exterior solution. In this case the user can apply *Don't care* BC on the interior side of the mesh. (Figure 6.28)

6.4.10.2. Uniform Pressure

This Boundary Condition is applied on the side of the element where the pressure is uniformly distributed. That is, there is no variation of pressure with position. The pressure values can

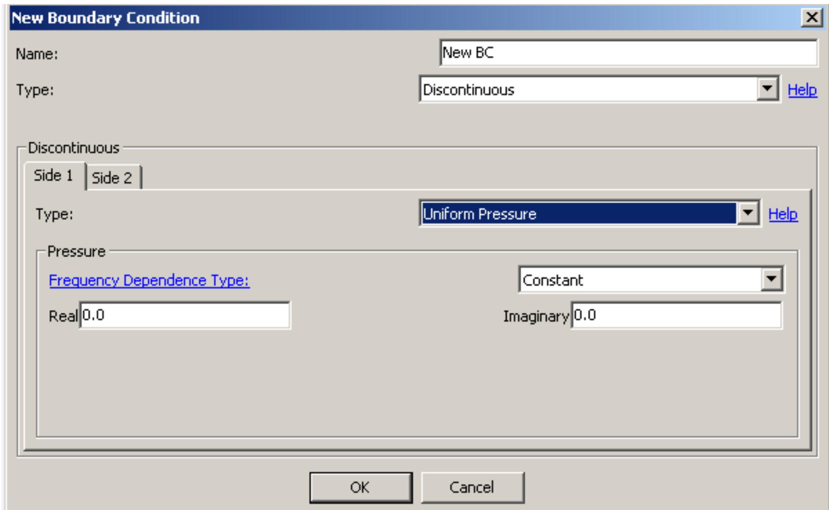


Figure 6.29: Discontinous Uniform Pressure

be specified by selecting any of the frequency dependence types: *Constant*, *Table*, or *Script*. (Figure 6.29)

6.4.10.3. Non-uniform Pressure

This Boundary Condition is applied on the side of the element where pressure varies with position. The pressure is defined in the script by the function *GetPressure*, which takes in the predefined position vector *PosnVec* as the argument. Other predefined variables that can be used in the script are *AngularFreq* (ω), *SoundSpeed* (c), *WaveNumber* ($k = \frac{\omega}{c}$), and *AmbientDensity* (ρ). (Figure 6.30)

6.4.10.4. Uniform Normal Velocity

This Boundary Condition is applied on the side of the element where the normal velocity is uniformly distributed. The normal velocity values can be specified by selecting any of the frequency dependence types: *Constant*, *Table*, or *Script*. The user should be aware that the normal velocity defined here is in the direction of the element normal (Figure 6.16). (Figure 6.31)

Use Impedance can be used to specify impedance. Refer to Section 6.3 for more details. (Figure 6.31)

6.4.10.5. Uniform Velocity

This Boundary Condition is applied on the side of the element where the velocity vector is uniformly distributed. That is, there is no variation of velocity components (v_x, v_y, v_z) with position over an element. The velocity vector components values can be specified by selecting any of the frequency dependence types: *Constant*, *Table*, or *Script*.

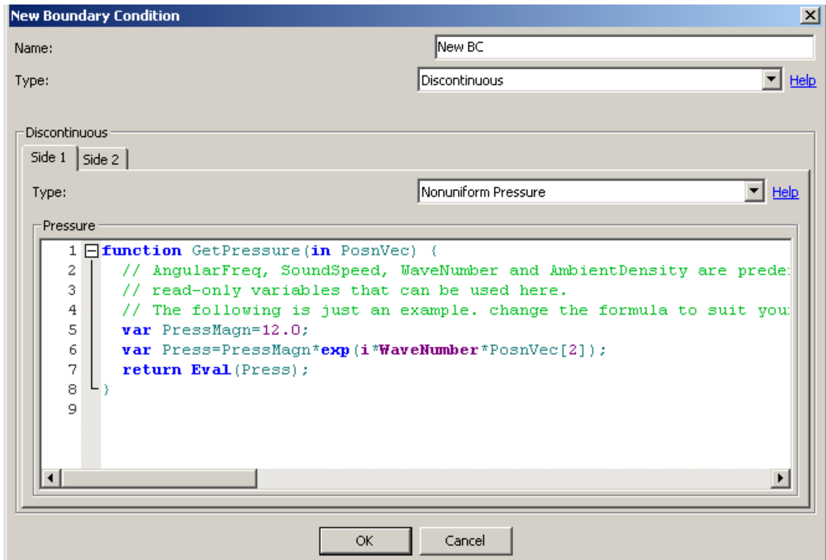


Figure 6.30: Discontinous Non-Uniform Pressure

Use Impedance can be used to specify impedance. Refer to Section 6.3.6.1 for more details. (Figure 6.32)

6.4.10.6. Non-uniform Normal Velocity

This Boundary Condition is applied on the side of the element where normal velocity varies with position and normal. The normal velocity is defined in the script by the function *GetNormalVelocity*, which takes in the predefined position vector *PosnVec* and normal vector *NormalVec* as the arguments. Other predefined variables that can be used in the script are *AngularFreq* (ω), *SoundSpeed* (c), *WaveNumber* ($k = \frac{\omega}{c}$), *AmbientDensity* (ρ), and reference frame unit vectors **e1**, **e2**, **e3**. (Figure 6.33)

Use Impedance can be used to specify impedance. Refer to Section 6.3.7.1 for more details. (Figure 6.33)

6.4.10.7. Structure Velocity

This Boundary Condition is applied on the side of the element if the frequency response data is known from the FEA analysis of the structure (Figure 6.34). Refer to Section 5.1.2 for details on how to load frequency response data. **Structure Name** is the name of structure from which the velocity values are loaded. **Structure Interface Name** is the interface, group of element faces, which is used to apply BC. More details on Structure Name, and Structure Interface Name are given in Section 6.3.8.1, and Section 6.3.8.2. Refer to Section 6.2 for details about **Interpolation Options for Mismatched Meshes**. (Figure 6.34)

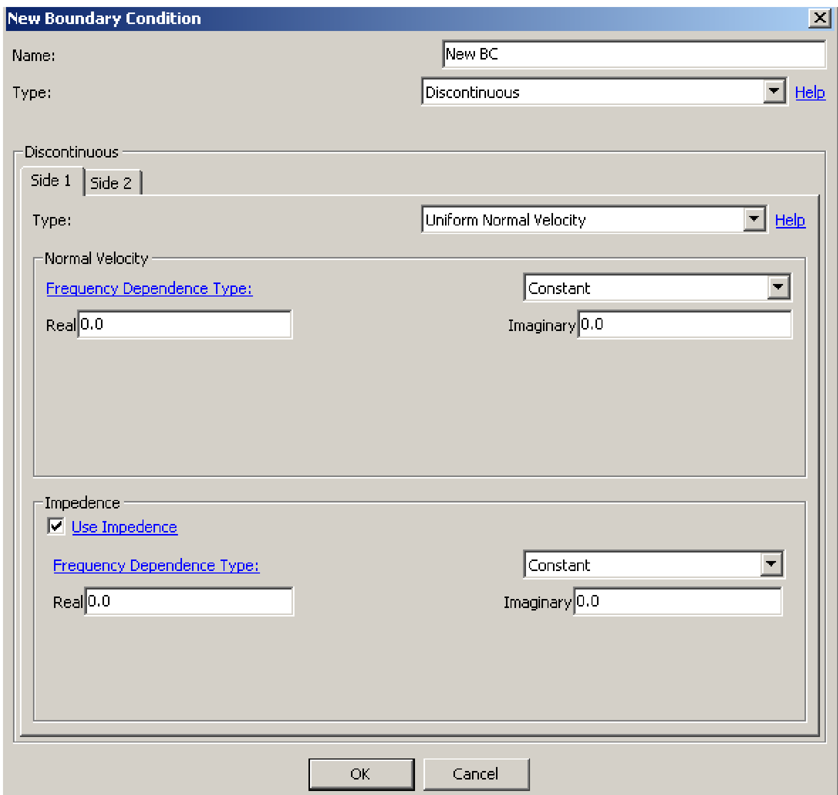


Figure 6.31: Discontinous Uniform Normal Velocity

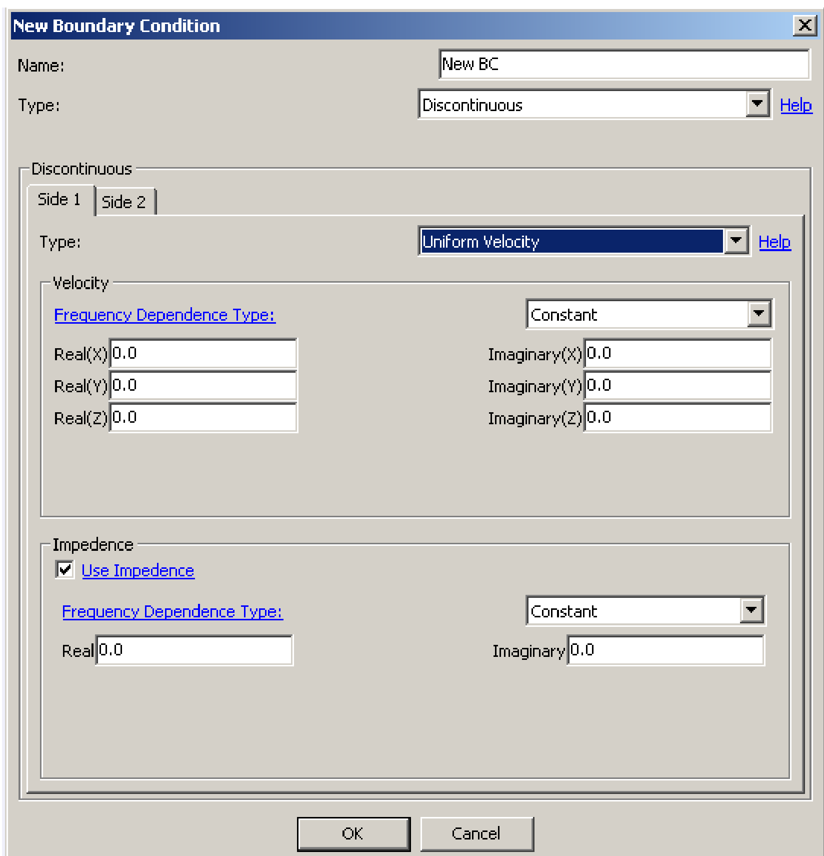


Figure 6.32: Discontinous Uniform Velocity

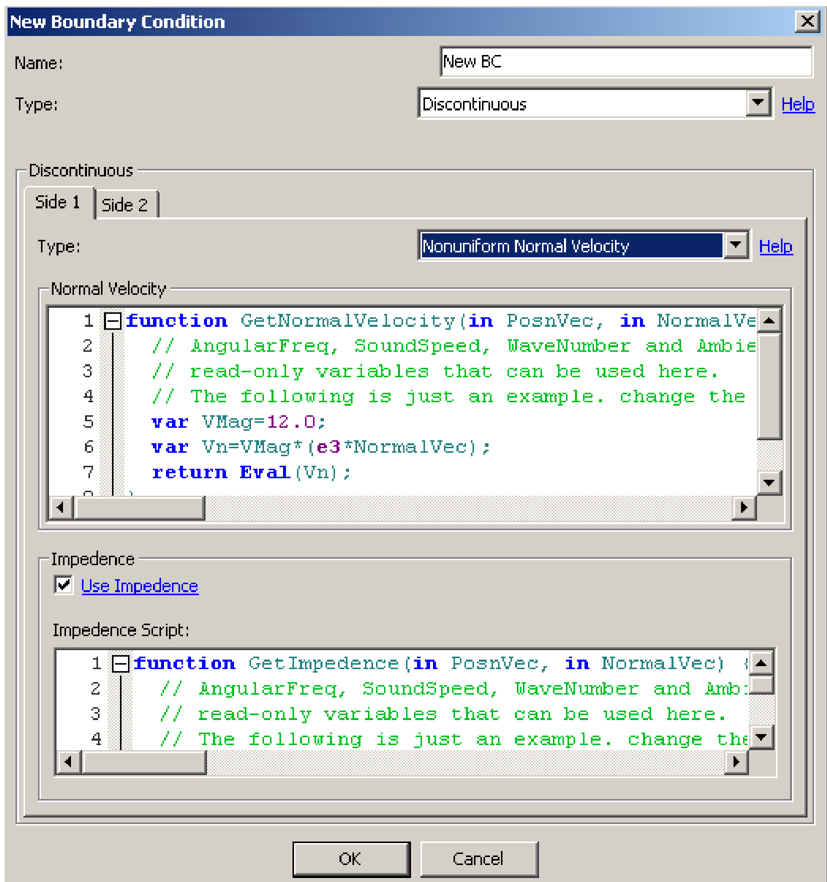


Figure 6.33: Discontinous Non-Uniform Normal Velocity

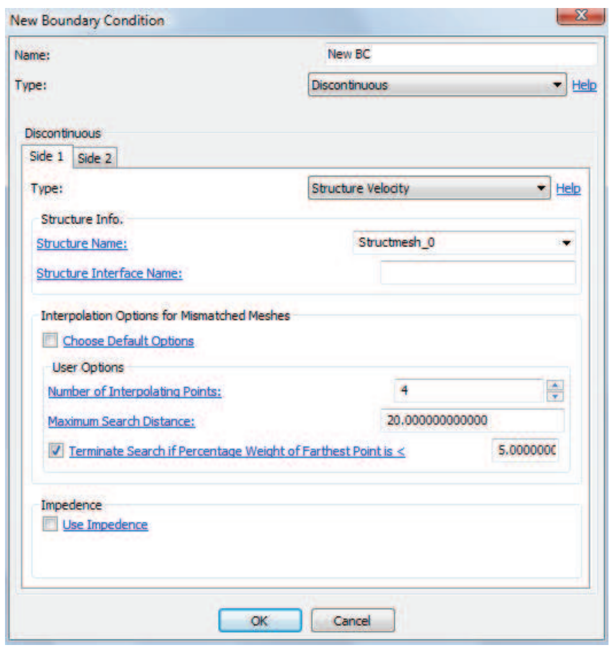


Figure 6.34: Structure Velocity

6.4.11. Uniform Arbitrary BC

This Boundary Condition option could be used to define uniform boundary conditions which don't fall in any of the other BC types. This is the most general type of boundary condition possible. Any of the uniform BCs can be expressed in this form. The user can specify the relation between the pressure and normal velocity on one side of the element to the other side by the following transfer function. (Figure 6.35)

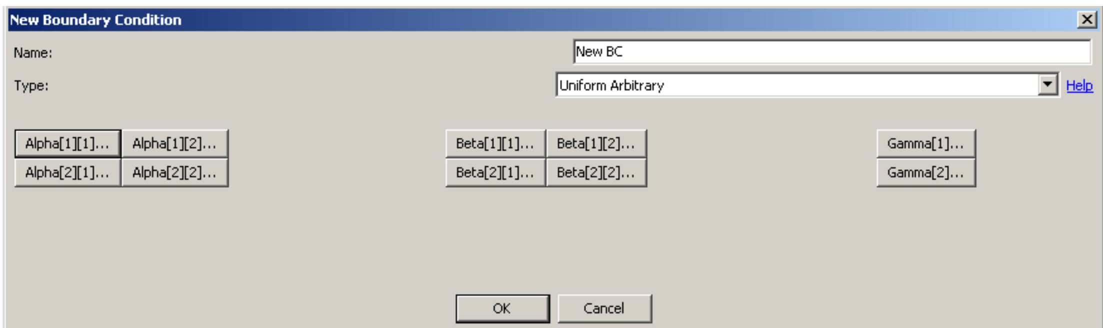


Figure 6.35: Uniform Arbitrary BC

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} p^{+} \\ p^{-} \end{bmatrix} + \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix} \begin{bmatrix} v_n^{+} \\ v_n^{-} \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \tag{6.10}$$

where p^+ and p^- are pressures on side 1 and side 2, v_n^+ and v_n^- are normal velocity on side 1 and side 2.

The variables α_{11}, α_{12}, α_{21}, α_{22}, β_{11}, β_{12}, β_{21}, β_{22}, γ_1, γ_2 are uniform over the element and are defined by any of the frequency dependent types: *Constant*, *Table*, or *Script*. These values do not vary with position over the element.

All the entries in the coefficient matrices of Equation 6.10 are not independent. They should satisfy the following relations for *Coustyx* to apply the arbitrary BC correctly.

$$\begin{aligned} (\alpha_{11}\alpha_{22} - \alpha_{12}\alpha_{21}) &\neq 0 \\ (\beta_{11}\beta_{22} - \beta_{12}\beta_{21}) &\neq 0 \\ (\alpha_{21} + \alpha_{22})(\beta_{11} + \beta_{12}) - (\alpha_{11} + \alpha_{12})(\beta_{21} + \beta_{22}) &\neq 0 \\ \alpha_{11}\beta_{21} - \alpha_{21}\beta_{11} - \alpha_{12}\beta_{22} + \alpha_{22}\beta_{12} &= 0 \end{aligned} \tag{6.11}$$

6.4.12. Non-uniform Arbitrary BC

This Boundary Condition option could be used to define non-uniform boundary conditions which do not fall into any of the other BC types. Any of the non-uniform BC type can be expressed in this general form. The general transfer relation between pressures and normal velocities on either side of the boundary at any point is given by Equation 6.10.

The variables α_{11}, α_{12}, α_{21}, α_{22}, β_{11}, β_{12}, β_{21}, β_{22}, γ_1, γ_2 can be defined by the function call *GetAlphaBetaGamma* (Figure 6.36). The input arguments position vector *PosnVec*, and normal vector *NormalVec*, along with other predefined variables such as *AngularFreq* (ω), *SoundSpeed* (c), *WaveNumber* ($k = \frac{\omega}{c}$), *AmbientDensity* (ρ), and reference frame unit vectors **e1**, **e2**, **e3**, can be used to compute the values. The output arguments of this function call include the following variables. See Figure 6.37, Figure 6.38, and Figure 6.39.

$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix}, \quad \gamma = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \tag{6.12}$$

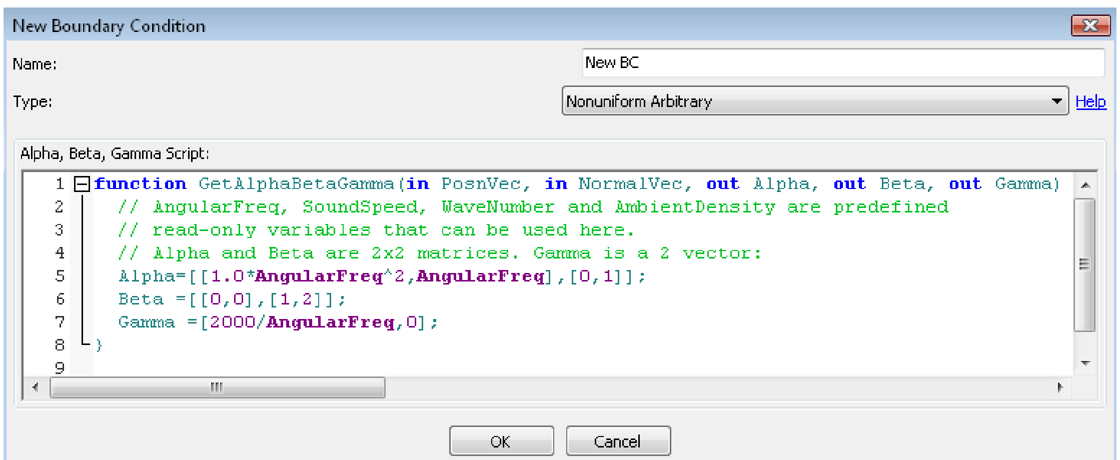


Figure 6.36: Sample script for the function GetAlphaBetaGamma().

6.5. Applying BCs

First, the user needs to define boundary conditions and save them using unique names. For information on how to define boundary conditions in *MultiDomain* and *Indirect* BE models refer to Section 6.3 and Section 6.4.

Once the boundary conditions are defined they are applied over elements by any of the following methods:

6.5.1. Apply BCs directly to Elements

To apply BCs directly to elements follow these steps:

- Open the BE mesh in the GUI.

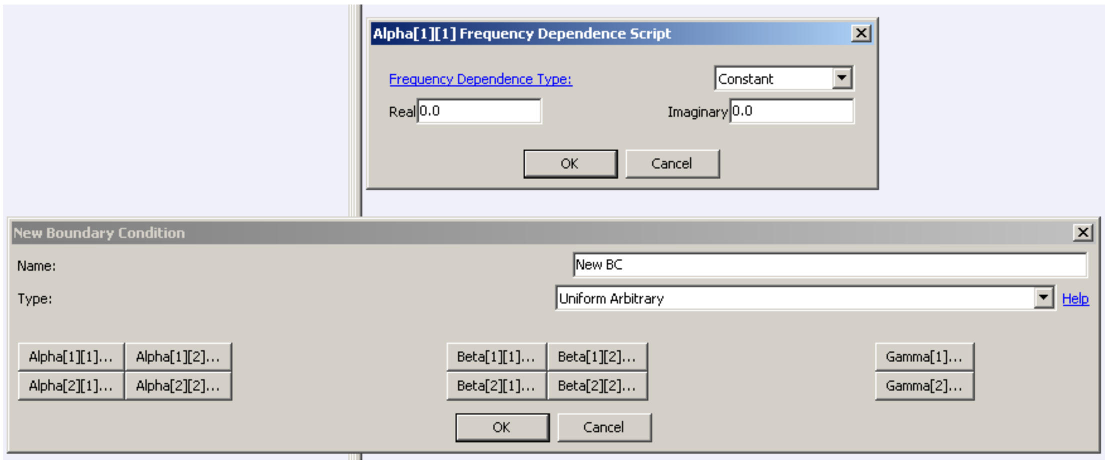


Figure 6.37: Alpha Frequency Dependence Script

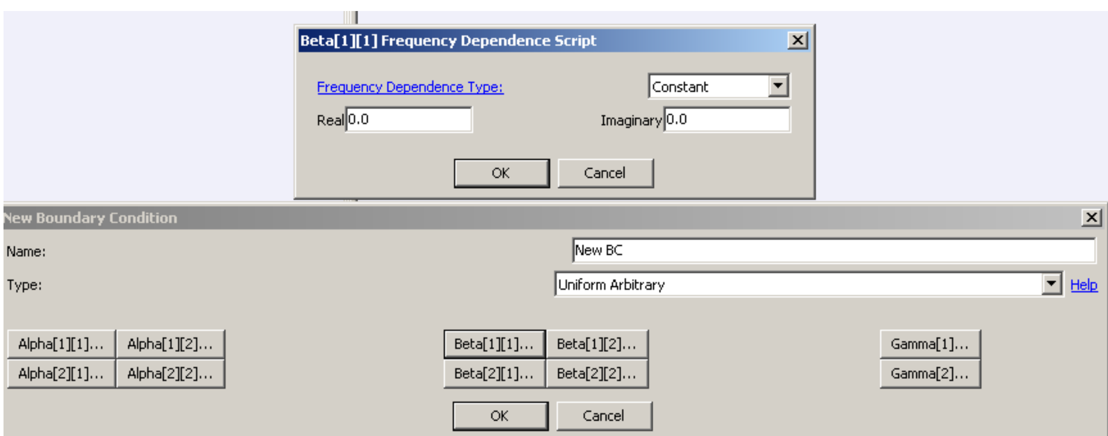


Figure 6.38: Beta Frequency Dependence Script

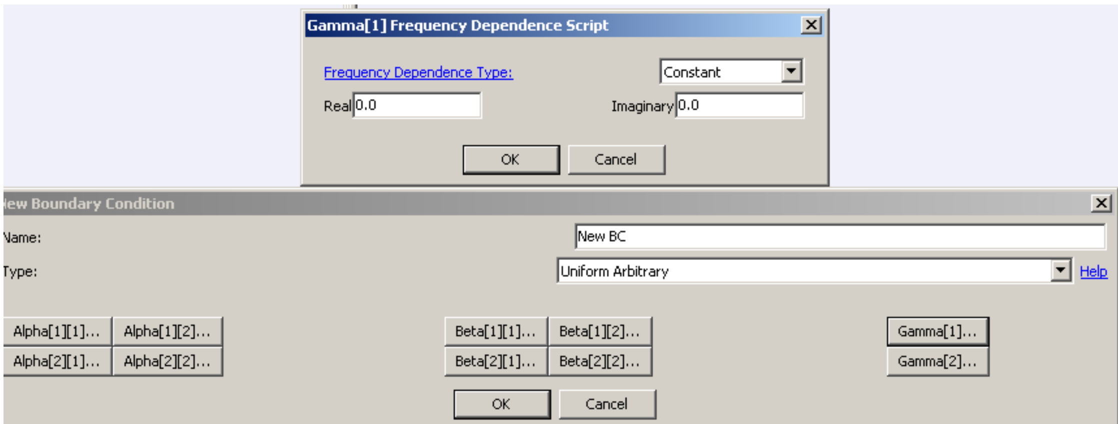


Figure 6.39: Gamma Frequency Dependence Script

For a *MultiDomain* model: right-click on **Model** → **Direct BE Meshes** → <*Direct BE Mesh Name*> and select **Open**.

For an *Indirect* model: right-click on **Model** → **Indirect BE Mesh** and select **Open**.

- Select the elements to which the BCs are applied from the mesh in the GUI. To select an element: Left-click on the element with the *shift-key* held down.
- Apply BC. To apply BC on all selected elements in the GUI, right-click with the *shift-key* held down and select: **Operations on Selection** → **Selected Elements** → **Set Boundary Condition** → <*Boundary Condition Name*> (Figure 6.40). Note that the boundary condition list is only visible if BCs are defined before hand.

6.5.2. Apply BCs through Sets

To apply BCs through sets, follow these steps:

- Create a new set.

  For a *MultiDomain* model: right-click on **Model** → **Direct BE Meshes** → <*Direct BE Mesh Name*> → **Sets** and select **New**. (Figure 6.41)

  For an *Indirect* model: right-click on **Model** → **Indirect BE Mesh** → **Sets** and select **New**. (Figure 6.41)

- Add elements to a set.

  Select elements in the GUI which are to be clubbed together to form a set by left-clicking on them while holding down the *shift-key*.

  Right-click with the *shift-key* held down and select: **Operations on Selection** → **Selected Elements** → **Add to Set** → <*Set Name*> (Figure 6.42). Note that the list of sets are only visible if sets are defined before hand.

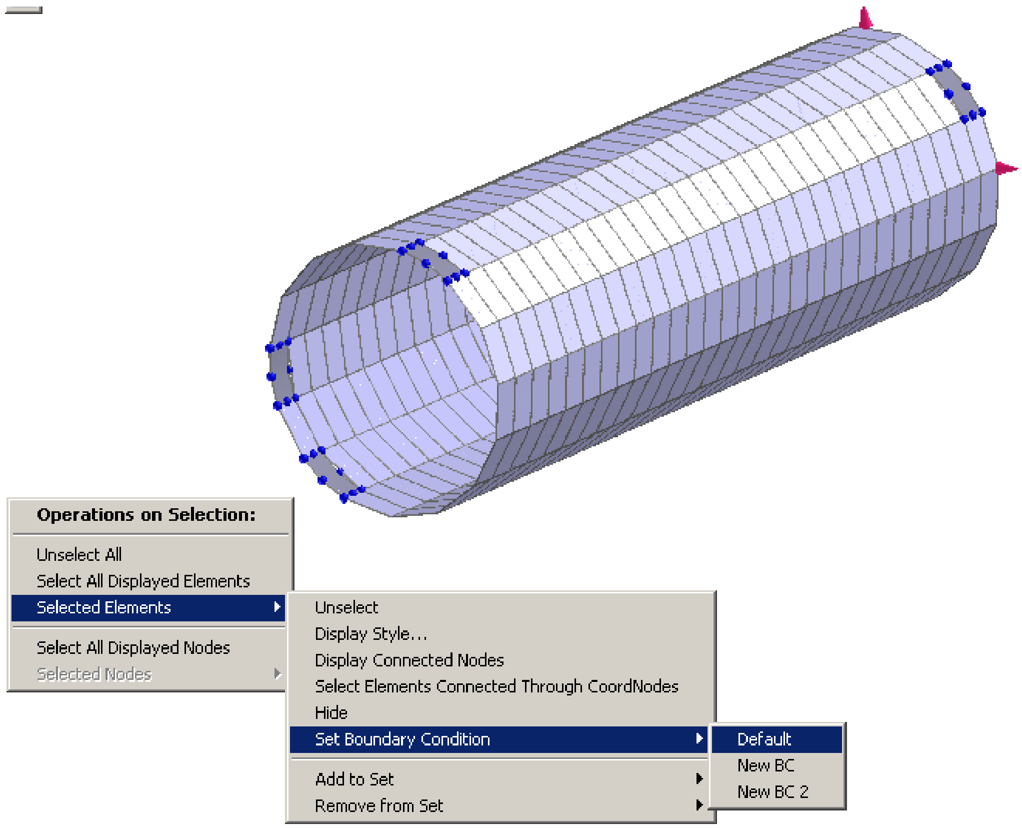


Figure 6.40: Applying boundary conditions through elements.

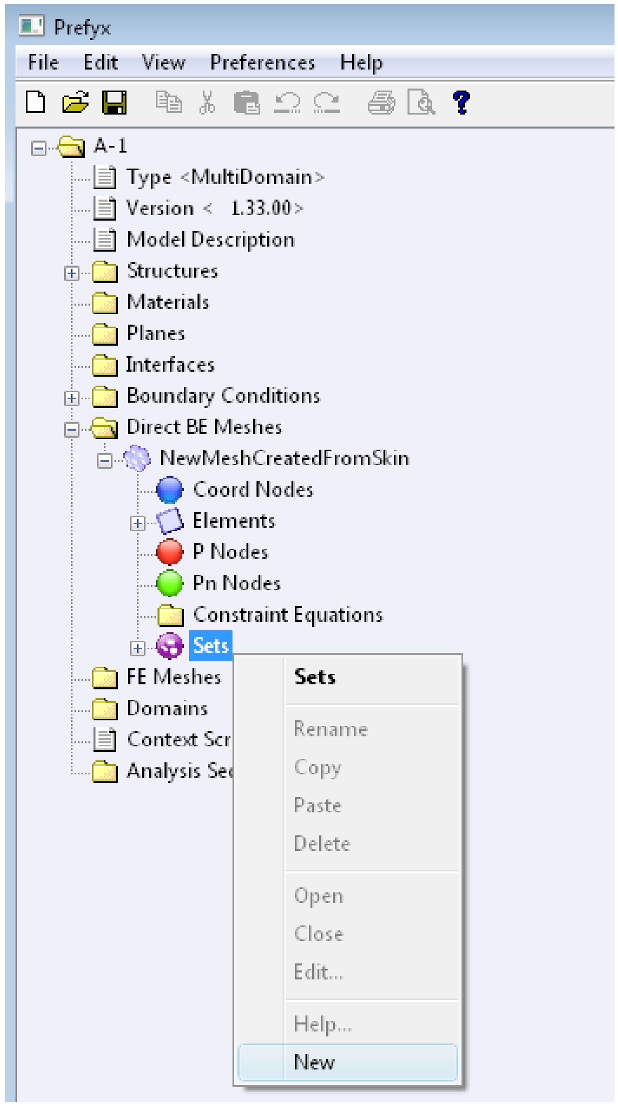


Figure 6.41: Creating a new set, a group of elements and nodes.

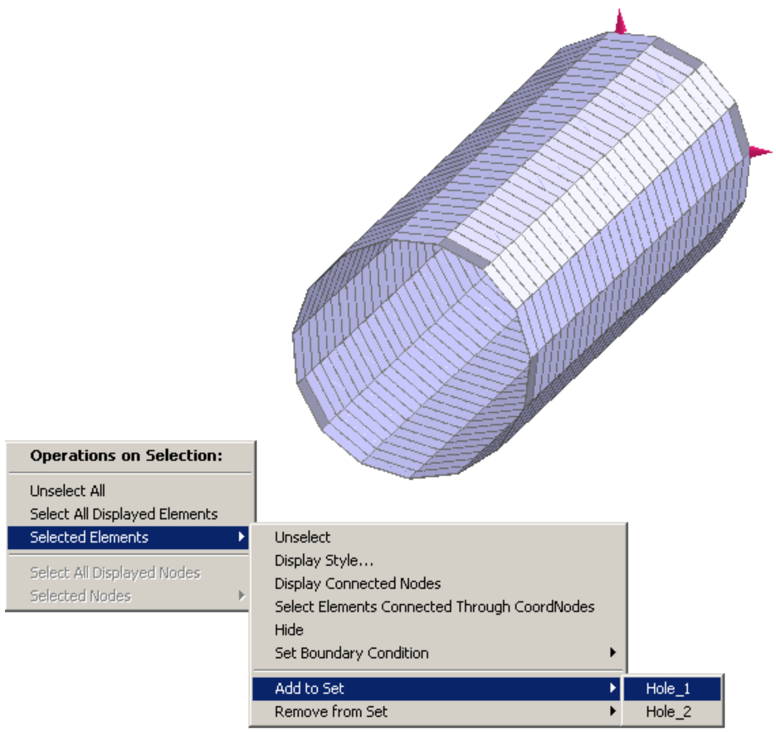


Figure 6.42: Add elements to a set.

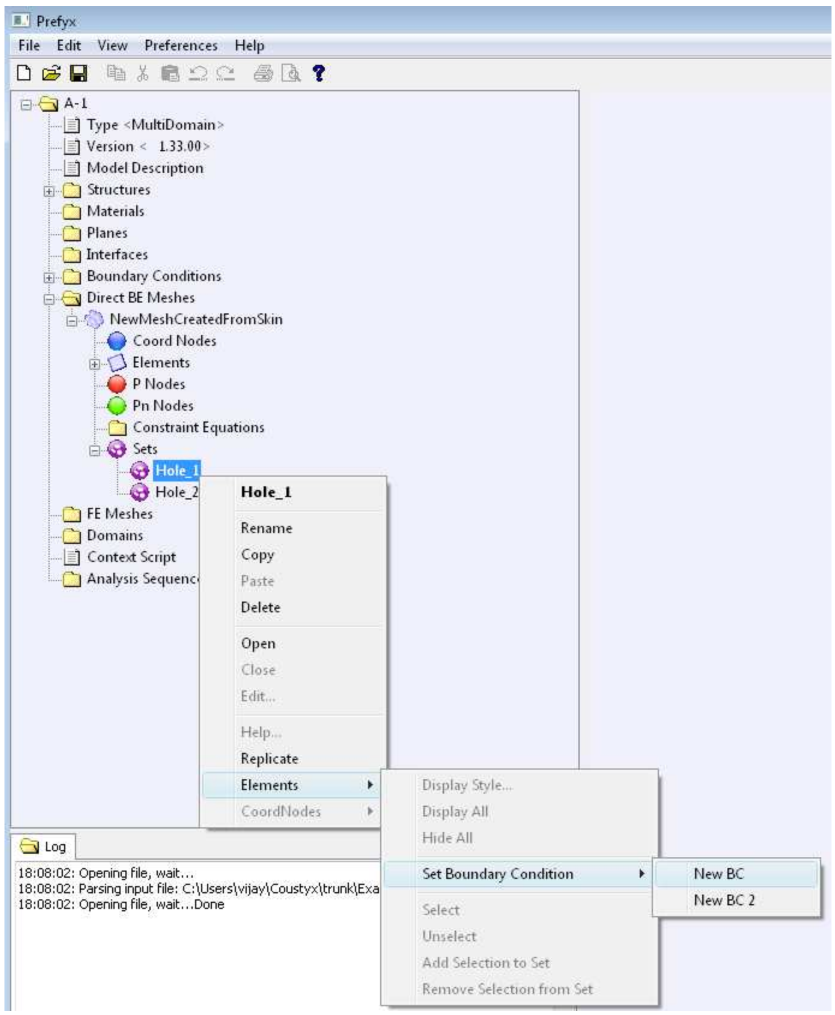


Figure 6.43: Applying boundary conditions through sets.

- Apply BC through sets.

  Select the set for a *MultiDomain* model from **Model** → **Direct BE Meshes** → <*Direct BE Mesh*> → **Sets** → <*Set Name*>.

  Select the set for an *Indirect* model from **Model** → **Indirect BE Mesh** → **Sets** → <*Set Name*>.

  Apply BC to all the elements in the Set through the Set by right-clicking on <*Set Name*> and selecting **Elements** → **Set Boundary Condition** → <*Boundary Condition Name*> (Figure 6.43).

Bibliography

[1] A. D. Pierce. *Acoustics - An introduction to its physical principles and applications*. Acoustical Society of America, 1991. Pages 107–113.

[2] V. K. Ambarisha, R. Gunda, and S. M. Vijayakar. A new indirect formulation to address the non-uniqueness problem in acoustic bem. *INCE conference proceedings*, 116, 2007. Pages 1046–1055.

[3] J. W. Sullivan and M. J. Crocker. Analysis of concentric-tube resonators having unpartitioned cavities. *Journal of Acoustic Society of America*, 64:207–215, 1978.

Chapter 7

Analysis Sequences

Coustyx allows users to specify set of parameters and commands required to run an analysis through model tree member **Analysis Sequences**. Typical set of parameters include type of analysis, solution accuracy, analysis frequency, types of outputs, sensor locations, etc. Multiple analysis sequences could be saved through this option.

Select: **Model** → **Analysis Sequences**.

- **Create a New Analysis Sequence**. To create a new analysis sequence right-click on **Analysis Sequences** and select **New**. A dialog box with new analysis sequence opens up. Go through all the *tabbed* windows to modify analysis parameters. To accept changes click OK. (Figure 7.1)

- **Edit an Analysis Sequence**. Select: **Analysis Sequences** → <*Analysis Sequence Name*>. Right-click on it and select **Edit** to open the analysis sequence edit dialog box. Make changes and click OK. (Figure 7.2)

- **Run an Analysis Sequence**. To run acoustic analysis right-click on the desired analysis sequence and select **Run**. You could also run a selected analysis sequence from the Main menu: **Analysis** → **Run** or by clicking on the blue run button (Figure 7.3). Make sure *Coustyx* model setup is completed before running the analysis.

- **Abort the analysis run**. To abort a run right-click on the analysis sequence running current analysis and select **Abort**. You could also abort a run from the Main menu: **Analysis** → **Abort** or by clicking on the red abort button (Figure 7.4). *Coustyx* won't stop the analysis run immediately. The analysis continues to run until it reaches a valid state to abort gracefully. This may take a few minutes. See Figure 7.5.

Figure 7.2 shows an Analysis Sequence edit dialog box. The dialog box consists of five *tabbed* windows storing different parameters. The first *tabbed* window **Description** could be used to add the description of the analysis sequence. All the required **Inputs** for *Coustyx* analysis are found in the *tabbed* windows: **Solver Controls** and **Frequency Ranges**. The **Outputs** window has options to save the results to a binary file, compute acoustic variables at Sensor locations (field points), and save iGlass file for post-processing visualization. **Script** *tabbed* window lists the summary of commands required to run an analysis. The parameters in the script are set by the

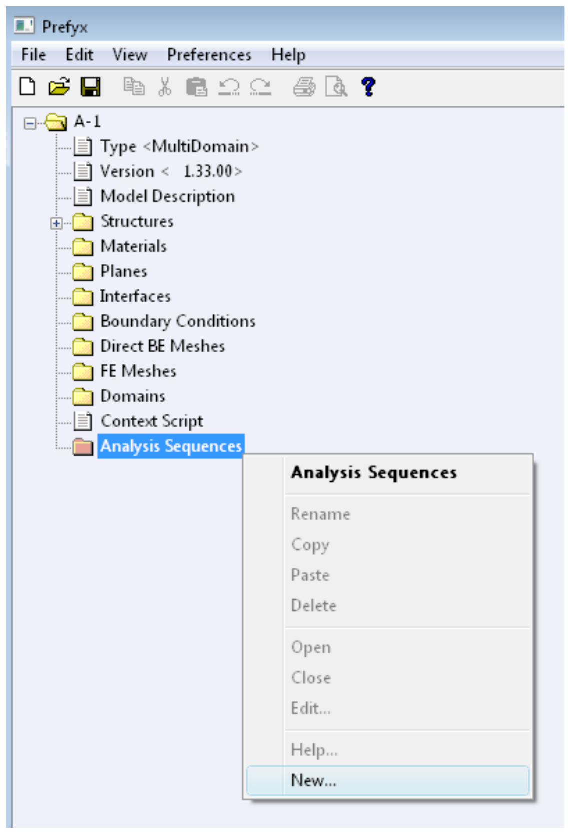


Figure 7.1: Creating new analysis sequence.

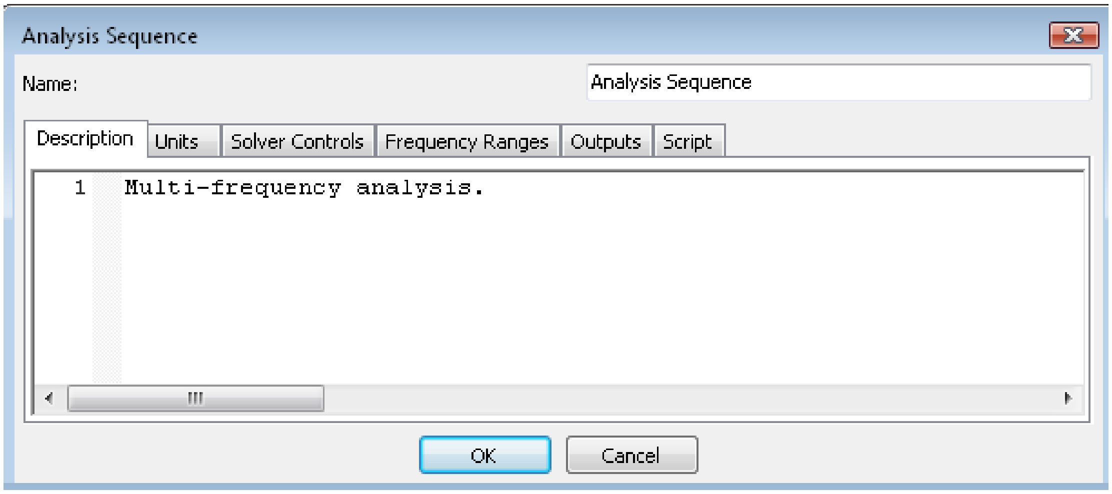


Figure 7.2: Analysis sequence edit dialog box.

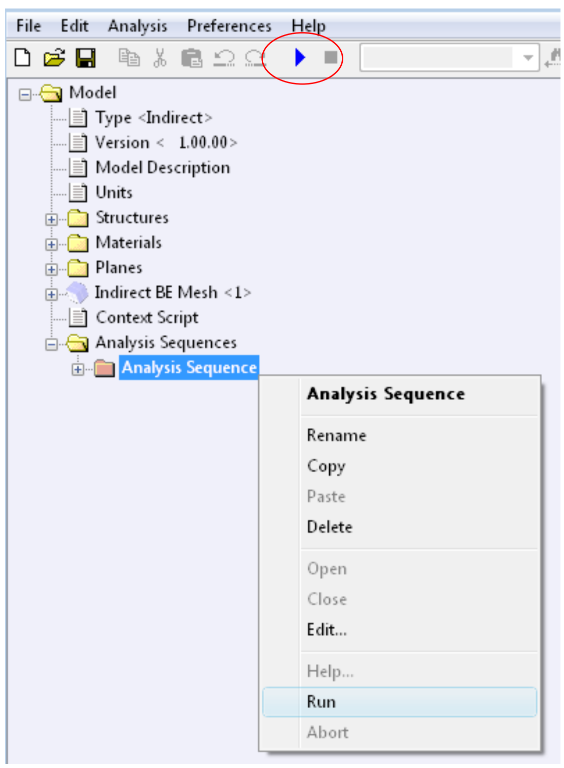


Figure 7.3: Run new analysis sequence.

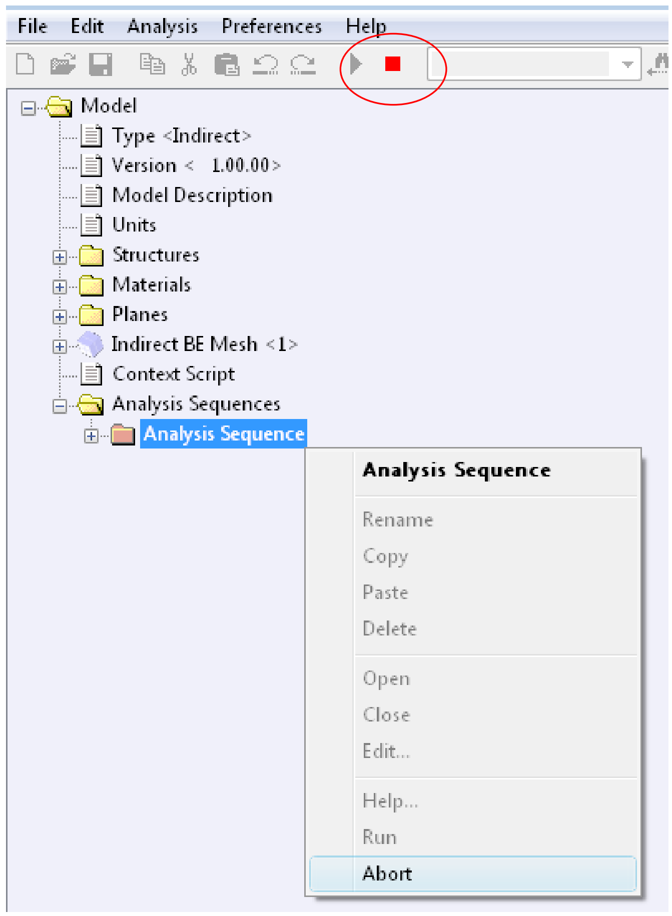


Figure 7.4: Abort the analysis run.

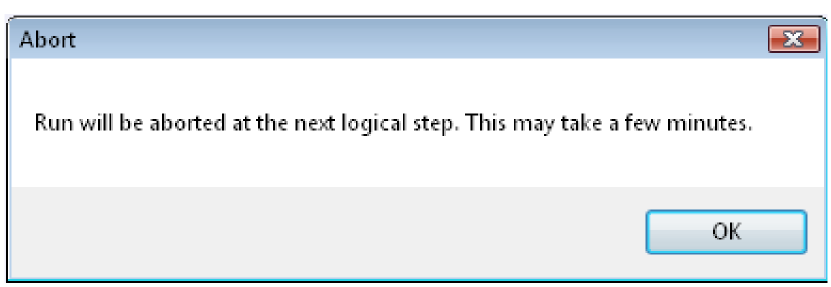


Figure 7.5: Abort message.

selections in the previous *tabbed* windows. Advanced users can directly modify the script to suit their requirements.

Users should exercise reasonable caution in the selection of the input parameters due to their influence on the solution accuracy and the speed of the analysis.

7.1. Inputs

Input parameters required to run an analysis are spread over **Solver Controls** and **Frequency Ranges** *tabbed* windows.

7.1.1. Solver Controls

The **Solver Controls** window is shown in Figure 7.6.

7.1.1.1. Parallel Processing

The bulk of the computations in *Coustyx* are parallelized on shared-memory multi-CPU machines. To use system resources effectively, the user can select the number of CPUs to be available for *Coustyx*. (Figure 7.6)

Use Maximum Possible Number of CPUs This option can be checked if the user wants *Coustyx* to choose the number of CPUs. *Coustyx* selects either the maximum CPUs available to the system, or the maximum number of CPUs allowed by the user's license, whichever is smaller.

Number of CPUs requested This option is activated by un-checking the option **Use Maximum Possible Number of CPUs**. The user can then select the number of CPUs to be available to *Coustyx*. If the number of CPUs selected is higher than the maximum allowed by the user's license, *Coustyx* automatically selects the latter. (Figure 7.6)

7.1.1.2. Formulation Type

This option is valid only for *MultiDomain* models. For *Indirect* models variational formulation is the only choice available.

Variational Select this option to perform the analysis using the variational formulation.

HIE Collocation Select this option to perform the analysis using the Helmholtz Integral Equation (HIE) collocation method.

Burton Miller Galerkin Select this option to perform the analysis using the Burton Miller formulation.

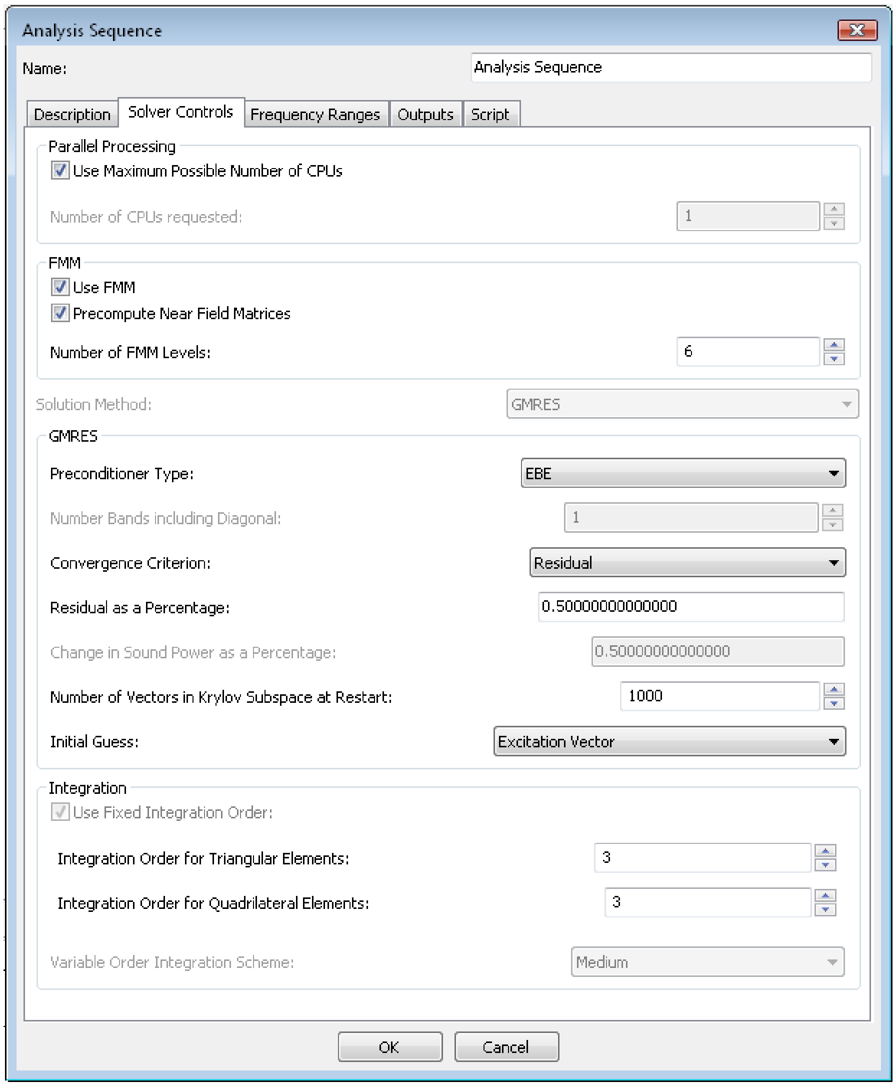


Figure 7.6: Solver controls window.

7.1.1.3. FMM

The Fast Multipole Method (FMM) facilitates fast computation of acoustic fields by an ensemble of point sources at a large number of observation points. System matrices are not computed, instead an iterative solver is used to solve the matrix-vector product computed by FMM for each iteration. This reduces the memory usage for large problems. A *Regular method* (default option when FMM is not chosen) can only solve a maximum of 10,000 unknowns on a 2GB machine. However, FMM is capable of solving much larger problems. Acoustic simulations using FMM are 50 times faster at 10,000 unknowns compared to the regular method.

Use FMM This option is selected when the user wants *Coustyx* to use Fast Multipole Method (FMM) for the analysis. If this option is un-ticked, *Coustyx* uses *Regular method* to do the analysis. In a *Regular method Coustyx* computes system matrix and solves the linear system of equations for the unknowns.

Precompute Near Field Matrices This option is enabled only when the option **Use FMM** is selected. If this option is enabled, *Coustyx* computes near field matrices at the beginning of each frequency and stores them for usage in matrix-vector computations at each iteration. If this option is un-ticked, *Coustyx* computes the near field matrices during each iteration. This increases the total analysis run time, but helps avoid using memory to store these matrices. If memory usage is not an issue we advise the users to precompute near field matrices for speeding up the analysis. (Figure 7.6)

FMM Transition Method This option is enabled only when the option **Use FMM** is selected.

- **Speed** Select this option for faster run times. The solution is accurate at higher frequencies. However, the solution may be less accurate at low frequencies.
- **Best blend** Select this option to blend **Speed** and **Accuracy** methods. This method uses **Speed** method at higher frequencies and **Accuracy** method for low frequencies.
- **Accuracy** Select this option to obtain accurate solutions at all frequencies. This method, however, may be slower than the other two methods.

Number of FMM Levels This option is enabled only when the option **Use FMM** is selected. Proper selection of number of FMM levels is very important to run the analysis efficiently. Always use the *Levels Suggested* in the edit dialog box (Figure 7.7).

Level 0 represents a cube (root cell) enclosing the entire acoustic model. Level 1 represents eight children cells formed by sub-dividing the root cell into eight octants. Each of the children cells are further divided into eight more to represent the next level. This is applied recursively until we reach the lowest level. Figure 7.8 shows a cell hierarchy. FMM is applied only for far field cells. All near field interactions are computed directly. A minimum of Level 2 is required to apply FMM to the acoustic problem (refer to Figure 7.8), far cells appear only for number of FMM levels ≥ 3). As a general rule of thumb, the number of FMM levels are selected such that the edge length of the cell in the final level is similar to the size of the elements in the boundary element mesh. For example, consider a cube of size $a \times a \times a$ with a mesh of 30×30 elements on each side. Select the $number of \text{FMM} levels = 6$. The size of the cell at the final level is $a/32$, which is of the same order as the element length $a/30$. The *Levels Suggested* in the analysis sequence is computed similarly.

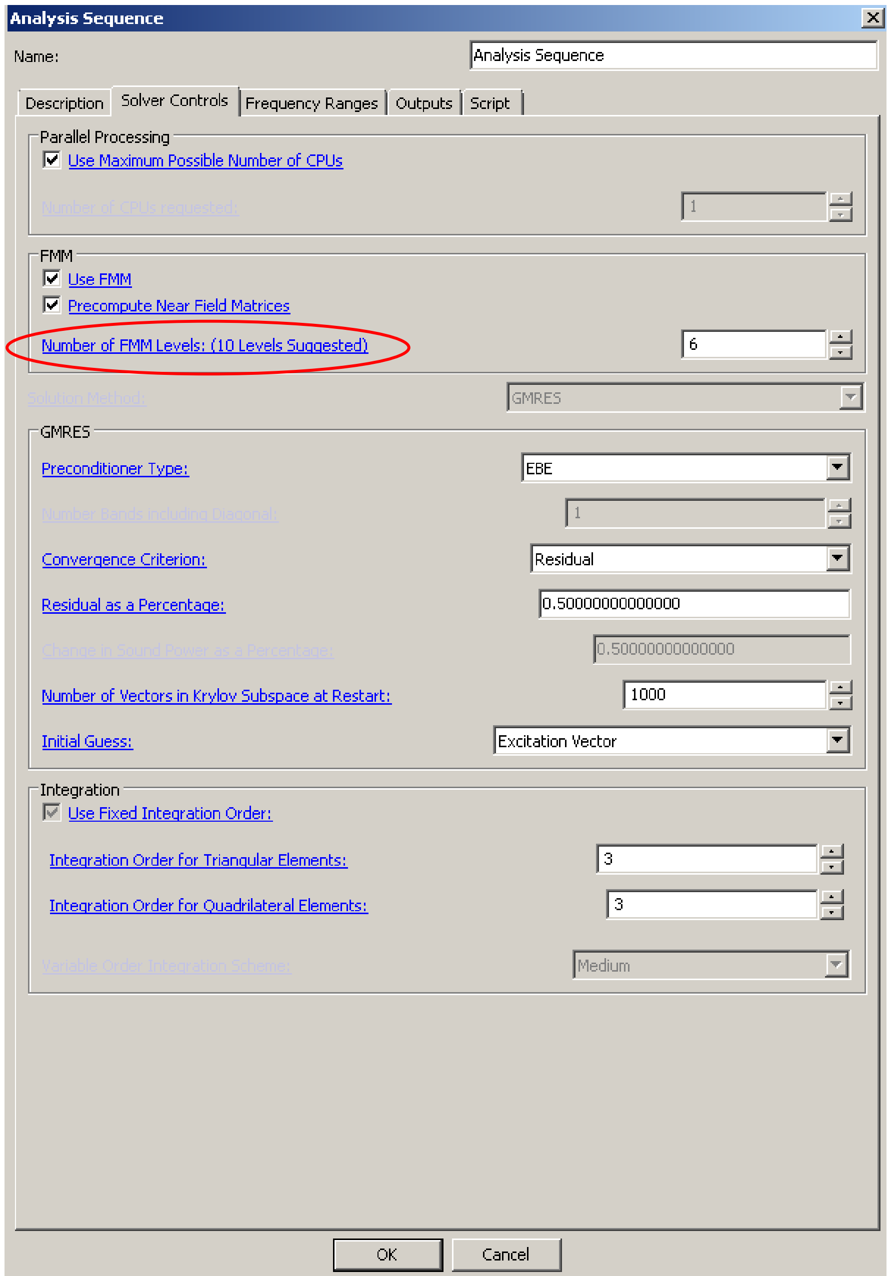


Figure 7.7: Suggested number of FMM levels.

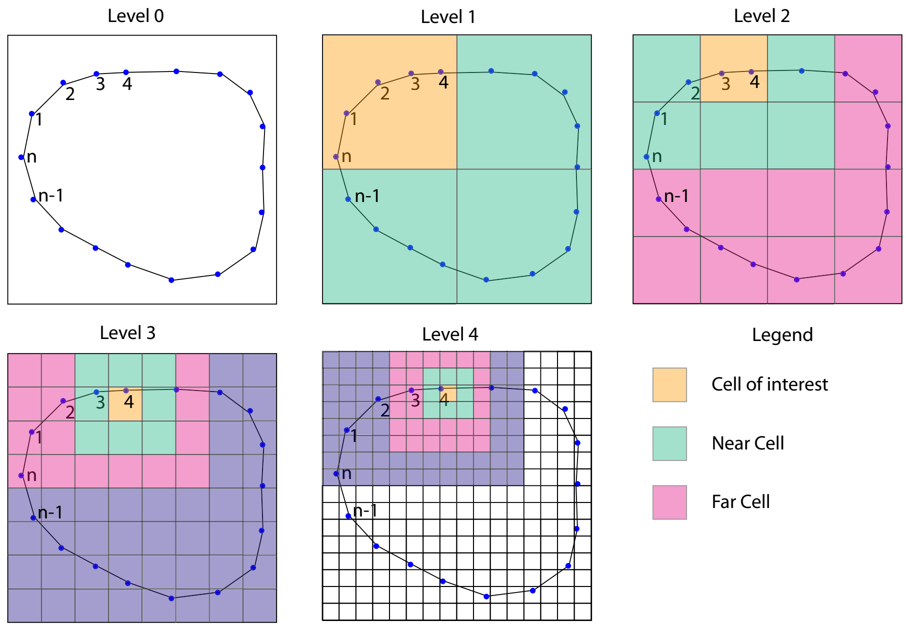


Figure 7.8: Computational cell hierarchy constructed at different FMM levels.

7.1.1.4. Solution Method

This option is enabled only when the **Use FMM** is not selected. In the *Regular method* (when FMM is not being used), *Coustyx* builds system matrices and the linear system of equations are solved either by an iterative method, GMRES, or by a direct method, LU decomposition. The user can select either of these solution methods from the drop-down menu.

GMRES This option solves the linear system of equations by the iterative method *Generalized Minimum Residual Method* (GMRES). (Figure 7.10).

Direct This option solves the linear system of equations by LU decomposition.

7.1.1.5. GMRES

The contents of this panel are activated only when GMRES is selected as the **Solution Method**. *Coustyx* uses GMRES as the default iterative solver when **Use FMM** option is selected. The linear system of equations are solved by the iterative method *Generalized Minimum Residual Method* (GMRES). (Figure 7.6).

Preconditioner Type Preconditioners are used in iterative solvers to improve spectral properties of the system matrices for faster convergence. *Coustyx* implements three precondi-

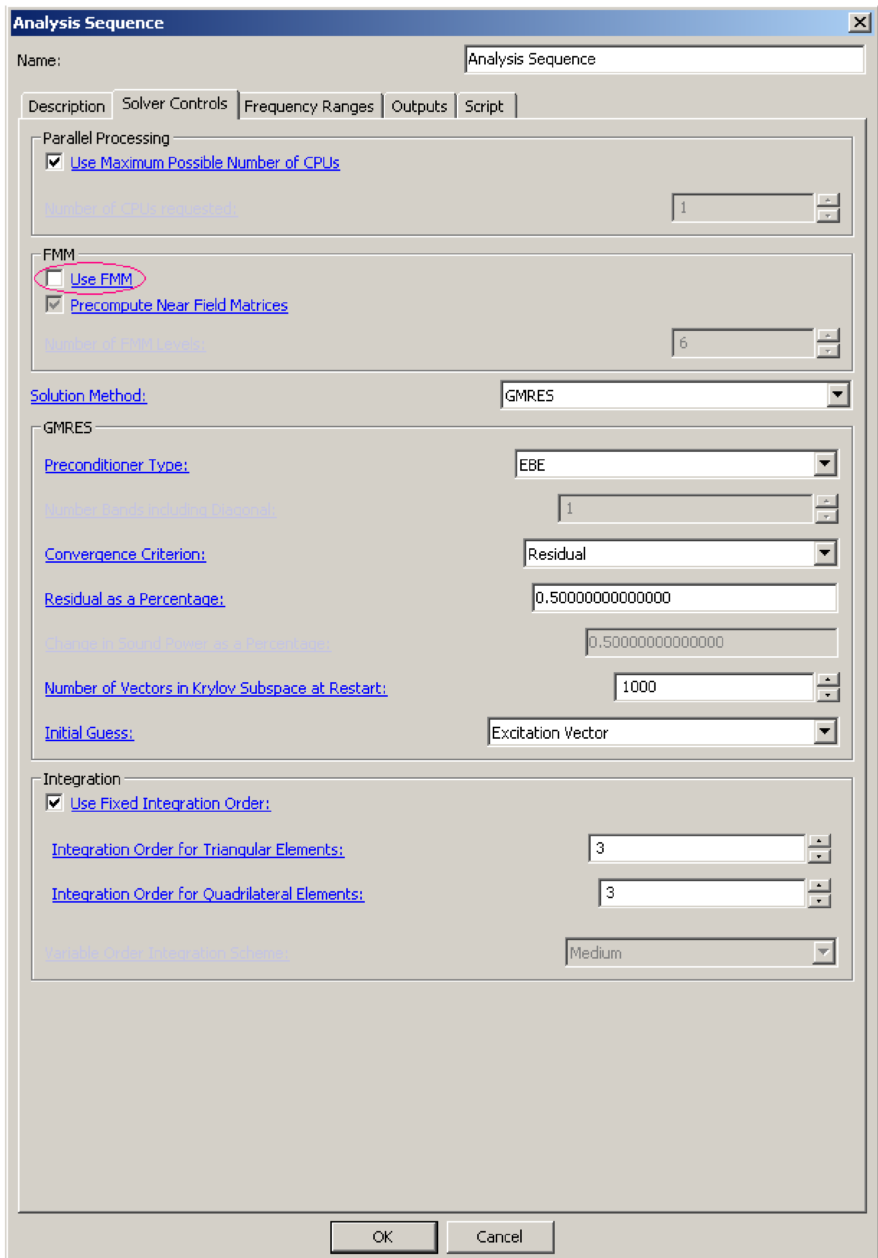


Figure 7.9: Solution Method FMM

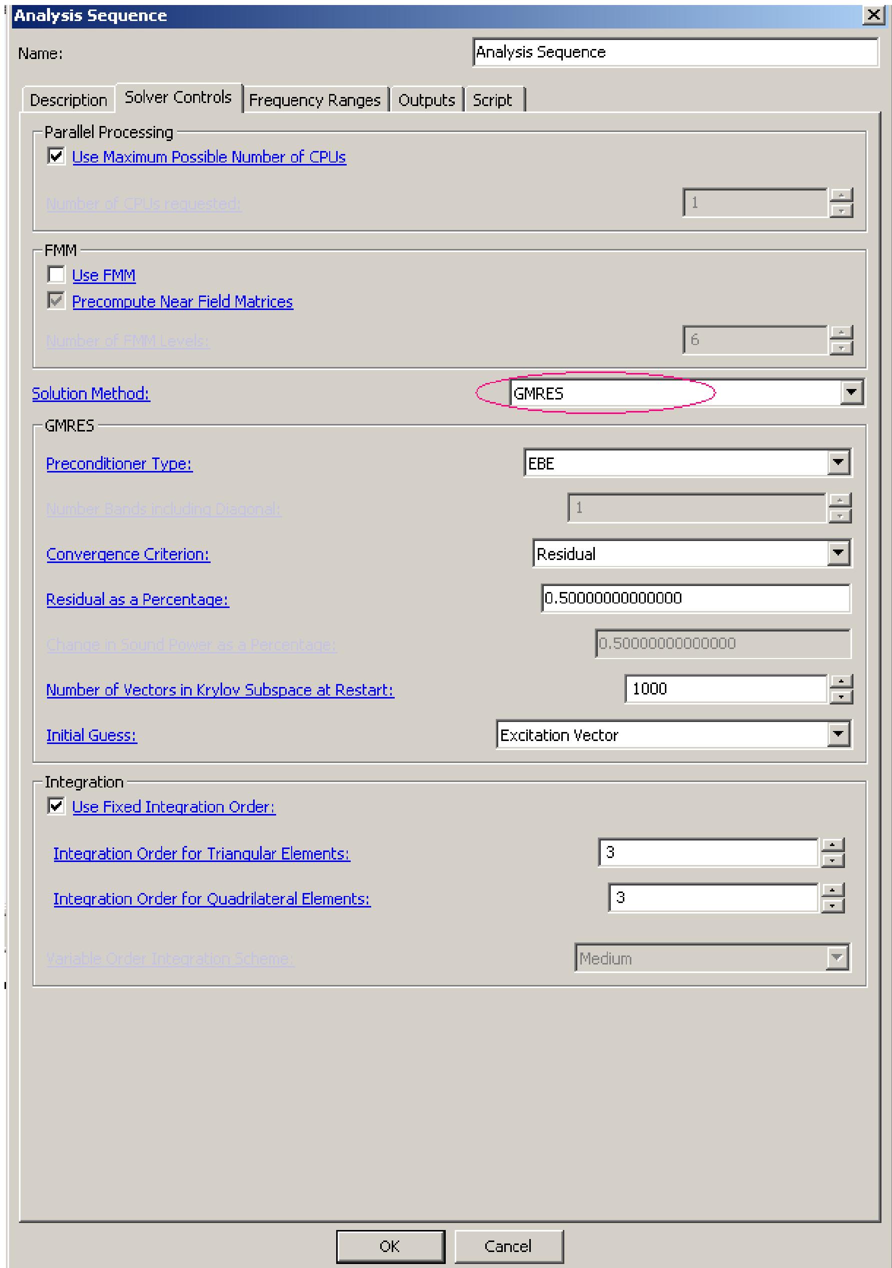


Figure 7.10: GMRES

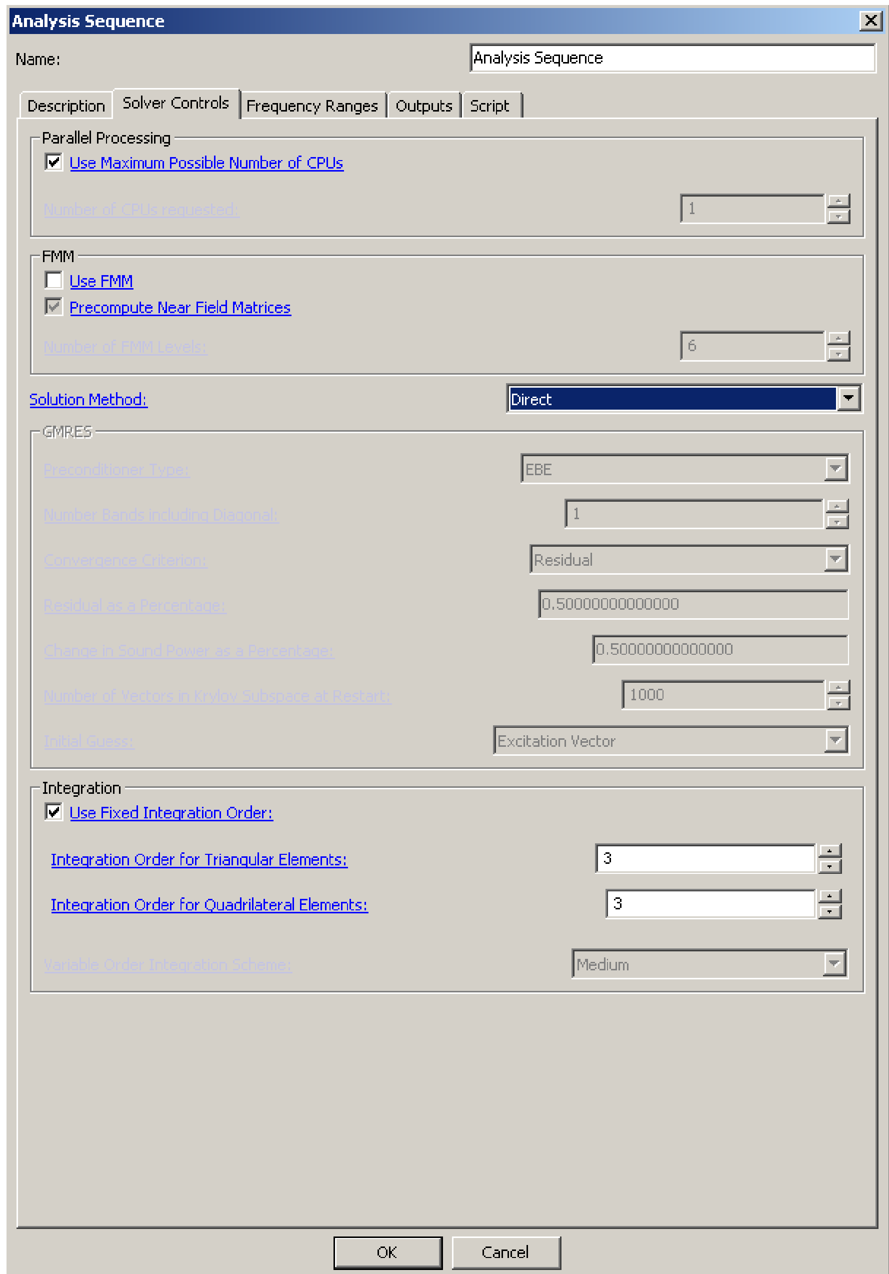


Figure 7.11: Direct Solution Method

tioners which can be selected based on the user requirements for reducing memory usage, faster convergence, parallel-processing, etc. (Figure 7.6).

None This option is selected if the user wants to use GMRES without any preconditioner. Selection of this option reduces the memory usage, as the preconditioner matrix is not created, but results in large number of iterations before the solution converges to the specified tolerance.

EBE This option selects the *Element-By-Element* (**EBE**) preconditioner to be used in GMRES. The preconditioner matrix is represented as a product of element matrices. This helps to store individual element self-influence matrices in *unassembled* form, thus reducing the memory usage to $O(N)$ (from $O(N^2)$ for *Near field* preconditioner), where N is the number of unknowns. The convergence rate (that is, the number of iterations taken by GMRES to converge to a specified tolerance) is in between that of **Near field** and **Diagonal** preconditioners.

Near Field This option selects the *Near field* preconditioner to be used in GMRES. The preconditioner is assembled from the element self-influence matrices. By far, this is the best preconditioner known for its fast convergence rate. However, it's memory usage increases scales as $O(N^2)$ and rapidly with the number of unknowns (N). This method is suggested if the number of unknowns are small.If memory usage is not an issue, the user is strongly recommended to use this option for all problems. For a large number of unknowns, use the EBE method.

Diagonal This option selects the *Diagonal* preconditioner where only the terms in certain number of bands along the main diagonal of the preconditioner matrix are non-zero. The non-zero terms are assembled from the element self-influence matrices. The convergence rate is the worst among all the preconditioners. However, the time taken for each iteration is less than other preconditioners. The number of bands to be considered while assembling *Diagonal* preconditioner are specified by the option **Number Bands including Diagonal**

Number Bands including Diagonal This option is enabled only when **Diagonal** preconditioner is selected. The preconditioner is assembled upto the number of bands specified from the element self influence matrices. (Figure 7.6)

Convergence Criterion GMRES breaks out of the iteration when the error in the solution becomes less than the specified tolerance. *Coustyx* offers the following two choices for the convergence criterion. (Figure 7.6)

Residual This option computes the residue from the solution vector at each iteration. It is the most commonly used convergence criterion. (Figure 7.6)

Sound Power This option computes the error between sound powers from two consecutive iterations and verify if that is less than the specified tolerance. The sound power convergence criterion is useful in cases where the sound power converges faster than the field solution. (Figure 7.6)

Residual as a Percentage This option is enabled only when **Residual** option is selected as the **Convergence Criterion**. (Figure 7.6). For a linear system of equations $Ax = b$, the residue (R) as a percentage is given by,

$$R = \frac{\|b - Ax\|}{\|b\|} \times 100$$

GMRES keeps running until the percentage error in residue is less than the specified value. The smaller the chosen value the better the accuracy of the final solution will be. But this increases the analysis run time. The selection of this value is based on the user's requirement for the accuracy of the solution to the requirement in speed of the analysis. The final solution will have an error in percentage less than the value specified.

Change in Sound Power as a Percentage This option is enabled only when **Sound Power** option is selected as the **Convergence Criterion**. (Figure 7.6). The change in sound power as a percentage is used as the specified tolerance for the convergence of GMRES.

Number of Vectors in Krylov Subspace at Restart GMRES approximates the solution by minimizing the residue in an orthonormal basis spanned by vectors in Krylov subspace. For every iteration a new vector is added to the Krylov subspace. To control storage requirements in GMRES the maximum number of vectors in Krylov subspace are fixed with this option. When GMRES iterations reach this number, all the vectors in Krylov subspace are cleared and a new GMRES cycle is restarted using the latest iterate as the initial guess. The choice of the maximum number of these vectors is critical in implementing GMRES efficiently. The more the number of vectors in Krylov subspace, the better are the chances of finding the solution, that means faster convergence. However, a value that is larger than necessary involves excessive work and storage. And a value smaller than necessary may lead GMRES to converge slowly or even fail to converge. (Figure 7.6).

Initial Guess GMRES requires an initial guess to start the solver at each frequency. The choice of initial guess could determine the rate of convergence.

Previous Solution This option sets the solution from the previous frequency, while running a frequency sweep, as the initial guess to GMRES. For the first frequency it uses the excitation vector. This might result in faster convergence when adjacent frequencies have similar solutions.

Excitation Vector This option selects the excitation vector (b from system of equations $Ax = b$) as the initial guess to GMRES.

.

7.1.1.6. Integration

In *Coustyx* regular Gauss quadrature method is used to numerically evaluate element matrices. The number of quadrature points are represented by integration order. An integration order corresponds to the highest degree of the polynomial that can be integrated over an element with zero error. The user can select the integration order for triangle and quadrilateral elements separately. (Figure 7.6).

Table 7.1: List of valid integration orders and the corresponding quadrature points on triangle elements.

Order	No. of points
0	1
2	3
4	6
5	7
6	12
7	13
9	21
13	36
18	66
20	78
21	91
23	105

Table 7.2: List of the integration orders and the corresponding quadrature points on quadrilateral elements. Note, the total number of quadrature points are npts×npts

Order	No. of Points
n (even)	n/2+1
n (odd)	(n+1)/2

Use Fixed Integration Order This option is to set fixed number of quadrature points to be used for all elements. For a FMM case this option is selected by default. The user can't un-select this option when **Use FMM** is already selected.(Figure 7.6).

Integration Order of Triangular Elements This option is active only when **Use Fixed Integration Order** is selected. The user can select the order to which element integrals are evaluated on triangular elements. Table 7.1 shows the list of valid integration orders and the corresponding quadrature points on triangle elements. If the user selects an integration order that is not in the Table 7.1, *Coustyx* automatically considers the quadrature points belonging to the next highest order. (Figure 7.6).

Integration Order of Quadrilateral Elements This option is active only when **Use Fixed Integration Order** is selected. The user can select the order to which element integrals are evaluated on quadrilateral elements. Table 7.2 shows the list of valid integration orders and the corresponding quadrature points on quadrilateral elements. (Figure 7.6).

Variable Order Integration Scheme This option is enabled only when **Use Fixed Integration Order** is not selected. For FMM case this option is not enabled. The presence of Green's function (which has $1/r$ factor) effects the accuracy of the element integrals computed by Gauss quadrature procedure when two elements are very close to each other.

Table 7.3: Variable integration order used for element matrix computations. Note: D is the distance between centroids of the elements and L is the average length of the elements.

	Quadrilateral			Triangle		
	Medium	Fine	Finest	Medium	Fine	Finest
D/L<2	5	9	11	9	9	9
2≤D/L<5	5	7	9	9	9	9
5≤D/L<20	3	5	5	9	9	9
20≥D/L	3	3	3	9	9	9

We need more integration points for evaluating element integrals accurately for very close elements than compared to elements which are further away. Table 7.3 shows quadrature rules which are formulated based on multiple numerical experiments conducted over elements at various distances with variable number of quadrature points and the accuracy of the element matrix computations. For a given accuracy level the number of integration points required increases with the decrease in the ratio D/L, where D is the distance between the centroids of the elements, and L is the average edge length of the two element edges. (Figure 7.12)

Medium This option is selected when medium accuracy of the solution is acceptable. Table 7.3 shows the integration orders used for quadrilateral and triangular elements. For most of the problems this accuracy level is good enough. (Figure 7.12)

Fine This option is selected when high accuracy of the solution is required. Table 7.3 shows the integration orders used for quadrilateral and triangular elements. (Figure 7.12)

Finest This option is selected when very high accuracy of the solution is required. Table 7.3 shows the integration orders used for quadrilateral and triangular elements. (Figure 7.12)

7.1.2. Frequency Ranges

The **Frequency Ranges** window in Figure 7.13 shows a table of frequency ranges. The table contains columns **Start(Hz)**, **Delta(Hz)**, **No.Freqs.**, **End(Hz)**.

- **Start(Hz)** is the start frequency in Hz.
- **Delta(Hz)** is the frequency resolution in Hz.
- **No.Freqs.** is the number of frequencies.
- **End(Hz)** is the end frequency in Hz.

The user can set a frequency range by inputting values in any of the three columns of the table, the fourth value is derived from the other three. Multiple frequency ranges can either be

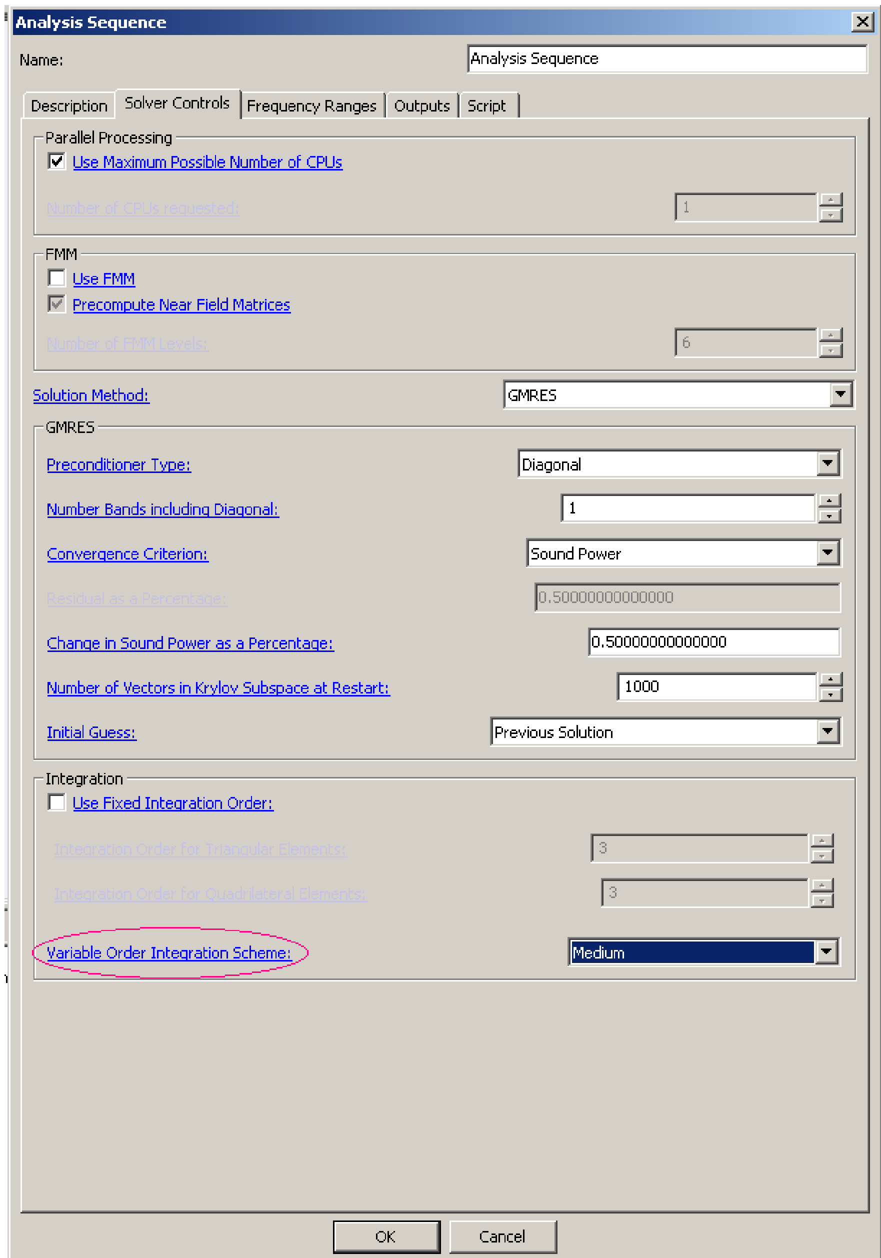


Figure 7.12: Variable Order Integration Scheme

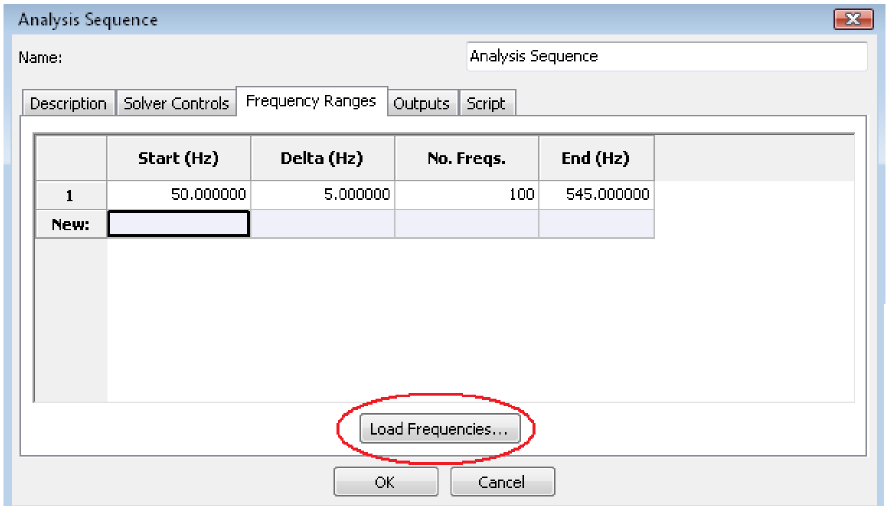


Figure 7.13: Frequency ranges window.

manually added to the table or can be loaded into the table by clicking on **Load Frequencies...** button (Figure 7.13). More details on various options provided to load frequencies are discussed next. Refer to Figure 7.14.

- **Source** is the option type from which frequencies are loaded. Frequencies can be loaded from a File, Structure Freq Response Data, Structure Natural Mode Data, Octave-Band Center Frequencies, 1/3 Octave-Band Center Frequencies, 1/6 Octave-Band Center Frequencies, 1/12 Octave-Band Center Frequencies, and 1/24 Octave-Band Center Frequencies.
- **Whether to replace the table, or append data to it** The data from the **Source** can be used to either replace the current table or append to the existing table with the selection of one of the options: **Replace** or **Append** respectively.

7.1.2.1. File

Analysis frequencies could be loaded from an *ASCII* file using this option (Figure 7.15). The components in the file should be separated by commas, tabs or spaces. Each line in the *ASCII* file represents one frequency range. A line may contain at the most three columns; the first column represents **Start(Hz)** frequency, the second represents **Delta(Hz)**, and the third represents **No.Freqs.**. If a row contains only one column, the value is accepted as the **Start** frequency and the other values are set to the default values, that is, **Delta=0.0** and **No.Freqs=1**.

Click on **Browse...** button to find the desired *ASCII* file. Select the file and click OK. To load frequencies from the file to the Frequency Ranges table press OK. To discard changes press CANCEL.

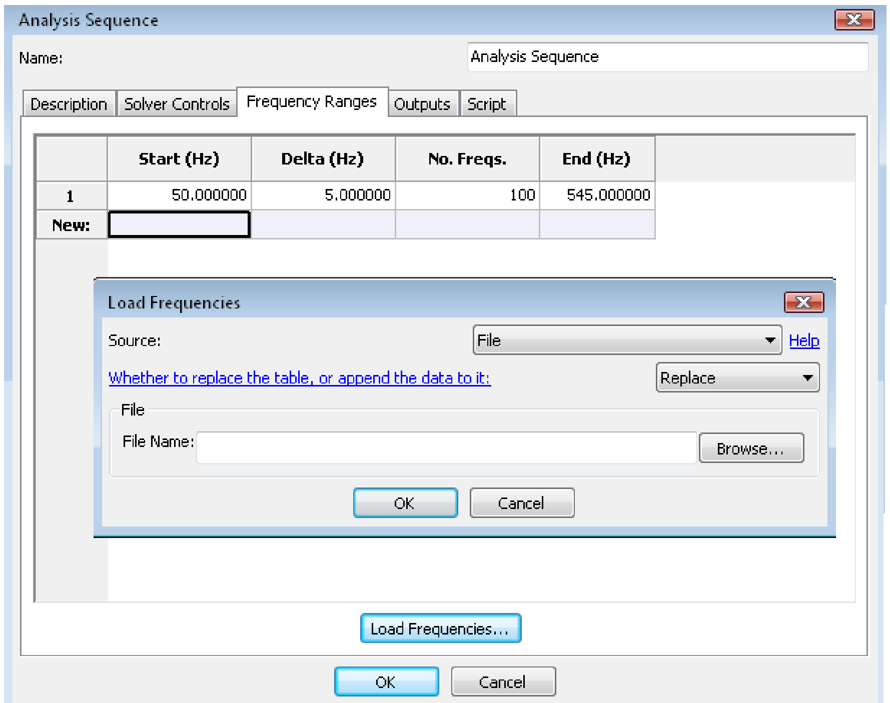


Figure 7.14: Load frequencies window.

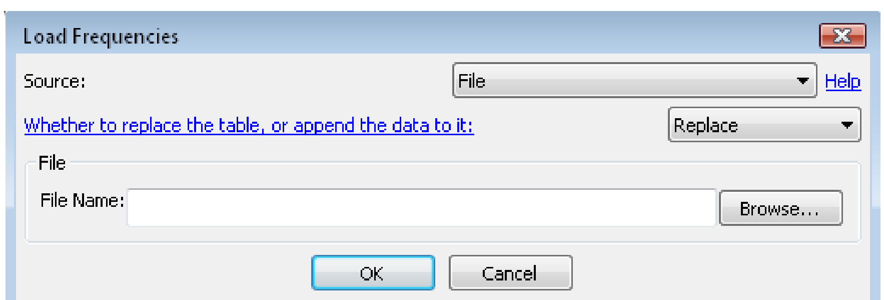


Figure 7.15: Load frequencies from a file window.

- **File Name** is the full path of the *ASCII* file used to load analysis frequencies.

7.1.2.2. Structure Freq Response Data

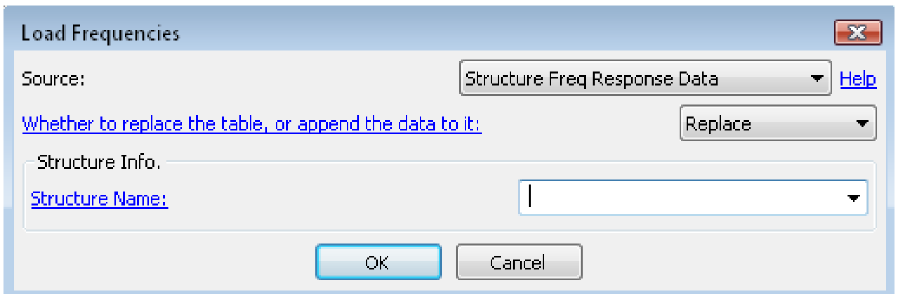


Figure 7.16: Load frequencies from the structure frequency response data.

Analysis frequencies could be extracted from a structure's frequency response data using this option (Figure 7.16). Verify whether the selected structure (defined in **Structure Name**) has frequency response data from: **Model** → **Structures** → <*Structure Mesh Name*> → **Freq Response Data**. See Section 5.1.2 on how to load frequency response data of a structure to the *Coustyx* model. Press OK to load frequencies. A new dialog box with the list of available frequencies from the frequency response data appear. See Figure 5.3. Select the frequencies of interest by checking/unchecking a frequency or by using any of the buttons: **Select All**, **Unselect All**, or **Select by Frequency Range...**. If there is no frequency response data available for the selected structure, an alert window will pop up with the same message.

- **Structure Name** This drop down menu lists names of all structures present in the *Coustyx* model. Select the desired structure from the menu. Analysis frequencies are extracted from this structure's frequency response data.

7.1.2.3. Structure Natural Mode Data

Analysis frequencies could be extracted from a structure's natural mode data using this option (Figure 7.17). Verify whether the selected structure (defined in **Structure Name**) has natural mode data from: **Model** → **Structures** → <*Structure Mesh Name*> → **Natural Mode Data**. See Section 5.1.3 on how to load natural mode data of a structure to the *Coustyx* model. Press OK to load natural frequencies. A new dialog box with a list of available natural frequencies from the natural mode data appear. See Figure 5.5. Select the frequencies of interest by checking/unchecking a natural frequency or by using any of the buttons: **Select All**, **Unselect All**, or **Select by Frequency Range...**. If there is no natural mode data available for the selected structure, an alert window will pop up with the same message.

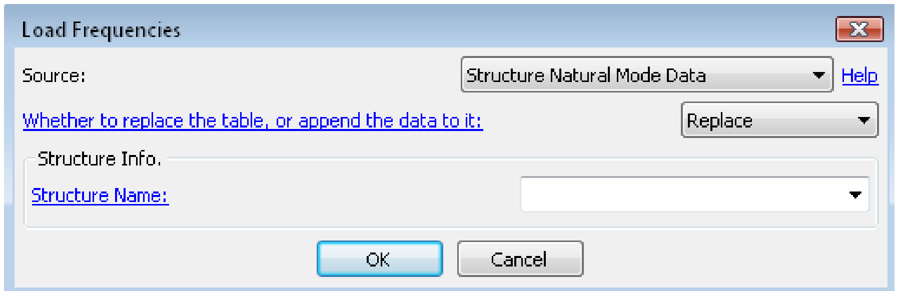


Figure 7.17: Load frequencies from the structure natural mode data.

- **Structure Name** This drop down menu lists names of all structures present in the *Coustyx* model. Select the desired structure from the menu. Analysis frequencies are extracted from this structure's natural mode data.

7.1.2.4. Octave-Band Center Frequencies

The frequency scale for acoustic analysis is usually divided into contiguous *proportional* frequency bands. The partitioning of a bth band with lower cut-off frequency f_1 and upper cut-off frequency f_2 is said to be proportional if f_2/f_1 is the same for each band. The center frequency f_o of such a band is defined as the geometric mean of the upper and lower cut-off frequencies, that is, $f_o = \sqrt{(}f_1 f_2)$ (refer [5]). The center frequencies of successive bands have the same ratio as the upper and lower cut-off frequencies for any band.

In an *octave band*, the ratio between the upper and lower cut-off frequency (or two successive band center frequencies) is 2:1, that is, $f_2/f_1 = 2$. A *1/3-octave band* has $f_2/f_1 = 2^{1/3}$. For any $1/N$th-octave band, the ratio of band limits (or two successive band center frequencies) is given by,

$$\frac{f_2}{f_1} = 2^{1/N}$$

Octave-band,

$$\frac{f_2}{f_1} = 2$$

1/3 octave-band,

$$\frac{f_2}{f_1} = 2^{1/3}$$

1/6 octave-band,

$$\frac{f_2}{f_1} = 2^{1/6}$$

1/12 octave-band,

$$\frac{f_2}{f_1} = 2^{1/12}$$

1/24 octave-band,

$$\frac{f_2}{f_1} = 2^{1/24}$$

Hence, any *proportional* or *octave* band is defined by its center frequency and by N. Analysis done at these center frequencies are assumed to be valid over the entire band width. Refer to Table 7.4 for comparison of center frequencies of various fractional octave-bands between the frequency limits 1000–10000Hz.

Table 7.4: Comparison of preferred center frequencies for fractional octave-bands between 1000–10000 Hz.

1 Octave	1/3 Octave	1/6 Octave	1/12 Octave	1/24 Octave
1000	1000	1000	1000	1000
				1030
			1060	1060
				1090
		1120	1120	1120
				1150
			1180	1180
				1220
	1250	1250	1250	1250
				1280
			1320	1320
				1360
		1400	1400	1400
				1450
			1500	1500
				1550
	1600	1600	1600	1600
				1650
			1700	1700
				1750
		1800	1800	1800
				1850
			1900	1900
				1950
2000	2000	2000	2000	2000
				2060
			2120	2120
				2180
		2240	2240	2240
				2300
			2360	2360
				2430

Table 7.4: (continued)

1 Octave	1/3 Octave	1/6 Octave	1/12 Octave	1/24 Octave
	2500	2500	2500	2500
				2580
			2650	2650
				2720
		2800	2800	2800
				2900
			3000	3000
				3070
	3150	3150	3150	3150
				3250
			3350	3350
				3450
		3550	3550	3550
				3650
			3750	3750
				3870
4000	4000	4000	4000	4000
				4120
			4250	4250
				4370
		4500	4500	4500
				4620
			4750	4750
				4870
	5000	5000	5000	5000
				5150
			5300	5300
				5450
		5600	5600	5600
				5800
			6000	6000
				6150
	6300	6300	6300	6300
				6500
			6700	6700
				6900
		7100	7100	7100
				7300
			7500	7500
				7750
8000	8000	8000	8000	8000

Table 7.4: (continued)

1 Octave	1/3 Octave	1/6 Octave	1/12 Octave	1/24 Octave
				8250
			8500	8500
				8750
		9000	9000	9000
				9250
			9500	9500
				9750
	10000	10000	10000	10000

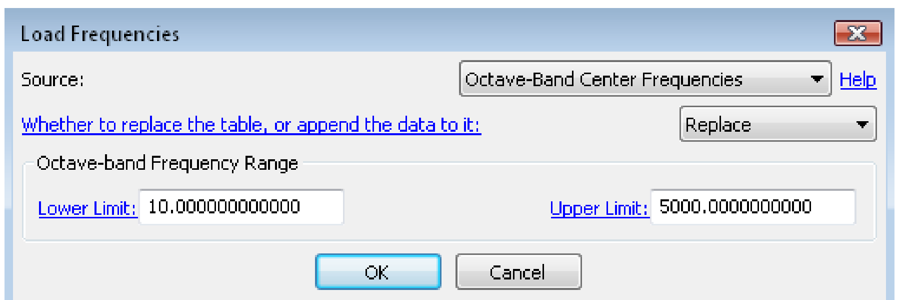


Figure 7.18: Load center frequencies of an octave band.

Steps to be followed to generate octave-band center frequencies:

- Select one of the $1/N$ Octave-Band Center Frequencies from the **Source** drop-down menu. Figure 7.18.
- Select one of the options: **Replace** or **Append** from **Whether to replace the table, or append data to it** drop-down menu.
- Enter the lower and upper frequency limits in text boxes: **Lower Limit** and **Upper Limit** respectively.
- Press OK to generate a list of preferred center frequencies for the selected octave-band within the upper and lower frequency limits. Press CANCEL to discard changes.

Octave-Band frequency range definitions:

- **Lower Limit**. The lower limit for the selection of octave-band center frequencies. The frequency defined here should lie between 10Hz–31500Hz. Center frequencies generated lie within the upper and lower frequency limits.

- **Upper Limit**. The upper limit for the selection of octave-band center frequencies. The frequency defined here should lie between 10Hz–31500Hz. Center frequencies generated lie within the upper and lower frequency limits

7.1.2.5. 1/3 Octave-Band Center Frequencies

The center frequencies of two successive bands in a 1/3 octave-band have a ratio of $2^{1/3}$:1. Refer to Octave-Band Center Frequencies for more details.

7.1.2.6. 1/6 Octave-Band Center Frequencies

The center frequencies of two successive bands in a 1/6 octave-band have a ratio of $2^{1/6}$:1. Refer to Octave-Band Center Frequencies for more details.

7.1.2.7. 1/12 Octave-Band Center Frequencies

The center frequencies of two successive bands in a 1/12 octave-band have a ratio of $2^{1/12}$:1. Refer to Octave-Band Center Frequencies for more details.

7.1.2.8. 1/24 Octave-Band Center Frequencies

The center frequencies of two successive bands in a 1/24 octave-band have a ratio of $2^{1/24}$:1. Refer to Octave-Band Center Frequencies for more details.

7.2. Outputs

The **Outputs** window is shown in Figure 7.19. It consists of options to save the results to a binary file, save pressure and velocity at sensor locations to an *ASCII* file, create an iGlass file for post-processing visualization, and options to compute sound power levels from ISO standards.

7.2.1. Binary Results

7.2.1.1. Create a Binary Results File

This option is selected to save the solution state at the end of each analysis frequency. The binary file could later be read using a *Coustyx* function to recreate the solution state. Figure 7.19.

7.2.2. Sensors

Sensors are field point locations at which pressure and particle velocities are computed.

7.2.2.1. Create an Ascii Sensors File

This option is selected to save the field point pressure and velocities at sensor locations to an *ASCII* file *sensors.dat*. Figure 7.20. The sensor coordinates could be entered manually or imported from an *ASCII* file.

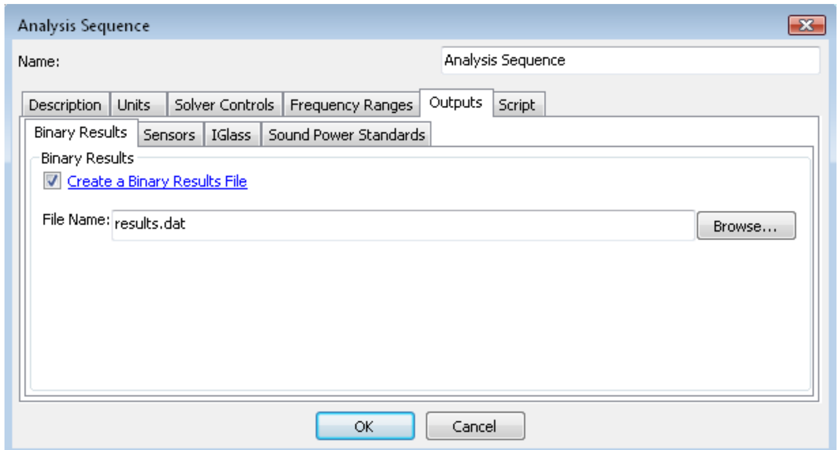


Figure 7.19: Analysis outputs window.

The output data is written to a file entitled *sensors.dat*. You can modify the output file name by typing in a new name or by selecting an existing file through the *Browse* button. Each row in the output file corresponds to the data at all sensors at one frequency. The first column corresponds to the frequency of analysis, the second and third correspond to the real and imaginary parts of the field point pressure at the first sensor, the fourth and fifth contain the real and imaginary parts of the x component of the velocity vector (v_x) at the first sensor, columns six to nine contain velocity components in y (v_y) and z (v_z) directions. The data for n^{th} sensor is written between columns $8(n-1)+2$ and $8(n-1)+9$. Use the matlab utility function "ImportSensorData.m" provided with *Coustyx* to import the data into matlab workspace. You can find this file in the folder \$*InstallDir*\MatlabFiles. To set the folder in the matlab path use: >>addpath("\$*InstallDir*\MatlabFiles") or go to: Matlab → File → Set Path → Add Folder after opening Matlab.

7.2.2.2. Import From File

The sensor coordinates can be imported from an *ASCII* file with components separated by commas, tabs or spaces. The file must contain three columns representing X, Y and Z coordinates of a sensor (Figure 7.21). Each new sensor is added to a new row. Import Options window opens up after the selection of the file with the following options:

Whether to replace the table, or append the data to it The imported data from the file can be used to either replace the current table or append to the existing table by the selection of one of the options: **Replace** or **Append**. (Figure 7.21)

Scale factor All the values in the *ASCII* file are multiplied by the Scale factor before being read into the table (Figure 7.21). This is specifically useful when the imported file has different units compared to *Coustyx* model. A unit conversion factor should be used as the Scale factor to convert these values. For example, when the coordinates in the *ASCII* file are in m and the *Coustyx* units are in mm, a Scale factor of 10^3 is entered to convert the

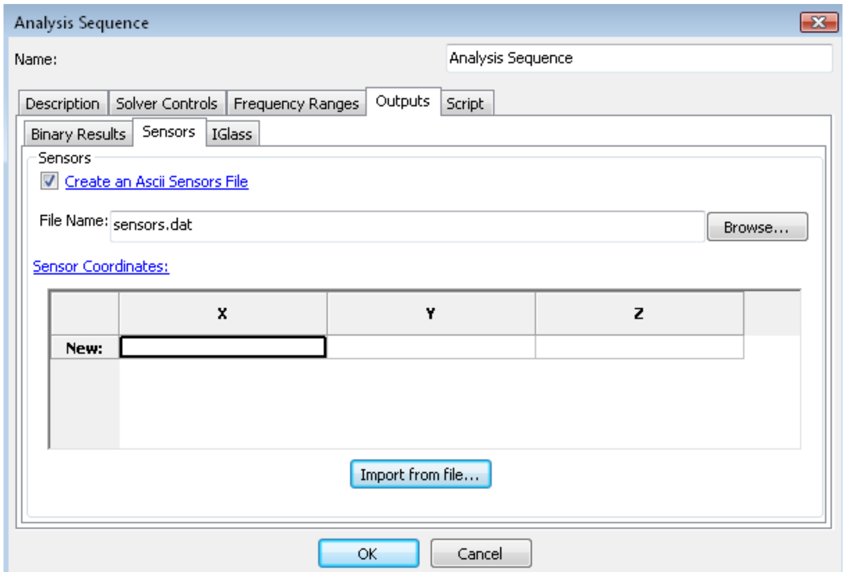


Figure 7.20: Sensor output window.

values in the file from m to mm. A Scale factor of 1 imports the values as they are.

7.2.3. IGlass

IGlass files are data files used to visualize in three dimension, the boundary element solution from *Coustyx*. In addition to the generation of surface potentials data on the boundary element mesh, the user has the option of creating different field point grids to get surface color maps of pressures and particle velocities. Figure 7.22 shows the IGlass outputs window.

7.2.3.1. Create an IGlass File

This option is selected when the user wants to save the output data for IGlass post-processing visualization. The surface potentials of the boundary element mesh are saved to a binary file with extension *.igl. If any field point grids are selected the acoustic variables at these field points are also computed and stored into the file. After the analysis, the user can open the file using the IGlass Viewer.

7.2.3.2. Field Point Grids

Coustyx offers a variety of field point grid options to visualize acoustic field generated by the boundary element solution.

Type There are six different types of field point grids offered by *Coustyx*. They are **Quadrilateral**, **Triangle**, **Annular Disc**, **Box**, **Sphere**, and **Structure Mesh**. The field point

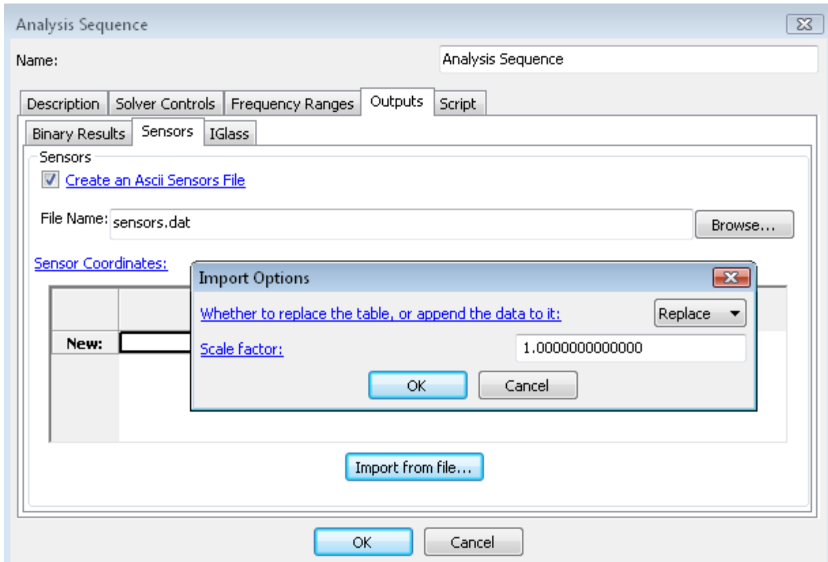


Figure 7.21: Import Sensor Coordinates from a File.

pressure, particle velocity, time-averaged intensity are some of the acoustic variables computed at these points. Figure 7.23 shows some of the field point grid types and the definition of variables required to generate these grids. Field point grids can be added or deleted by pressing the buttons **Add** or **Delete** located at the bottom of the IGlass window (refer to Figure 7.22).

Quadrilateral This option is selected to generate a quadrilateral grid, specified by four corners and the number of divisions in each direction. The coordinates of the four corners need to be entered in a particular order, as shown in Figure 7.23 for quadrilateral grid. (Figure 7.24)

No. of divisions N1 The number of divisions in the direction connecting **Corner 1** to **Corner 2** in a quadrilateral grid (refer to Figure 7.23). The grid defined by the four corners is divided into N1×N2 divisions.

No. of divisions N2 The number of divisions in the direction connecting **Corner 1** to **Corner 3** in a quadrilateral grid (refer to Figure 7.23). The grid defined by the four corners is divided into N1×N2 divisions.

Triangle This option is selected to generate a triangle grid, specified by three corners and equal number of divisions on all sides. The coordinates of the vertices are entered in the table. (Figure 7.25)

No. of divisions N The number of divisions in a triangle grid. Each side of the triangle is divided into equal number of divisions, N.

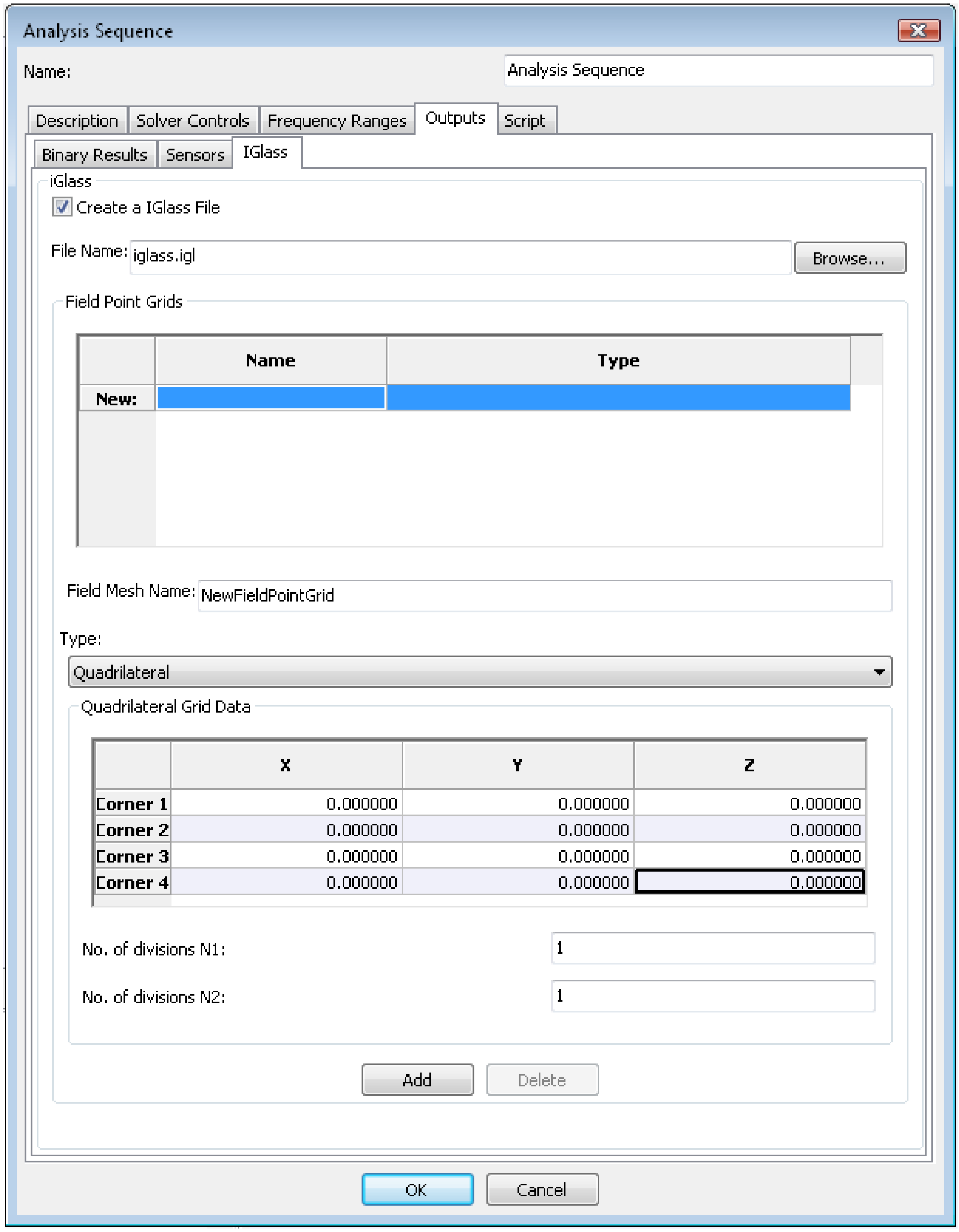


Figure 7.22: IGlass output window.

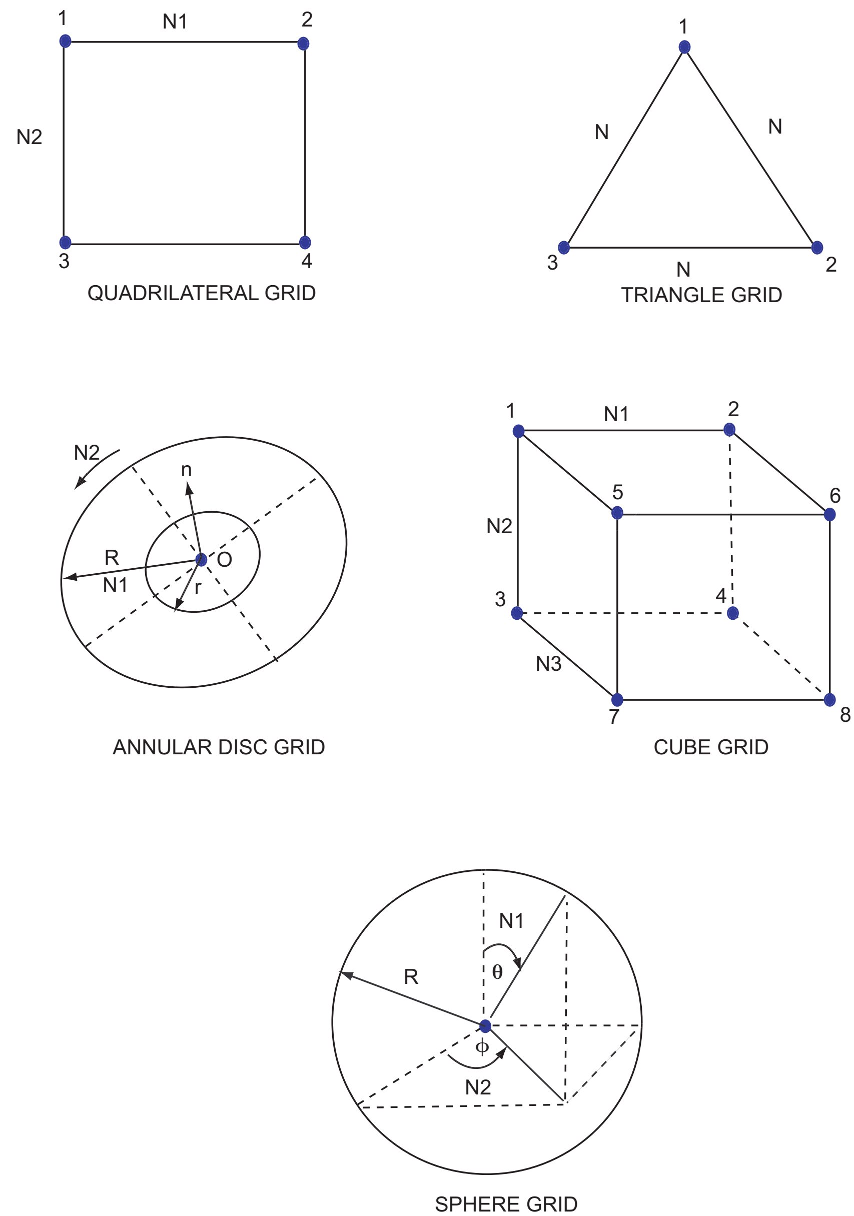


Figure 7.23: IGlass field point grid types - Quadrilateral, Triangle, Annular Disc, Box, and Sphere.

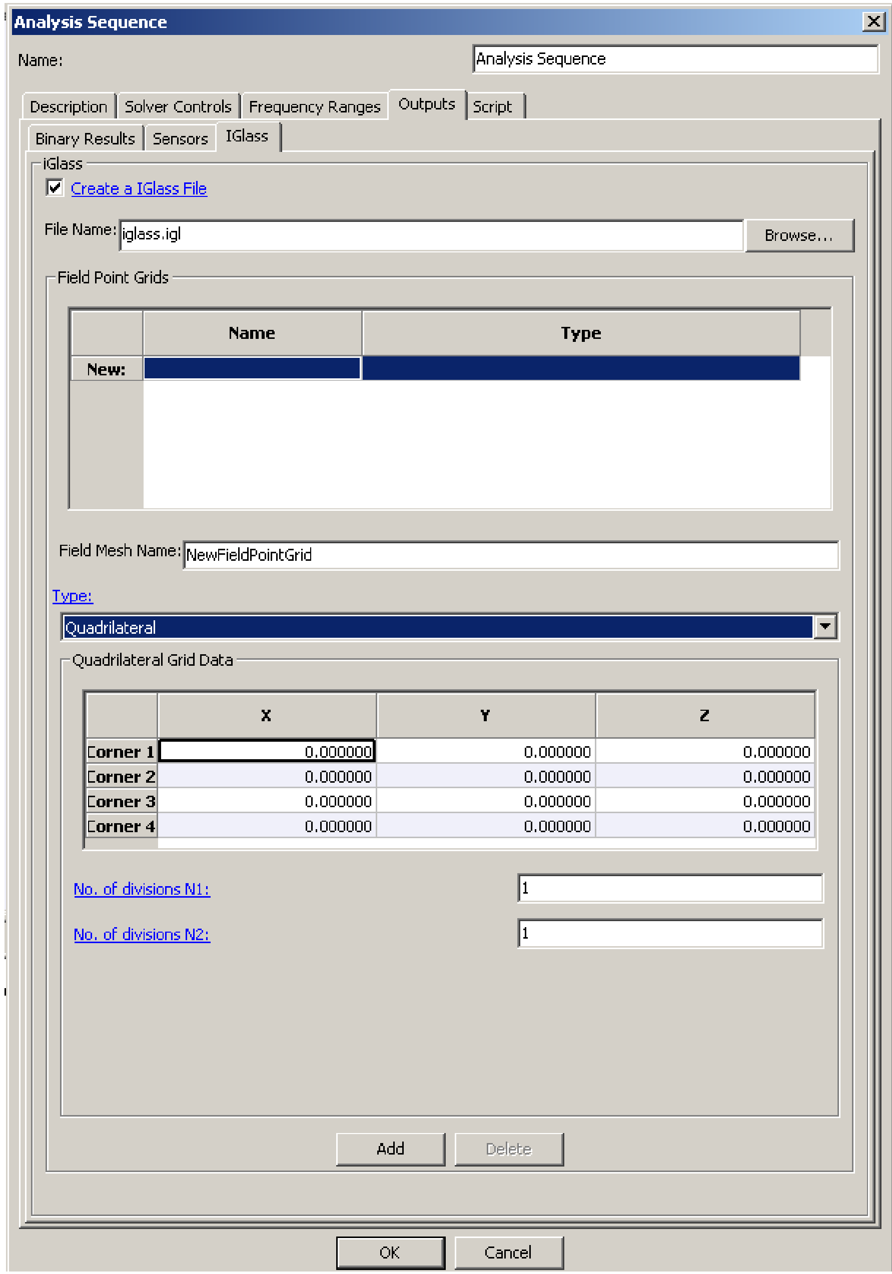


Figure 7.24: Quadrilateral IGlass field point grid types.

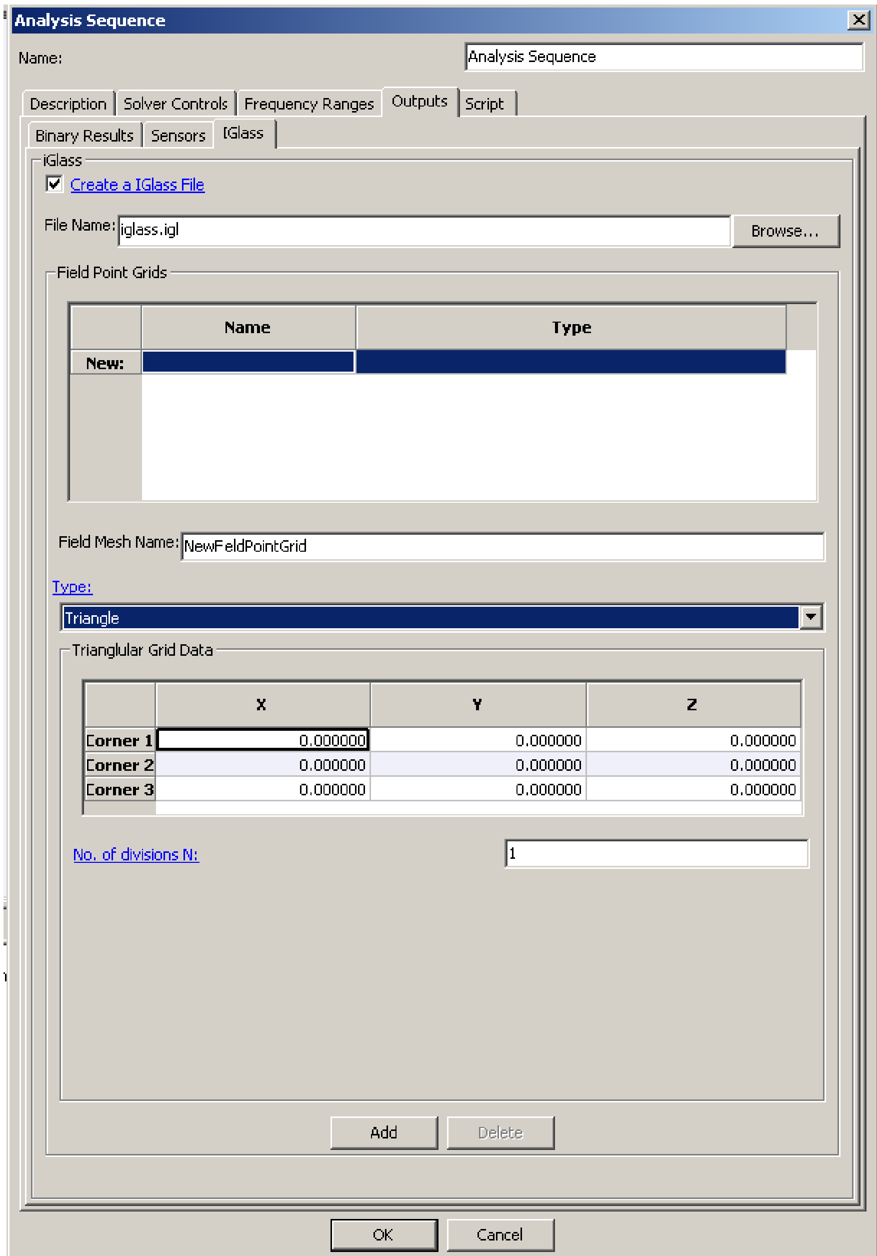


Figure 7.25: IGlass Triangle field point grid types.

Annular Disc This option is selected to generate an annular disc grid, specified by the position of its center, the normal vector to the plane of the disc, the inner and outer radii, and the number of divisions in the radial and angular directions. (Figure 7.26)

- **Center** The coordinates of the center of the annular disc are set using this option. (Figure 7.26)
- **Normal** The normal to the plane of the annular disc is set using this option. (Figure 7.26)
- **Inner Radius** This option sets the inner radius of the annular disc. Any value greater than or equal to zero and less than the outer radius is valid. (Figure 7.26)
- **Outer Radius** This option sets the outer radius of the annular disc. Any value greater than the inner radius is valid. (Figure 7.26)
- **No. of divisions in radial direction** The number of divisions in the radial direction of the annular disc grid (refer to Figure 7.23).
- **No. of divisions in circular direction** The number of divisions in the circular direction of the annular disc grid (refer to Figure 7.23).

Box This option is selected to generate a box grid, specified by eight corners and different number of divisions in each direction. The coordinates of the eight corners need to be entered in the specific order shown in Figure 7.23 for the box grid. (Figure 7.27)

- **No. of divisions N1** The number of divisions in the direction connecting **Corner 1** to **Corner 2** (refer to Figure 7.23). (Figure 7.27) The box grid surface is divided into N1×N2×N3 divisions.
- **No. of divisions N2** The number of divisions in the direction connecting **Corner 1** to **Corner 3** (refer to Figure 7.23). (Figure 7.27) The box grid surface is divided into N1×N2×N3 divisions.
- **No. of divisions N3** The number of divisions in the direction connecting **Corner 1** to **Corner 5** (refer to Figure 7.23). (Figure 7.27) The box grid surface is divided into N1×N2×N3 divisions.

Sphere This option is selected to generate a sphere grid, specified by the position of the center, its radius and the number of divisions in θ and ϕ directions. (Figure 7.28)

- **Center** The coordinates of the center of the sphere are set using this option.
- **Radius** This option sets the radius of the sphere. Any value greater than zero is valid.
- **No. of divisions in zenith angle direction** The number of divisions in θ direction (refer to Figure 7.23).
- **No. of divisions in azimuth angle direction** The number of divisions in ϕ direction (refer to Figure 7.23).

Structure Mesh This option is selected to use existing structure mesh in the model as the field point grid. To use a structure mesh, first Import the mesh into **Structures** model tree member. Refer Section 5.1.1 on how to import a structure mesh into *Coustyx* model. Then, select the structure from the list of structure names provided in the drop-down menu. (Figure 7.29). Note that the iglass data is created only for the visible faces of the structure mesh.

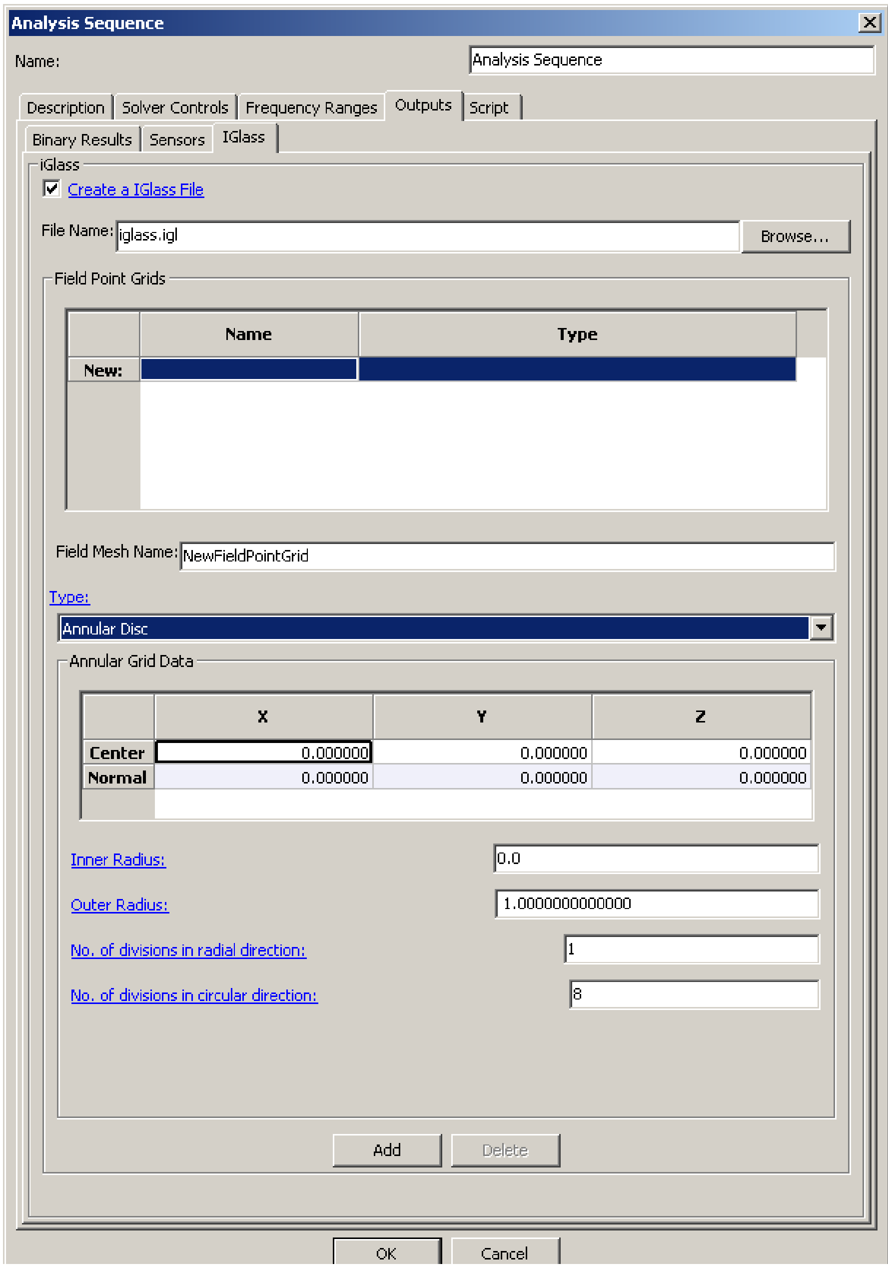


Figure 7.26: IGlass Annular Disc field point grid types.

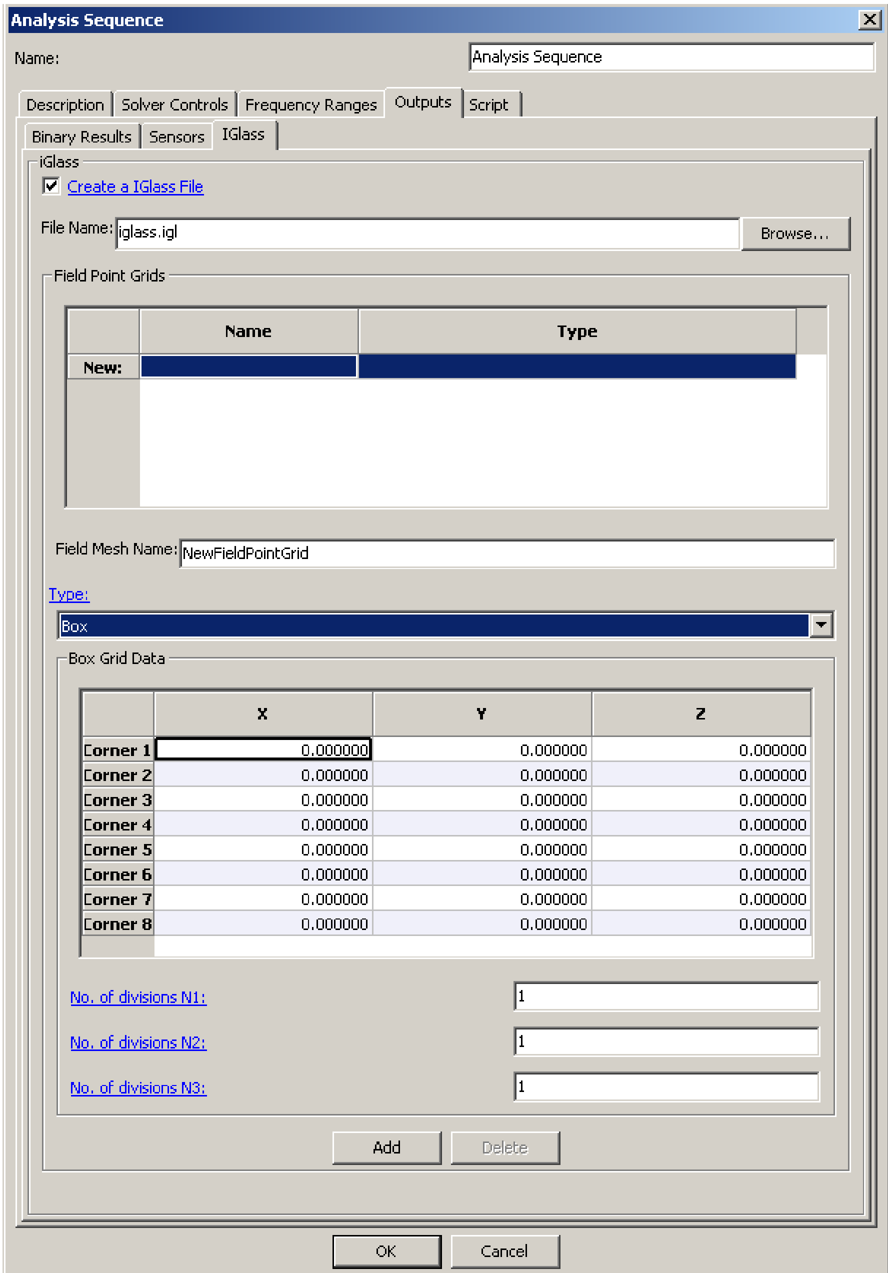


Figure 7.27: IGlass Box field point grid types.

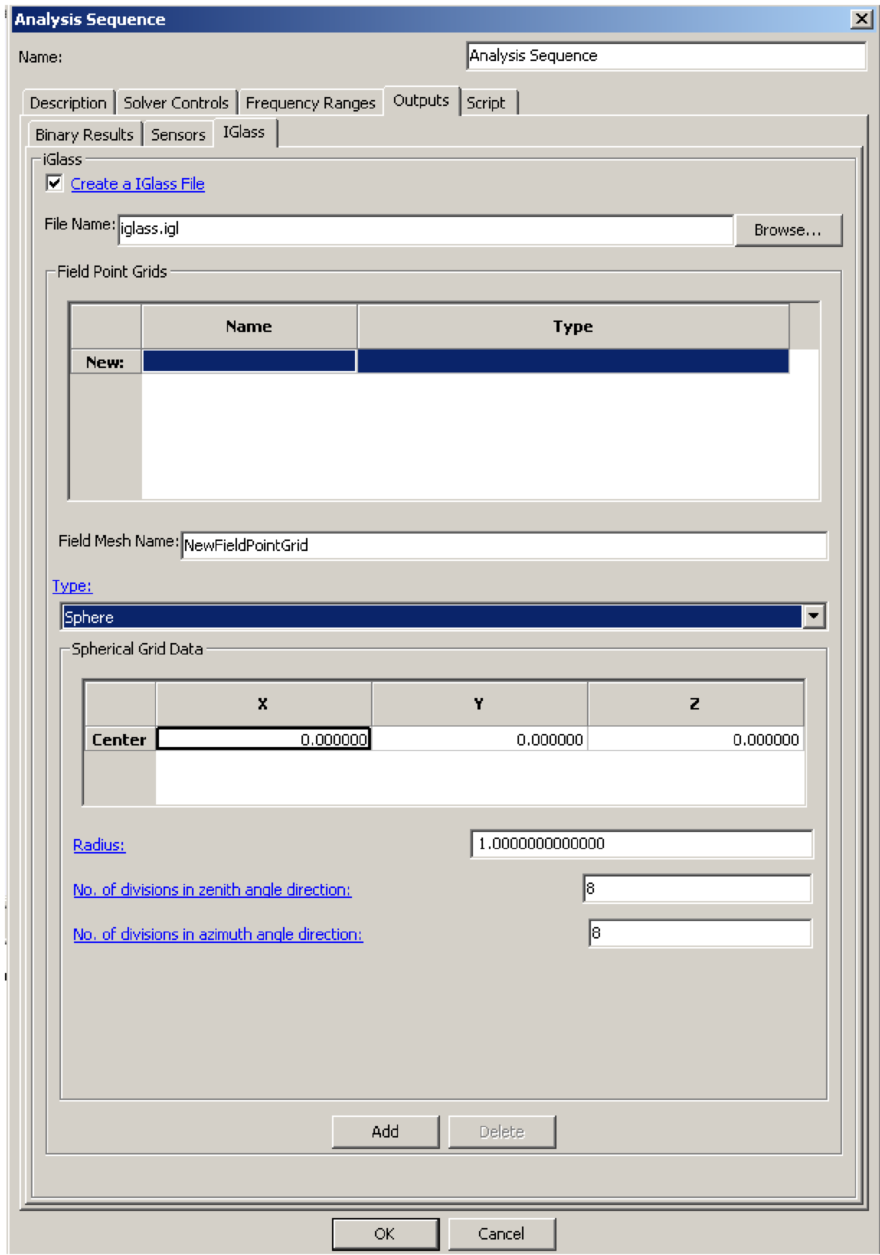


Figure 7.28: IGlass Sphere field point grid types.

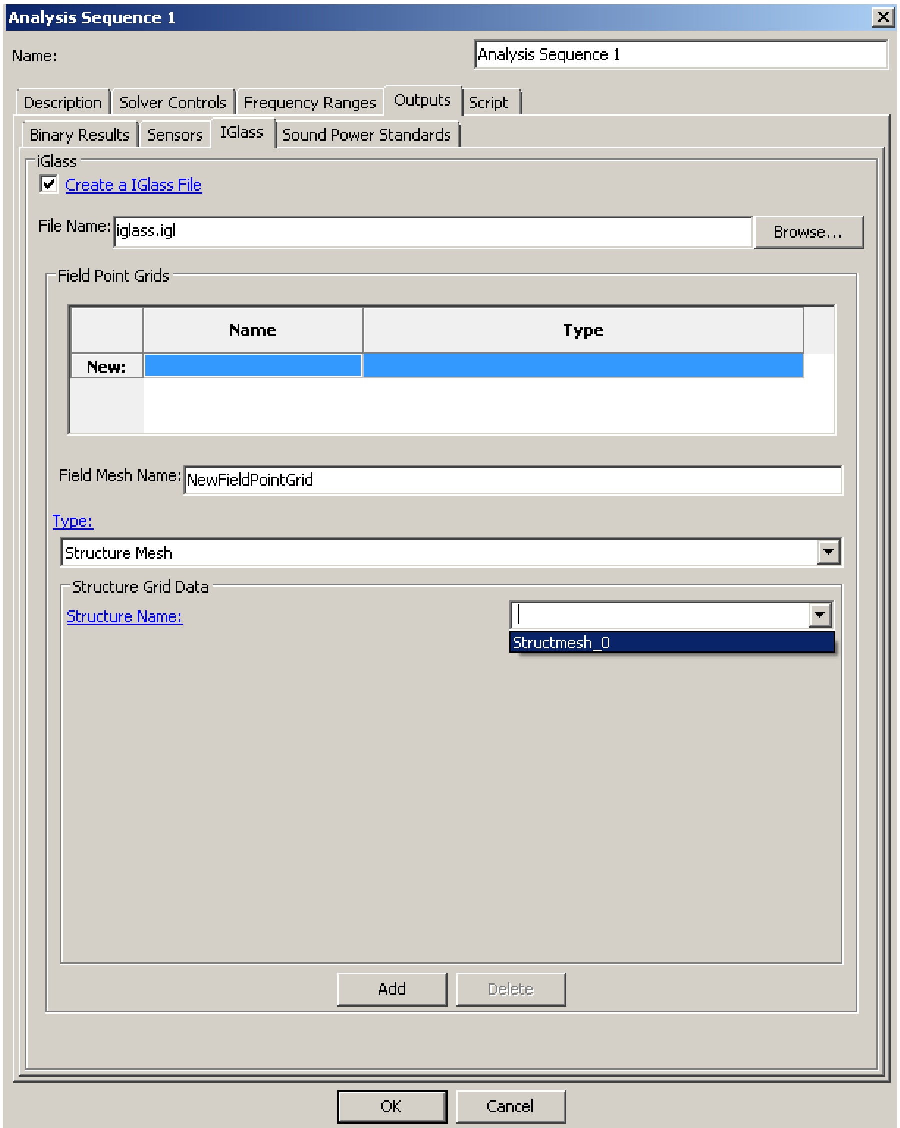


Figure 7.29: IGlass Structure mesh field point grid types.

Structure Name The name of the structure mesh to be used as the field point grid. This list would be populated only if there are structure meshes in the main model tree member **Structures**.

7.2.3.3. IGlass Outputs

- Run acoustic analysis and create an IGlass file.
- Double-click on the IGlass file to open it using IGlass Viewer (Figure 9.48).
- Click on the **Attribs** tab on the top left of the viewer.
- Visualize outputs by selecting any of the acoustic variables listed in the **Attribute** drop-down menu.

Multi-Domain Model For a Multi-Domain model, the **Attribute** drop-down menu lists the following outputs:

Displacement This attribute displays the acoustic particle displacement vector over the surface of the boundary element mesh and at field point grids.

Surface Normal This attribute displays the surface normals on the boundary element mesh and field point grids. Even though this attribute is not an acoustic property it is listed to clarify the normal vectors used to compute **Normal Velocity** and **Normal Sound Intensity**.

For the boundary element mesh in Multi-Domain model, the surface normal is in the direction of the *Domain Normal*. Note that the *Domain Normal* always points away from the domain of interest (refer to Section 6.3 for definition).

Surface normals for different types of field point grids provided in Coustyx (Figure 7.23) are defined below. For **Quadrilateral** and **Triangle** grids, the surface normals point into the surface for the grid coordinates defined in the order shown in Figure 7.23. For **Annular Disc** grid, the surface normal is in the direction of the disc normal. For **Cube** grid, the surface normals on all faces of the cube point outward when the grid coordinates are defined in the order shown in Figure 7.23. For **Sphere** grid, the surface normal at any point on the grid points radially away from the center.

Pressure This attribute displays the sound pressure (amplitude with phase) on the surface of the boundary element mesh and at field point grids.

Normal Velocity This attribute displays the component of acoustic particle velocity (amplitude with phase) in the *Surface Normal* direction.

Velocity This attribute displays the acoustic particle velocity vector (amplitude with phase) on the surface of the boundary element mesh and at field point grids.

Sound Intensity This attribute displays the time averaged sound intensity vector on the boundary element mesh and over field point grids.

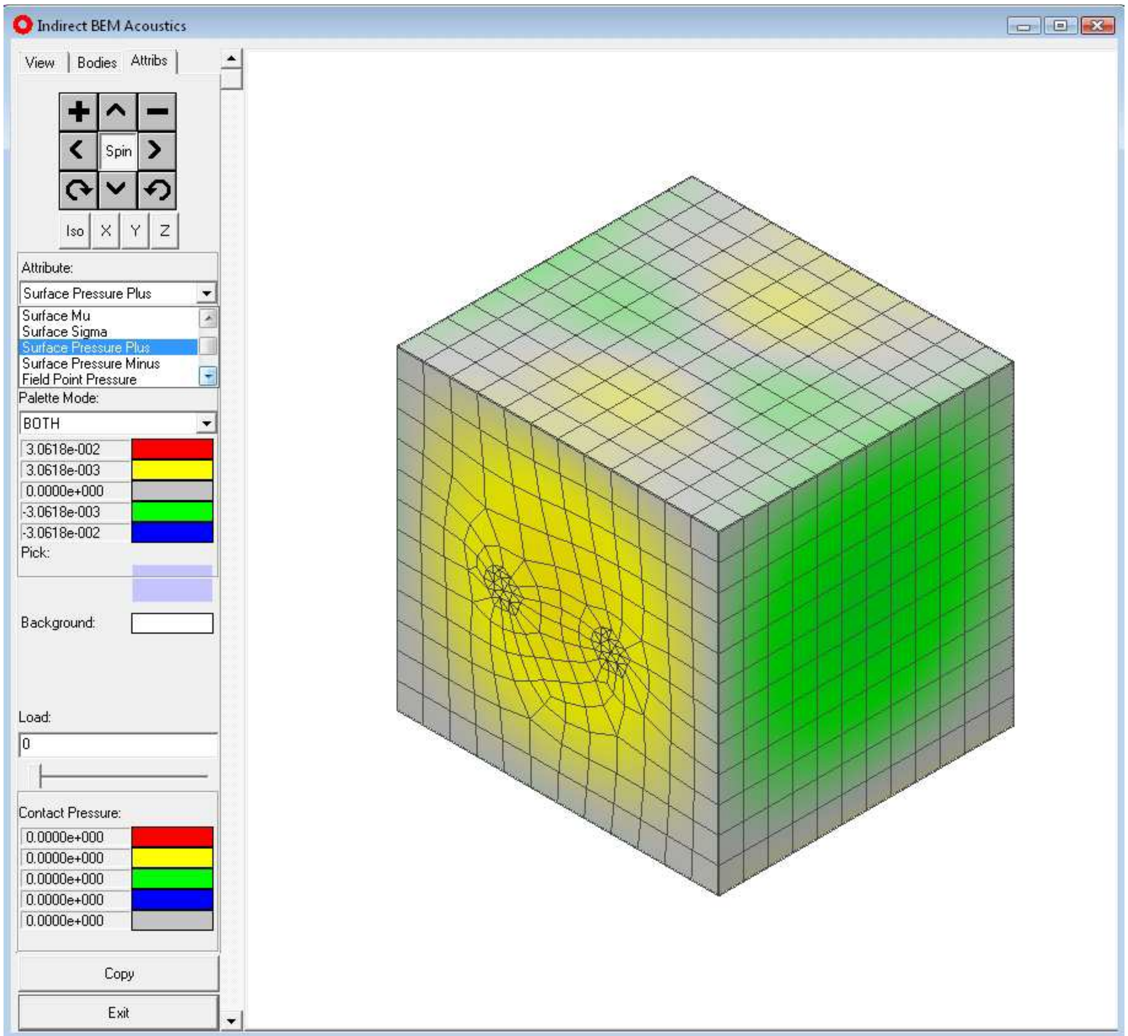


Figure 7.30: IGlass viewer showing sound pressure distribution on the exterior surface of a housing.

Normal Sound Intensity This attribute displays the component of the time averaged sound intensity in the *Surface Normal* direction.

Indirect Model For an Indirect model, the **Attribute** drop-down menu lists the following outputs:

Displacement This attribute displays the acoustic particle displacement vector at field point grids. The acoustic particle displacements over the boundary element surface is not shown.

Surface Normal This attribute displays the surface normals on the boundary element mesh and field point grids. Even though this attribute is not an acoustic property it is listed to clarify the normal vectors used to compute **Normal Velocity-Plus**, or **Normal Velocity-Minus** and **Normal Sound Intensity-Plus**, or **Normal Sound Intensity-Minus**. Also, the **Plus** and **Minus** values in the Indirect BE solution are defined with respect to the surface normals. The plus side of a boundary element is the side at the leading end of the surface normal and the minus side is the side at the trailing end.

For the boundary element mesh in Indirect model, the surface normal is in the direction of the *Element Normal*. Surface normals for different types of field point grids provided in Coustyx (Figure 7.23) are defined below. For **Quadrilateral** and **Triangle** grids, the surface normals point into the surface for grid coordinates defined in the order shown in Figure 7.23. For **Annular Disc** grid, the surface normal is in the direction of the disc normal. For **Cube** grid, the surface normals on all faces of the cube point outward when the grid coordinates are defined in the order shown in Figure 7.23. For **Sphere** grid, the surface normal at any position on the grid points radially away from the center.

Pressure-Plus This attribute displays the sound pressure (amplitude with phase) on the positive side (or the plus side) of the boundary element mesh and at field point grids. Note that the plus side of a boundary element is the side at the leading end of the element normal (refer to Figure 6.16). For field point grids, there is no distinction between the positive side (plus side) and the negative side (minus side). Hence same values are displayed for Pressure-Plus and Pressure-Minus.

Pressure-Minus This attribute displays the sound pressure (amplitude with phase) on the negative side (or the minus side) of the boundary element mesh and at field point grids. Note that the minus side of a boundary element is the side at the trailing end of the element normal (refer to Figure 6.16). For field point grids, there is no distinction between the positive side (plus side) and the negative side (minus side). Hence same values are displayed for Pressure-Plus and Pressure-Minus.

Normal Velocity-Plus This attribute displays the normal component of the acoustic particle velocity (amplitude with phase) on the positive side (or the plus side) of the boundary element mesh surface and at field point meshes. For field point grids there is no distinction between the positive side (plus side) and the negative side (minus side). Hence same values are displayed for Normal Velocity-Plus and Normal Velocity-Minus.

Normal Velocity-Minus This attribute displays the normal component of the acoustic particle velocity (amplitude with phase) on the negative side (or the minus side) of the boundary element mesh surface and at field point meshes. For field point grids there is no distinction between the positive side (plus side) and the negative side (minus side). Hence same values are displayed for Normal Velocity-Plus and Normal Velocity-Minus.

Velocity-Plus This attribute displays the acoustic particle velocity vector (amplitude with phase) on the positive side (or the plus side) of the boundary element mesh surface and at field point grids. For field point meshes, there is no distinction between the positive side (plus side) and the negative side (minus side). Hence same values are displayed for Velocity-Plus and Velocity-Minus.

Velocity-Minus This attribute displays the acoustic particle velocity vector (amplitude with phase) on the negative side (or the minus side) of the boundary element mesh surface and at field point grids. For field point meshes, there is no distinction between the positive side (plus side) and the negative side (minus side). Hence same values are displayed for Velocity-Plus and Velocity-Minus.

Sound Intensity-Plus This attribute displays the time averaged sound intensity vector over the positive side (or the plus side) of the boundary element mesh and at field point grids. For field point grids, there is no distinction between the positive side (plus side) and the negative side (minus side). Hence same values are displayed for Sound Intensity-Plus and Sound Intensity-Minus.

Sound Intensity-Minus This attribute displays the time averaged sound intensity vector over the negative side (or the minus side) of the boundary element mesh and at field point grids. For field point grids, there is no distinction between the positive side (plus side) and the negative side (minus side). Hence same values are displayed for Sound Intensity-Plus and Sound Intensity-Minus.

Normal Sound Intensity-Plus This attribute displays the normal component of the time averaged sound intensity over the positive side (or the plus side) of the boundary element mesh and at field point grids. For field point meshes there is no distinction between the positive side (plus side) and the negative side (minus side). Hence same values are displayed for Normal Sound Intensity-Plus and Normal Sound Intensity-Minus.

Normal Sound Intensity-Minus This attribute displays the normal component of the time averaged sound intensity over the negative side (or the minus side) of the boundary element mesh and at field point grids. For field point meshes there is no distinction between the positive side (plus side) and the negative side (minus side). Hence same values are displayed for Normal Sound Intensity-Plus and Normal Sound Intensity-Minus.

7.2.4. Sound Power Levels from ISO Standards

Coustyx provides options to compute sound power levels from any of the following standard methods: ISO 3744, ISO 3745 and ISO 9614-1. Figure 7.32 shows the Sound Power Standards outputs window.

7.2.4.1. Create a Sound Power File

This option is selected when the user wants to create an output data file with sound power levels computed from ISO standards. The output file is entitled *soundpower_from_standards* by default. You can modify the file name by typing in a new name or by selecting an existing file through the *Browse* button. The output file consists of two columns: the first column contains the frequency of analysis in hertz, and the next contains the sound power level computed from the chosen standard.

7.2.4.2. Weighting Filters

Weighting filters in acoustics are used to enhance or attenuate measured sound pressure levels based on the spectral content of sound to closely simulate the perceived loudness by human hearing. The sound pressure levels at frequencies at which human ear is less sensitive are weighted less than those at which human ear is more sensitive. Figure 7.31 plots the relative response functions for the following filters: Z-weighting, A-weighting, B-weighting, C-weighting, and D-weighting. See Table 7.5 ([1]).

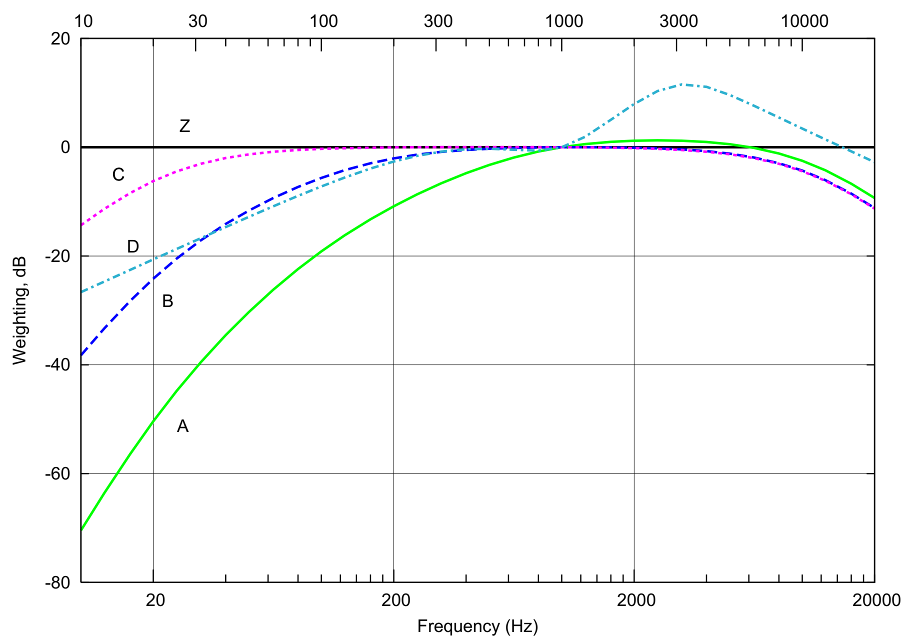


Figure 7.31: Relative response plots for different weighting filters.

Z (Zero) Filter Use this filter to apply zero weight. The measured value is not modified when this filter is used. Figure 7.31.

A Filter Use this filter to attenuate sound at very high and very low frequencies. This filter enhances sound pressure levels in the frequency range between 1kHz to 6kHz, where the human ear is most sensitive. A-weighting is the most commonly used filter. Figure 7.31.

B Filter Use this filter to assess loud noise. Figure 7.31.

C Filter Use this filter to assess loud noise. Figure 7.31.

D Filter This filter is used to assess loud noises in the aircraft industry. Figure 7.31.

7.2.4.3. ISO 3744

ISO 3744 is an international standard that specifies a method to determine sound power levels of noise sources using sound pressure measurements ([2]). The sound pressure levels are measured on a measurement surface enveloping the source. This method is applicable in an essentially free field near one or more reflecting planes. This method produces results that are of engineering grade (or grade 2) accuracy. For precision grade (grade 1) accuracy use methods specified in ISO 3745. Figure 7.32.

Definitions of some commonly used terms in the standard ([2]):

- **Measurement surface**: A hypothetical surface enveloping the noise source, on which measurement points are located. The measurement surface terminates on one or more reflecting planes.

  ISO 3744 allows six different measurement surfaces. They are, **Hemisphere**, **Quadrant**, **Octant**, **Parallelepiped**, **Parallelepiped against wall**, **Parallelepiped against corner**.

- **Reference box**: A hypothetical surface which is the smallest rectangular parallelepiped that just encloses the source and terminates on the reflecting plane or planes.

- **Characteristic source dimension,** d_o. Half the length of the diagonal of the box consisting of the reference box and its images in adjoining reflecting planes.

Hemisphere Use hemisphere measurement surface when there is only one reflecting plane. Figure 7.33 shows a hemisphere measurement surface centered at **Center**. Click on the **Suggest** button to auto fill the measurement surface variables in agreement with the standard. Verify the input you have entered by clicking on the **Check** button. *Coustyx* checks to see if the input variables satisfy the standard requirements. If the standard requirements are not met, a message window pops up to help you make appropriate corrections.

X-Axis Specify the orientation of the X-axis here. See Figure 7.33 for definition.

Y-Axis Specify the orientation of the Y-axis here. See Figure 7.33 for definition.

Center Specify the center of the hemisphere surface here. Select the coordinates such that the center is in the middle of the reference box and its image in the reflecting plane.

Radius The radius of the hemisphere surface shall be equal to or greater than twice the characteristic source dimension, d_o, and not less than $1\,m$.

Table 7.5: Relative response levels for various weightings ([1]).

Nominal Frequency, Hz	A-weighting dB	B-weighting dB	C-weighting dB	D-weighting dB
10	-70.4	-38.2	-14.3	-26.63
12.5	-63.4	-33.2	-11.2	-24.69
16	-56.7	-28.5	-8.5	-22.56
20	-50.5	-24.2	-6.2	-20.63
25	-44.7	-20.4	-4.4	-18.7
31.5	-39.4	-17.1	-3	-16.72
40	-34.6	-14.2	-2	-14.68
50	-30.2	-11.6	-1.3	-12.79
63	-26.2	-9.3	-0.8	-10.87
80	-22.5	-7.4	-0.5	-8.94
100	-19.1	-5.6	-0.3	-7.2
125	-16.1	-4.2	-0.2	-5.57
160	-13.4	-3	-0.1	-3.92
200	-10.9	-2	0	-2.63
250	-8.6	-1.3	0	-1.59
315	-6.6	-0.8	0	-0.81
400	-4.8	-0.5	0	-0.36
500	-3.2	-0.3	0	-0.28
630	-1.9	-0.1	0	-0.46
800	-0.8	0	0	-0.61
1000	0	0	0	0
1250	0.6	0	0	1.92
1600	1	0	-0.1	5.05
2000	1.2	-0.1	-0.2	7.95
2500	1.3	-0.2	-0.3	10.32
3150	1.2	-0.4	-0.5	11.54
4000	1	-0.7	-0.8	11.1
5000	0.5	-1.2	-1.3	9.61
6300	-0.1	-1.9	-2	7.63
8000	-1.1	-2.9	-3	5.46
10000	-2.5	-4.3	-4.4	3.44
12500	-4.3	-6.1	-6.2	1.43
16000	-6.6	-8.4	-8.5	-0.77
20000	-9.3	-11.1	-11.2	-2.74

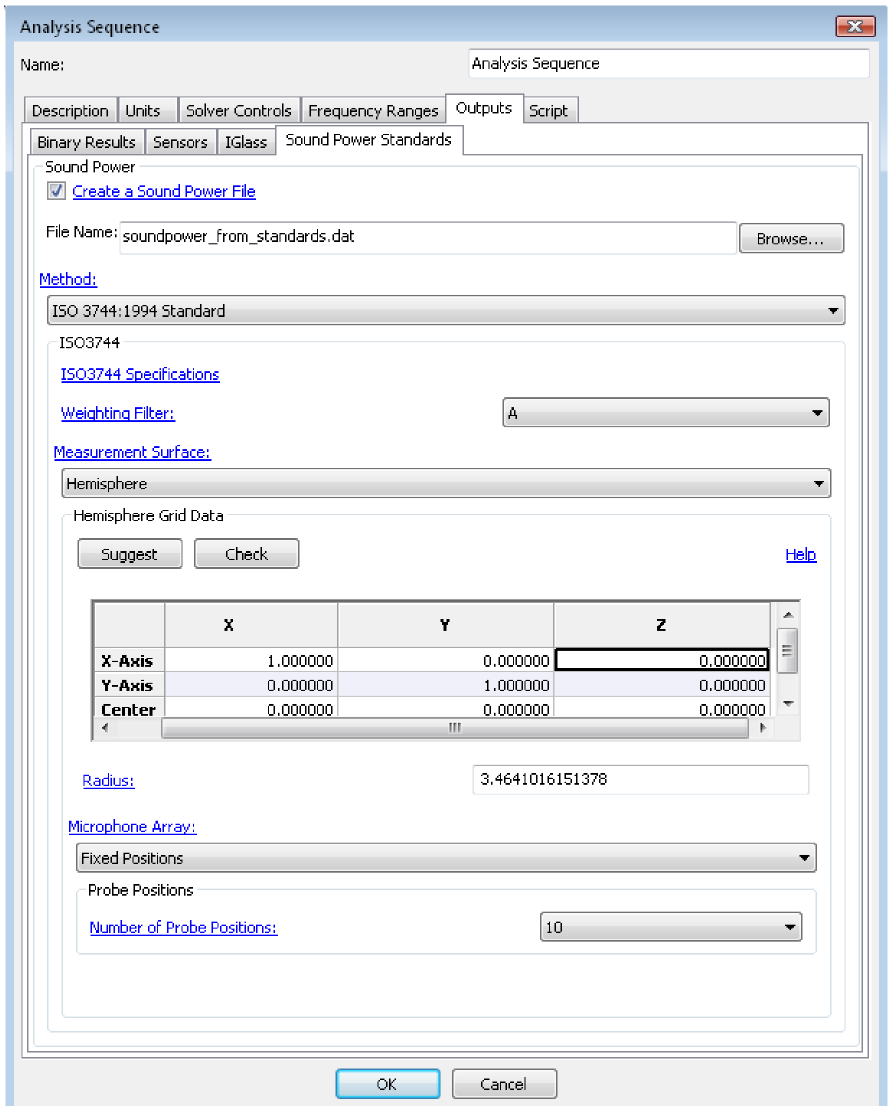


Figure 7.32: ISO3744 sound power standard window.

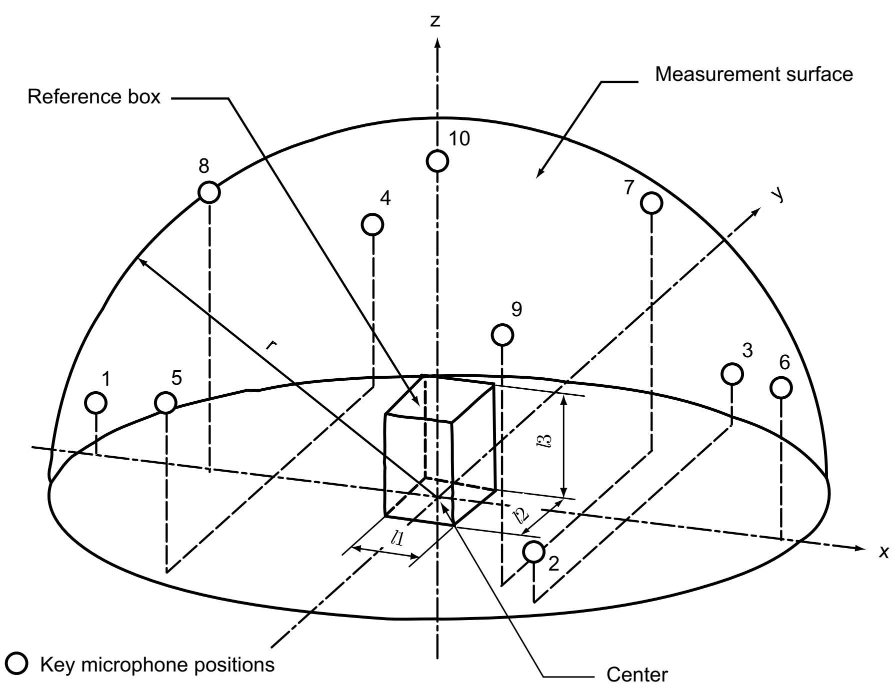


Figure 7.33: ISO3744 - Microphone array on the hemisphere.

Table 7.6: ISO3744 - Coordinates of key microphone positions (1-10) and additional microphone positions (11-20) on a hemisphere ([2]).

Microphone position	$\frac{x}{r}$	$\frac{y}{r}$	$\frac{z}{r}$
1	-0.99	0	0.15
2	0.5	-0.86	0.15
3	0.5	0.86	0.15
4	-0.45	0.77	0.45
5	-0.45	-0.77	0.45
6	0.89	0	0.45
7	0.33	0.57	0.75
8	-0.66	0	0.75
9	0.33	-0.57	0.75
10	0	0	1
11	0.99	0	0.15
12	-0.5	0.86	0.15
13	-0.5	-0.86	0.15
14	0.45	-0.77	0.45
15	0.45	0.77	0.45
16	-0.89	0	0.45
17	-0.33	-0.57	0.75
18	0.66	0	0.75
19	-0.33	0.57	0.75
20(=10)	0	0	1

Microphone Array Choose any of the available types of microphone arrays from the drop-down menu. Figure 7.33 shows a microphone array on the hemisphere.

Fixed Positions The microphone positions for this option are listed in Table 7.6 (Figure 7.33). The key microphone positions are numbered from 1 to 10 and the additional microphones are numbered from 11 to 20. Note that the overhead positions 10 and 20 coincide.

Number of Probe Positions Choose the number of microphones to be 10 or 19 from the drop-down menu. The microphones are associated with equal partial areas on the hemisphere surface, except for the 10^{th} position when 19 microphones are selected. The partial area associated with the 10^{th} position will be twice that of other microphones to account for the omission of 20^{th} position (Table 7.6).

Fixed Positions for Source Emitting Discrete Tones The microphone positions for this option are listed in Table 7.7. This option is recommended over the earlier option when the source emits discrete tones. The presence of microphones at the same height above the reflecting plane can lead to strong interference effects if the source emits discrete tones.

Coaxial Circular Paths Figure 7.34 shows coaxial circular paths in parallel planes

Table 7.7: ISO3744 - Coordinates of microphone positions for sources emitting discrete tones.

Microphone position	$\frac{x}{r}$	$\frac{y}{r}$	$\frac{z}{r}$
1	0.16	-0.96	0.22
2	0.78	-0.6	0.2
3	0.78	0.55	0.31
4	0.16	0.9	0.41
5	-0.83	0.32	0.45
6	-0.83	0.4	0.38
7	-0.26	-0.65	0.71
8	0.74	-0.07	0.67
9	-0.26	0.5	0.83
10	0.1	-0.1	0.99

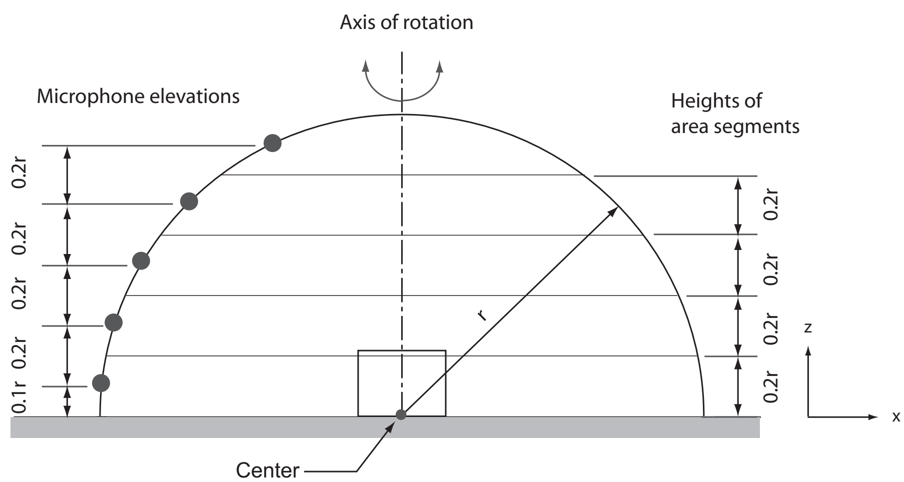


Figure 7.34: Coaxial circular paths in parallel planes for microphone traverses over a reflecting plane.

traversed by a microphone over a reflecting plane. The paths are selected such that the annular area associated with each path is the same.

Number of Circular Paths Traversed by a Probe Enter the number of circular paths traversed by a microphone. The least number is five.

Quadrant Use quadrant measurement surface when the source under test is in front of a wall. Figure 7.35 shows a quadrant measurement surface centered at **Center**. Click on the **Suggest** button to auto fill the measurement surface variables in agreement with the standard. Verify the input you have entered by clicking on the **Check** button. *Coustyx* checks to see if the input variables satisfy the standard requirements. If the standard requirements are not met, a message window pops up to help you make appropriate corrections.

X-Axis Specify the orientation of the X-axis here. See Figure 7.35 for definition.

Y-Axis Specify the orientation of the Y-axis here. See Figure 7.35 for definition.

Center Specify the center of the quadrant surface here. Select the coordinates such that the center is in the middle of the reference box and its images in the reflecting planes.

Radius The radius of the quadrant surface shall be equal to or greater than twice the characteristic source dimension, d_o, and not less than $1\,m$.

Microphone Array Figure 7.35 shows a microphone array on the quadrant surface. The microphone positions are listed in Table 7.6.

Number of Probe Positions Specify the number of microphones associated with equal areas spread over the quadrant surface. Choose the number of microphones to be 5 or 9 from the drop-down menu.

Octant Use octant measurement surface when the source under test is in front of a corner. Figure 7.36 shows a octant measurement surface centered at **Center**. Click on the **Suggest** button to auto fill the measurement surface variables in agreement with the standard. Verify the input you have entered by clicking on the **Check** button. *Coustyx* checks to see if the input variables satisfy the standard requirements. If the standard requirements are not met, a message window pops up to help you make appropriate corrections.

X-Axis Specify the orientation of the X-axis here. See Figure 7.36 for definition.

Y-Axis Specify the orientation of the Y-axis here. See Figure 7.36 for definition.

Center Specify the center of the octant surface here. Select the coordinates such that the center is in the middle of the reference box and its images in the reflecting planes.

Radius The radius of the octant surface shall be equal to or greater than twice the characteristic source dimension, d_o, and not less than $1\,m$.

Microphone Array Figure 7.36 shows a microphone array on the octant surface. The microphone positions are listed in Table 7.6.

Number of Probe Positions Specify the number of microphones associated with equal areas spread over the octant surface. Choose the number of microphones to be 2 or 3 from the drop-down menu.

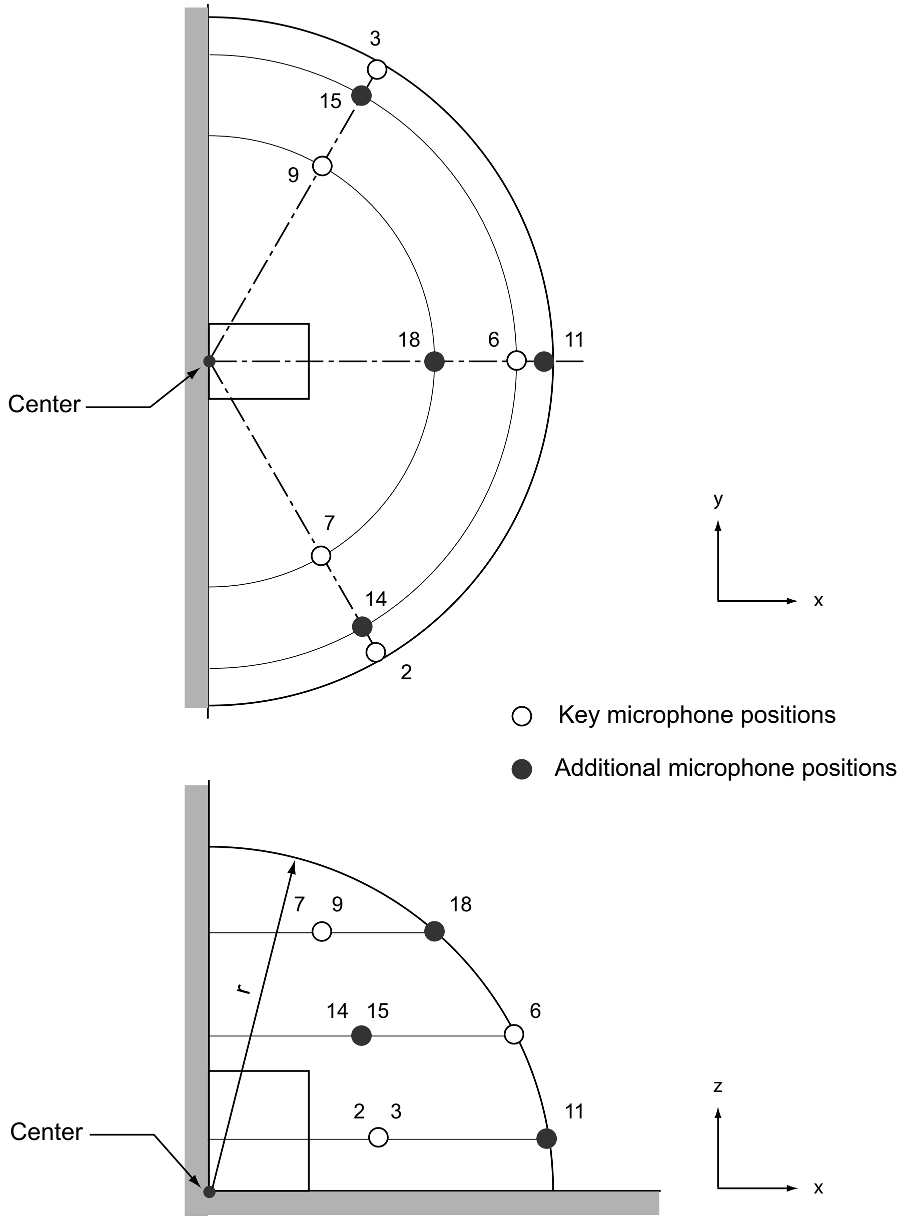


Figure 7.35: ISO3744 - Microphone array on the quadrant.

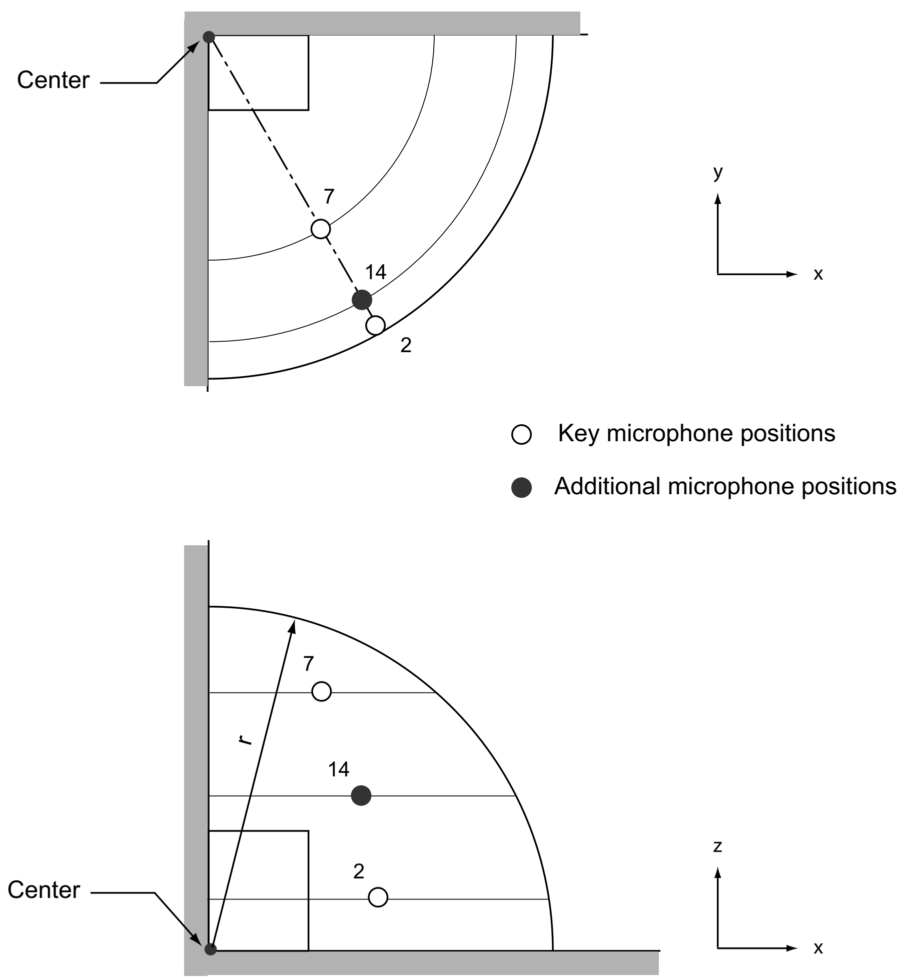


Figure 7.36: ISO3744 - Microphone array on the octant.

Parallelepiped Use parallelepiped measurement surface when there is only one reflecting plane. Figure 7.38 shows a parallelepiped measurement surface whose sides are parallel to those of the reference box. Click on the **Suggest** button to auto fill the measurement surface variables in agreement with the standard. Verify the input you have entered by clicking on the **Check** button. *Coustyx* checks to see if the input variables satisfy the standard requirements. If the standard requirements are not met, a message window pops up to help you make appropriate corrections.

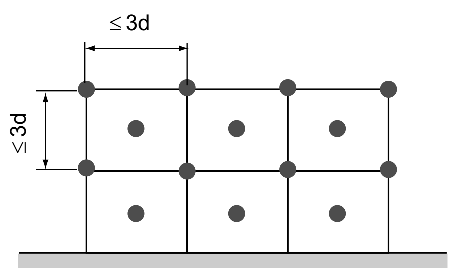


Figure 7.37: ISO3744 - Procedure for fixing microphone positions on a parallelepiped measurement surface.

X-Axis Specify the orientation of the X-axis here. See Figure 7.38 for definition.

Y-Axis Specify the orientation of the Y-axis here. See Figure 7.38 for definition.

Corner1 Specify Corner1 of the parallelepiped surface here. See Figure 7.38 for definition. The parallelepiped is constructed using Corner1 as the starting point and $L1$, $L2$, $L3$ as its dimensions along X, Y, Z-axis respectively.

L1 Length of the parallelepiped in X-direction (Figure 7.38). Specify the value of $L1$ such that it satisfies the definition, $L1 = l1 + 2d$, where $l1$ is the reference box dimension in X-direction, and d is the measurement distance. The recommended value for $d = 1\,m$.

L2 Length of the parallelepiped in Y-direction (Figure 7.38). Specify the value of $L2$ such that it satisfies the definition, $L2 = l2 + 2d$, where $l2$ is the reference box dimension in Y-direction, and d is the measurement distance. The recommended value for $d = 1\,m$.

L3 Length of the parallelepiped in Z-direction (Figure 7.38). Specify the value of $L3$ such that it satisfies the definition, $L3 = l3 + d$, where $l3$ is the reference box dimension in Z-direction, and d is the measurement distance. The recommended value for $d = 1\,m$.

N1 Number of subdivisions in X-direction (Figure 7.38). Select the value of $N1$ such that the length of the rectangular partial area formed by these subdivisions satisfies the criterion: $\frac{L1}{N1} \leq 3d$, where $L1$ is the length of parallelepiped in X-direction, and d is the measurement distance. Refer Figure 7.37. The microphone positions are in the center of each partial area and at each corner of the partial area (excluding the corners intruding into reflecting planes).

N2 Number of subdivisions in Y-direction (Figure 7.38). Select the value of $N2$ such that the length of the rectangular partial area formed by these subdivisions satisfies the

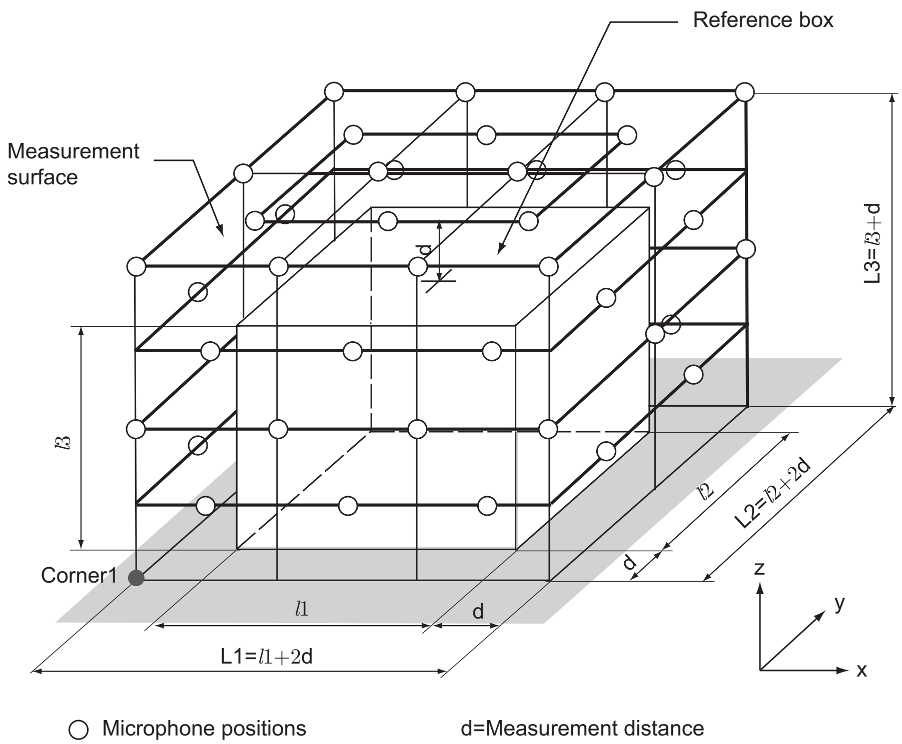


Figure 7.38: ISO3744 - Parallelepiped measurement surface with microphone positions.

criterion: $\frac{L2}{N2} \leq 3d$, where $L2$ is the length of parallelepiped in Y-direction, and d is the measurement distance. Refer Figure 7.37. The microphone positions are in the center of each partial area and at each corner of the partial area (excluding the corners intruding into reflecting planes).

N3 Number of subdivisions in Z-direction (Figure 7.38). Select the value of $N3$ such that the length of the rectangular partial area formed by these subdivisions satisfies the criterion: $\frac{L3}{N3} \leq 3d$, where $L3$ is the length of parallelepiped in Z-direction, and d is the measurement distance. Refer Figure 7.37. The microphone positions are in the center of each partial area and at each corner of the partial area (excluding the corners intruding into reflecting planes).

Parallelepiped against wall Use this measurement surface when the source under test is placed on a floor against a wall. Figure 7.39 shows a parallelepiped measurement surface whose sides are parallel to those of the reference box. Click on the **Suggest** button to auto fill the measurement surface variables in agreement with the standard. Verify the input you have entered by clicking on the **Check** button. *Coustyx* checks to see if the input variables satisfy the standard requirements. If the standard requirements are not met, a message window pops up to help you make appropriate corrections.

X-Axis Specify the orientation of the X-axis here. See Figure 7.39 for definition.

Y-Axis Specify the orientation of the Y-axis here. See Figure 7.39 for definition.

Corner1 Specify Corner1 of the parallelepiped surface here. See Figure 7.39 for definition. The parallelepiped is constructed using Corner1 as the starting point and $L1$, $L2$, $L3$ as its dimensions along X, Y, Z-axis respectively.

L1 Length of the parallelepiped in X-direction (Figure 7.39). Specify the value of $L1$ such that it satisfies the definition, $L1 = l1 + d$, where $l1$ is the reference box dimension in X-direction, and d is the measurement distance. The recommended value for $d = 1\,m$.

L2 Length of the parallelepiped in Y-direction (Figure 7.39). Specify the value of $L2$ such that it satisfies the definition, $L2 = l2 + 2d$, where $l2$ is the reference box dimension in Y-direction, and d is the measurement distance. The recommended value for $d = 1\,m$.

L3 Length of the parallelepiped in Z-direction (Figure 7.39). Specify the value of $L3$ such that it satisfies the definition, $L3 = l3 + d$, where $l3$ is the reference box dimension in Z-direction, and d is the measurement distance. The recommended value for $d = 1\,m$.

N1 Number of subdivisions in X-direction (Figure 7.39). Select the value of $N1$ such that the length of the rectangular partial area formed by these subdivisions satisfies the criterion: $\frac{L1}{N1} \leq 3d$, where $L1$ is the length of parallelepiped in X-direction, and d is the measurement distance. Refer Figure 7.37. The microphone positions are in the center of each partial area and at each corner of the partial area (excluding the corners intruding into reflecting planes).

N2 Number of subdivisions in Y-direction (Figure 7.39). Select the value of $N2$ such that the length of the rectangular partial area formed by these subdivisions satisfies the criterion: $\frac{L2}{N2} \leq 3d$, where $L2$ is the length of parallelepiped in Y-direction, and d is the measurement distance. Refer Figure 7.37. The microphone positions are in the center of each partial area and at each corner of the partial area (excluding the corners intruding into reflecting planes).

N3 Number of subdivisions in Z-direction (Figure 7.39). Select the value of $N3$ such that the length of the rectangular partial area formed by these subdivisions satisfies the criterion: $\frac{L3}{N3} \leq 3d$, where $L3$ is the length of parallelepiped in Z-direction, and d is the measurement distance. Refer Figure 7.37. The microphone positions are in the center of each partial area and at each corner of the partial area (excluding the corners intruding into reflecting planes).

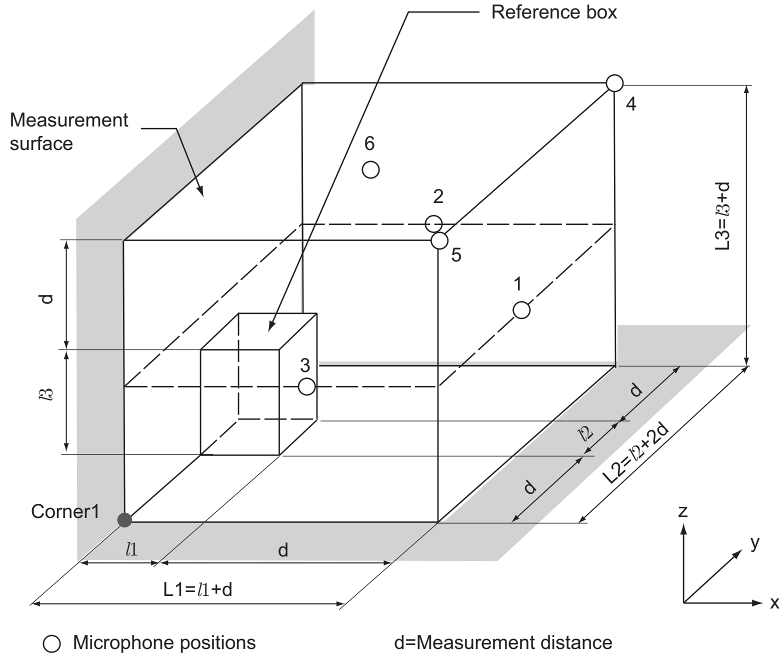


Figure 7.39: ISO3744 - Parallelepiped measurement surface with microphone positions for a source placed on the floor against a wall.

Parallelepiped against corner Use this measurement surface when the source under test is placed on a floor against two walls. Figure 7.40 shows a parallelepiped measurement surface whose sides are parallel to those of the reference box. Click on the **Suggest** button to auto fill the measurement surface variables in agreement with the standard. Verify the input you have entered by clicking on the **Check** button. *Coustyx* checks to see if the input variables satisfy the standard requirements. If the standard requirements are not met, a message window pops up to help you make appropriate corrections.

X-Axis Specify the orientation of the X-axis here. See Figure 7.40 for definition.

Y-Axis Specify the orientation of the Y-axis here. See Figure 7.40 for definition.

Corner1 Specify Corner1 of the parallelepiped surface here. See Figure 7.40 for definition. The parallelepiped is constructed using Corner1 as the starting point and $L1$, $L2$, $L3$ as its dimensions along X, Y, Z-axis respectively.

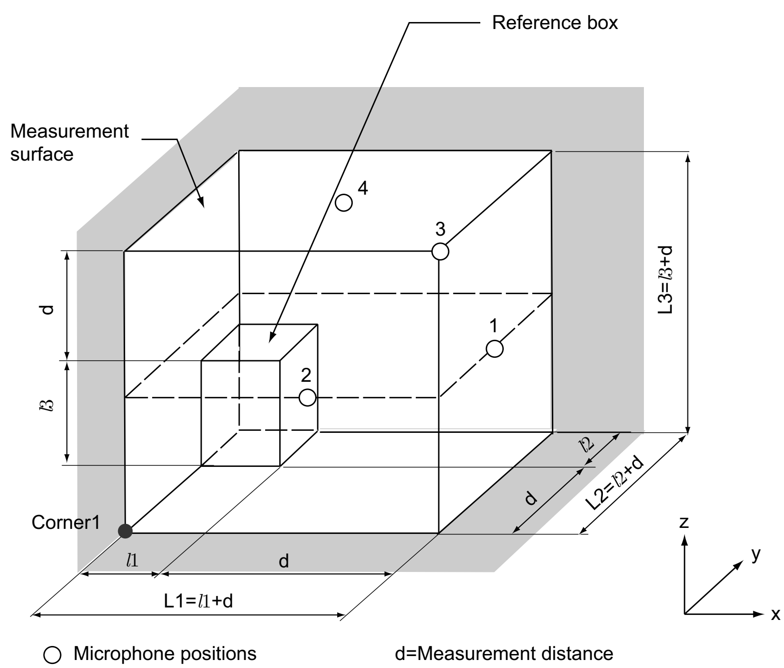


Figure 7.40: ISO3744 - Parallelepiped measurement surface with microphone positions for a source placed on the floor against two walls.

L1 Length of the parallelepiped in X-direction (Figure 7.40). Specify the value of $L1$ such that it satisfies the definition, $L1 = l1 + d$, where $l1$ is the reference box dimension in X-direction, and d is the measurement distance. The recommended value for $d = 1\,m$.

L2 Length of the parallelepiped in Y-direction (Figure 7.40). Specify the value of $L2$ such that it satisfies the definition, $L2 = l2 + 2d$, where $l2$ is the reference box dimension in Y-direction, and d is the measurement distance. The recommended value for $d = 1\,m$.

L3 Length of the parallelepiped in Z-direction (Figure 7.40). Specify the value of $L3$ such that it satisfies the definition, $L3 = l3 + d$, where $l3$ is the reference box dimension in Z-direction, and d is the measurement distance. The recommended value for $d = 1\,m$.

N1 Number of subdivisions in X-direction (Figure 7.40). Select the value of $N1$ such that the length of the rectangular partial area formed by these subdivisions satisfies the criterion: $\frac{L1}{N1} \leq 3d$, where $L1$ is the length of parallelepiped in X-direction, and d is the measurement distance. Refer Figure 7.37. The microphone positions are in the center of each partial area and at each corner of the partial area (excluding the corners intruding into reflecting planes).

N2 Number of subdivisions in Y-direction (Figure 7.40). Select the value of $N2$ such that the length of the rectangular partial area formed by these subdivisions satisfies the criterion: $\frac{L2}{N2} \leq 3d$, where $L2$ is the length of parallelepiped in Y-direction, and d is the measurement distance. Refer Figure 7.37. The microphone positions are in the center of each partial area and at each corner of the partial area (excluding the corners intruding into reflecting planes).

N3 Number of subdivisions in Z-direction (Figure 7.40). Select the value of $N3$ such that the length of the rectangular partial area formed by these subdivisions satisfies the criterion: $\frac{L3}{N3} \leq 3d$, where $L3$ is the length of parallelepiped in Z-direction, and d is the measurement distance. Refer Figure 7.37. The microphone positions are in the center of each partial area and at each corner of the partial area (excluding the corners intruding into reflecting planes).

The sound power level, L_w, from ISO3744 standard is computed as follows:

$$L_w = \bar{L}_{pf} + 10\log_{10}\left(\frac{S}{S_0}\right)\ \text{dB}$$

where S is the area of the measurement surface, in square meters; $S_0 = 1\,\text{m}^2$.

$\bar{L}_{pf}$ is the weighted surface sound pressure level over the measurement surface in decibels.

$$\bar{L}_{pf} = 10\log_{10}\left(\frac{1}{N}\left[\sum_{i=1}^{N} 10^{0.1(L_{pi}+W)}\right]\right)\ \text{dB}$$

N is the number of microphone positions associated with equal partial areas on the measurement surface.

W is the weight applied by the filter at the frequency of analysis (Figure 7.31).

L_{pi} is the sound pressure level measured at a microphone position.

$$L_{pi} = 10\log_{10}\left(\frac{p_i^2}{p_0^2}\right)\ \text{dB}$$

where p_i is the root mean square pressure (in Pascals), and $p_0 = 2 \times 10^{-5}$ Pa.

7.2.4.4. ISO 3745

ISO 3745 is an international standard that specifies methods to determine sound power levels of noise sources using sound pressure measurements in anechoic or semi-anechoic rooms ([3]). The sound pressure levels are measured on a measurement surface enveloping the source. This method produces results that are of precision grade (or grade 1) accuracy. Figure 7.41.

Definitions of some commonly used terms in the standard:

- **Measurement surface**: A hypothetical surface enveloping the noise source, on which measurement points are located. For measurements in semi-anechoic rooms the measurement surface terminates on the reflecting plane.

  ISO 3745 allows two different measurement surfaces. They are, **Sphere**, **Hemisphere**.

Lowest Frequency Specify the lowest frequency of interest in hertz. The measurement sphere or hemisphere radius is selected based on this value.

Ambient Pressure Specify the barometric pressure during the measurements in Pascals. This is used to compute environmental corrections to the sound power level value.

Ambient Temperature Specify the air temperature during measurements in degrees Celsius. This is used to compute environmental corrections to the sound power level value.

Sphere Use sphere measurement surface when you want to simulate measurements taken in an anechoic room. The definitions of **Center**, **X-Axis**, and **Y-Axis** are similar to that shown in Figure 7.33. Click on the **Suggest** button to auto fill the measurement surface variables in agreement with the standard. Verify the input you have entered by clicking on the **Check** button. *Coustyx* checks to see if the input variables satisfy the standard requirements. If the standard requirements are not met, a message window pops up to help you make appropriate corrections.

X-Axis Specify the orientation of the X-axis here. See Figure 7.33 for definition.

Y-Axis Specify the orientation of the Y-axis here. See Figure 7.33 for definition.

Center Specify the center of the sphere surface here. Select the coordinates of the sphere center to be same as the acoustic center of the source. As the location of the acoustic center is frequently not known, select the geometric center of the source, instead.

Radius The radius of the sphere surface shall be equal to or greater than any of the following:

- twice the largest source dimension;
- $\frac{\lambda}{4}$ of the lowest frequency of interest; and
- $1\,m$.

Microphone Array Choose any of the available types of microphone arrays from the drop-down menu.

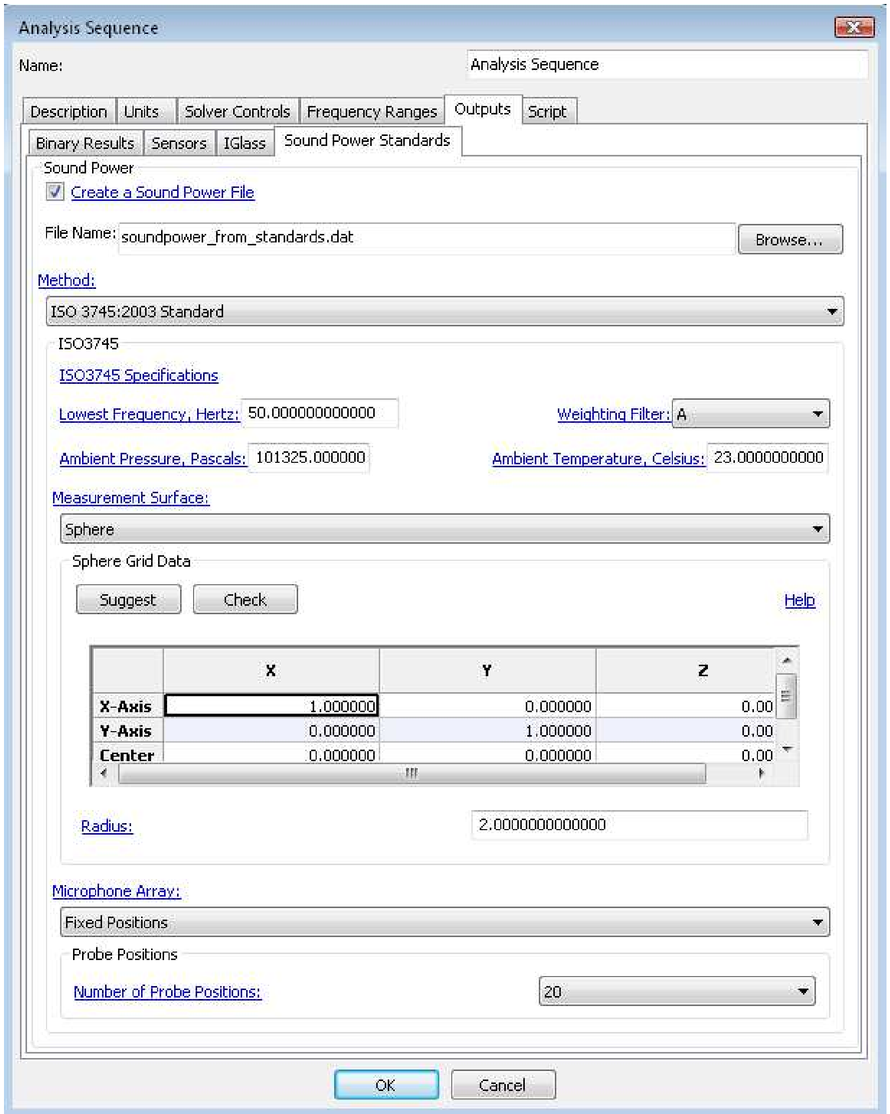


Figure 7.41: ISO3745 sound power standard window.

Table 7.8: ISO3745 - Coordinates of microphone positions on a sphere ([3]).

Microphone position	$\frac{x}{r}$	$\frac{y}{r}$	$\frac{z}{r}$
1	-1	0	0.05
2	0.49	-0.86	0.15
3	0.48	0.84	0.25
4	-0.47	0.81	0.35
5	-0.45	-0.77	0.45
6	0.84	0	0.55
7	0.38	0.66	0.65
8	-0.66	0	0.75
9	0.26	-0.46	0.85
10	0.31	0	0.95
11	1	0	-0.05
12	-0.49	0.86	-0.15
13	-0.46	-0.84	-0.25
14	0.47	-0.81	-0.35
15	0.45	0.77	-0.45
16	-0.84	0	-0.55
17	-0.38	-0.66	-0.65
18	0.66	0	-0.75
19	-0.26	0.46	-0.85
20	-0.31	0	-0.95

Fixed Positions Choose this option to specify an array of fixed microphone positions associated with equal partial areas distributed over the sphere surface.

Number of Probe Positions Choose the number of microphones to be 20 or 40 from the drop-down menu. In general, 20 microphone positions are sufficient. However, when the source has high directivity increase the number of microphone positions to 40. Table 7.8 lists coordinates of 20 microphone positions. The additional 20-point array can be obtained by rotating the original array by 180^o about the Z-axis.

Coaxial Circular Paths Figure 7.34 shows coaxial circular paths in parallel planes traversed by a microphone over the top half of the sphere. Repeat the same for the bottom half of the sphere with heights chosen symmetrical to the top half paths. The paths are selected such that the annular area associated with each path is the same.

Number of Circular Paths Traversed by a Probe Enter the number of circular paths traversed by a microphone on each half of the sphere.

Meridional Arc Traverses Figure 7.42 shows meridional arc traversed by a microphone. The meridional arc is the path traversed along a semicircular arc about a horizontal axis through the center of the source. The paths are selected such that the annular area associated with each path is the same.

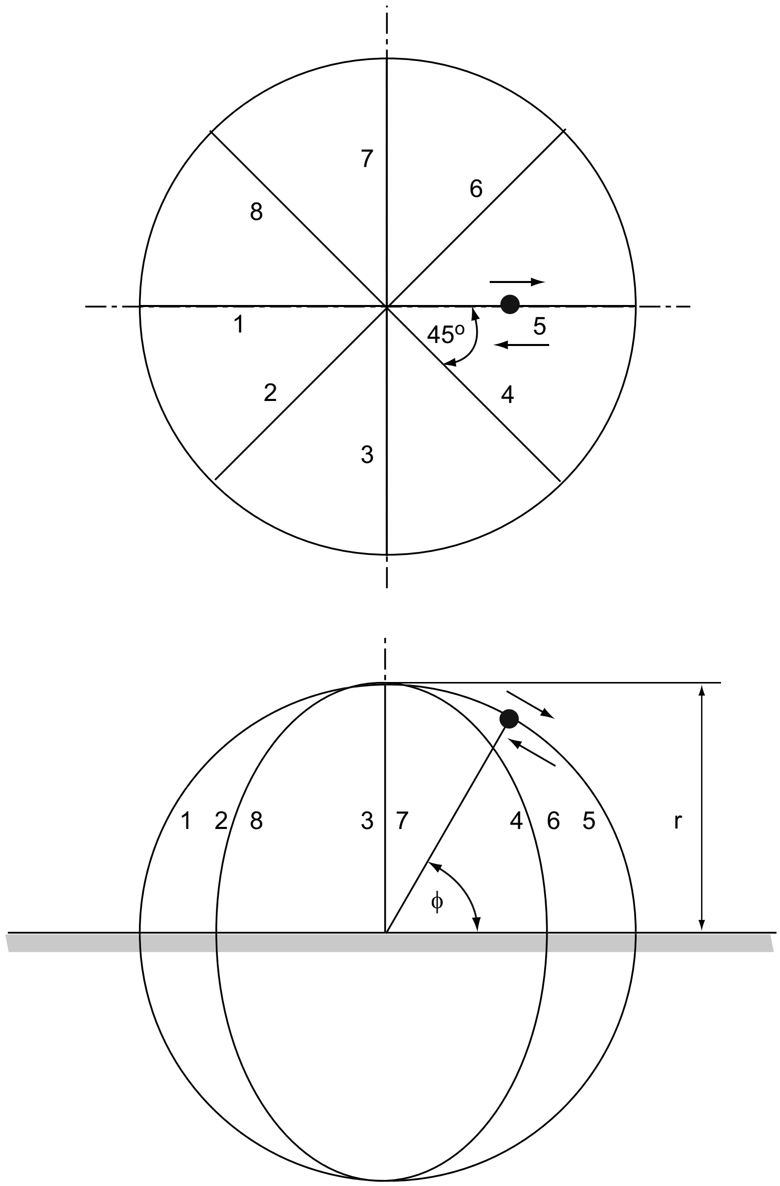


Figure 7.42: ISO3745 - Meridional paths for a moving microphone.

Number of Probe Traverses Enter the number of microphone traverses at equal increments of azimuth angle around the source (Figure 7.42). The least number of microphone traverses is eight.

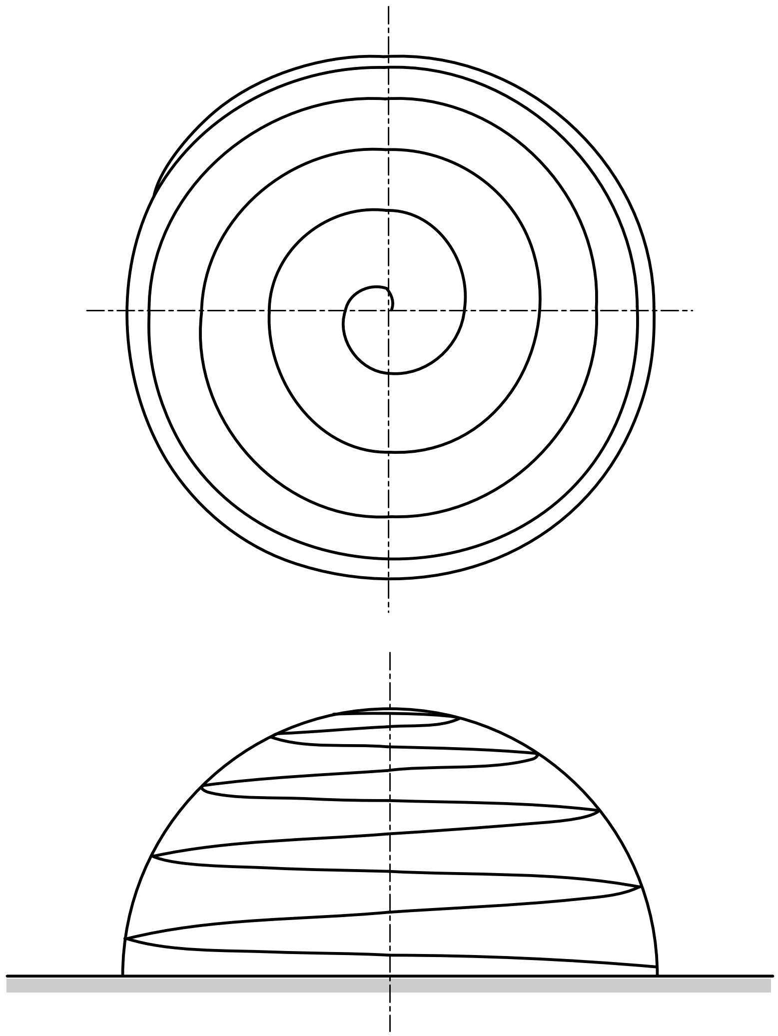


Figure 7.43: ISO3745 - Spiral path for a moving microphone.

Spiral Path This method uses a traverse along one meridional path and simultaneously traverses the microphone through an integral number of circular paths, thus forming a spiral path around the vertical axis of measurement surface. Figure 7.43 shows spiral path traversed by a microphone. The least number of circular turns that shall be completed by the microphone is five.

Number of Circular Paths Traversed by a Probe Enter the number of complete circular turns traversed by a microphone to form the spiral path. (Figure 7.43). The least number is five.

Hemisphere Use hemisphere measurement surface when you want to simulate measurements taken in a semi-anechoic room. The definitions of **Center**, **X-Axis**, and **Y-Axis** are similar to that shown in Figure 7.33. Click on the **Suggest** button to auto fill the measurement surface variables in agreement with the standard. Verify the input you have entered by clicking on the **Check** button. *Coustyx* checks to see if the input variables satisfy the standard requirements. If the standard requirements are not met, a message window pops up to help you make appropriate corrections.

X-Axis Specify the orientation of the X-axis here. See Figure 7.33 for definition.

Y-Axis Specify the orientation of the Y-axis here. See Figure 7.33 for definition.

Center Specify the center of the hemisphere surface here. Select the coordinates of the center by projecting the acoustic center of the sound source on the floor of the semi-anechoic room. As the location of the acoustic center is frequently not known, select the geometric center of the source, instead.

Radius The radius of the hemisphere surface shall be equal to or greater than any of the following:

- twice the largest source dimension or three times the distance of the acoustic center of the source from the reflecting plane, whichever is larger;
- $\frac{\lambda}{4}$ of the lowest frequency of interest; and
- $1\,m$.

Microphone Array Choose any of the available types of microphone arrays from the drop-down menu.

Fixed Positions Choose this option to specify an array of fixed microphone positions associated with equal partial areas distributed over the hemisphere surface.

Number of Probe Positions Choose the number of microphones to be 20 or 40 from the drop-down menu. In general, 20 microphone positions are sufficient. However, when the source has high directivity increase the number of microphone positions to 40. Table 7.9 lists coordinates of 20 microphone positions. The additional 20-point array can be obtained by rotating the original array by 180^o about the Z-axis.

Coaxial Circular Paths Figure 7.34 shows coaxial circular paths in parallel planes traversed by a microphone over a reflecting plane. The paths are selected such that the annular area associated with each path is the same.

Number of Circular Paths Traversed by a Probe Enter the number of circular paths traversed by a microphone on the hemisphere. The least number is five.

Meridional Arc Traverses Figure 7.42 shows meridional arc traversed by a microphone. The meridional arc is the path traversed along a semicircular arc about a horizontal axis through the center of the source. The paths are selected such that the annular area associated with each path is the same.

Table 7.9: ISO3745 - Coordinates of microphone positions on a hemisphere ([3]).

Microphone position	$\frac{x}{r}$	$\frac{y}{r}$	$\frac{z}{r}$
1	-1	0	0.025
2	0.5	-0.86	0.075
3	0.5	0.86	0.125
4	-0.49	0.85	0.175
5	-0.49	-0.84	0.225
6	0.96	0	0.275
7	0.47	0.82	0.325
8	-0.93	0	0.375
9	0.45	-0.78	0.425
10	0.88	0	0.475
11	-0.43	0.74	0.525
12	-0.41	-0.71	0.575
13	0.39	-0.68	0.625
14	0.37	0.64	0.675
15	-0.69	0	0.725
16	-0.32	-0.55	0.775
17	0.57	0	0.825
18	-0.24	0.42	0.875
19	-0.38	0	0.925
20	0.11	-0.19	0.975

Number of Probe Traverses Enter the number of microphone traverses at equal increments of azimuth angle around the source (Figure 7.42). The least number of microphone traverses is eight.

Spiral Path This method uses a traverse along one meridional path and simultaneously traverses the microphone through an integral number of circular paths, thus forming a spiral path around the vertical axis of measurement surface. Figure 7.43 shows spiral path traversed by a microphone. The least number of circular turns that shall be completed by the microphone is five.

Number of Circular Paths Traversed by a Probe Enter the number of complete circular turns traversed by a microphone to form the spiral path. (Figure 7.43). The least number is five.

The sound power level, L_w, from ISO3745 standard is computed as follows:

$$L_w = \bar{L}_{pf} + 10\log_{10}\left(\frac{S}{S_0}\right)\,\text{dB} + C_1 + C_2$$

$$C_1 = -10\log_{10}\left[\frac{B}{B_0}\sqrt{\left(\frac{313.15}{273.15+\theta}\right)}\right]\,\text{dB}$$

$$C_2 = -15\log_{10}\left[\frac{B}{B_0}\left(\frac{296.15}{273.15+\theta}\right)\right]\,\text{dB}$$

where S is the area of the measurement surface, in square meters; $S_0 = 1\,\text{m}^2$; B is the barometric pressure during measurements, in Pascals; B_0 is the reference barometric pressure, 1.01325×10^5 Pa; θ is the air temperature during measurement, in degrees Celsius.

$\bar{L}_{pf}$ is the weighted surface sound pressure level over the measurement surface in decibels.

$$\bar{L}_{pf} = 10\log_{10}\left(\frac{1}{N}\left[\sum_{i=1}^{N} 10^{0.1(L_{pi}+W)}\right]\right)\,\text{dB}$$

N is the number of microphone positions associated with equal partial areas on the measurement surface.

W is the weight applied by the filter at the frequency of analysis (Figure 7.31).

L_{pi} is the sound pressure level measured at a microphone position.

$$L_{pi} = 10\log_{10}\left(\frac{p_i^2}{p_0^2}\right)\,\text{dB}$$

where p_i is the root mean square pressure (in Pascals), and $p_0 = 2 \times 10^{-5}$ Pa.

7.2.4.5. ISO 9614-1

ISO9614-1 is an international standard that specifies a method for determining the sound power level of a noise source by measuring the component of sound intensity normal to a measurement surface enveloping the source. Figure 7.44.

Definitions of some commonly used terms in the standard:

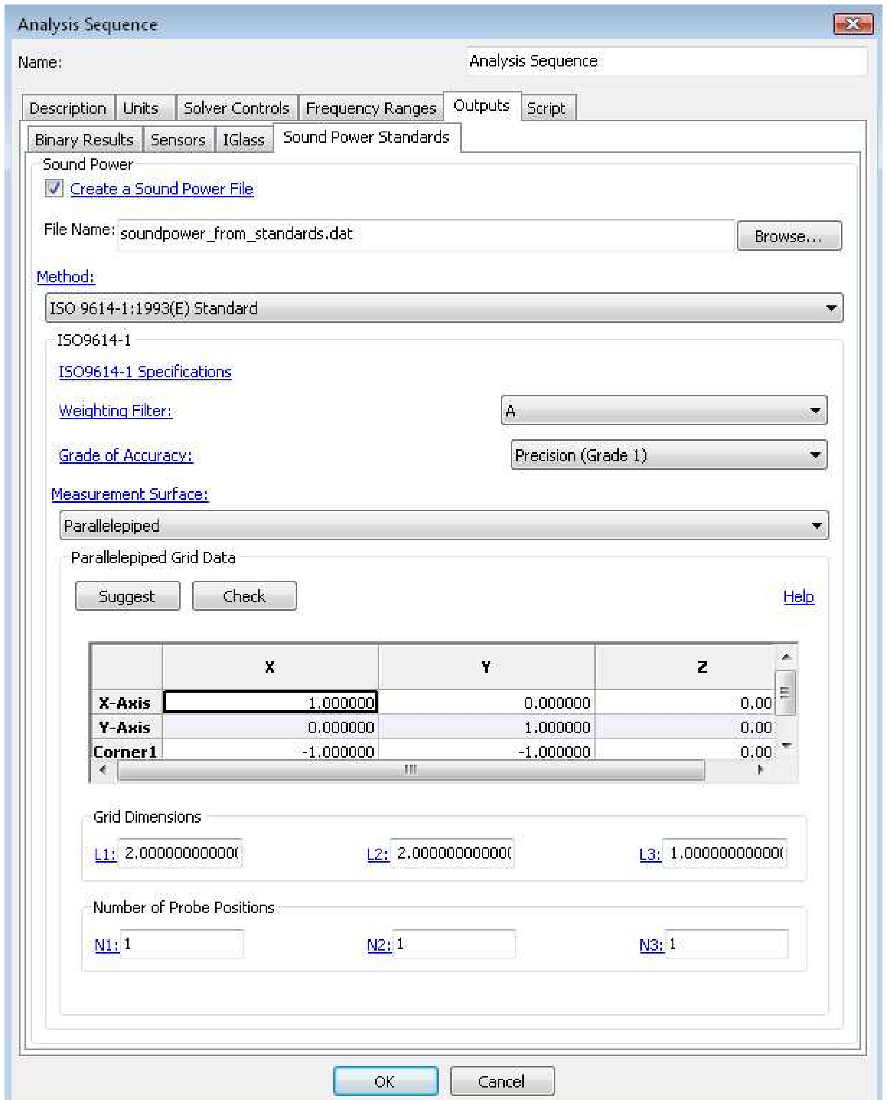


Figure 7.44: ISO9614-1 sound power standard window.

- **Measurement surface**: A hypothetical surface enveloping the noise source, on which measurement points are located. The measurement surface terminates on the reflecting plane.

  Two different measurement surfaces choices are provided. They are, **Hemisphere**, and **Parallelepiped**.

Grade of Accuracy Specify the grade of accuracy by choosing from the following options in the drop-down menu: Precision grade (Grade 1), Engineering grade (Grade 2), and Survey grade (Grade 3). This information is used to verify the adequacy of the chosen array of measurement positions and the dimensions of the measurement surface at the end of computations. If the measurement surface dimension(s) or the number of microphone positions are found inadequate appropriate log messages will be displayed. Make necessary modifications and redo the sound power level calculations to achieve the required accuracy.

Hemisphere Refer to Figure 7.33 for the definitions of **Center**, **X-Axis**, and **Y-Axis** for a hemisphere surface used in ISO9614-1 standard. Click on the **Suggest** button to auto fill the measurement surface variables in agreement with the standard. Verify the input you have entered by clicking on the **Check** button. *Coustyx* checks to see if the input variables satisfy the standard requirements. If the standard requirements are not met, a message window pops up to help you make appropriate corrections.

X-Axis Specify the orientation of the X-axis here. See Figure 7.33 for definition.

Y-Axis Specify the orientation of the Y-axis here. See Figure 7.33 for definition.

Center Specify the center of the hemisphere surface here. Select the coordinates of the center by projecting the acoustic center of the sound source on the reflecting plane. As the location of the acoustic center is frequently not known, select the geometric center of the source, instead.

Radius The radius of the hemisphere surface shall be equal to or greater than any of the following:

- twice the largest source dimension or three times the distance of the acoustic center of the source from the reflecting plane, whichever is larger; and
- $1\,m$.

Number of Probes Enter the number of probe positions, N, to be distributed on the surface of the hemisphere. Select $N = 2n^2$, where n is an integer. The hemisphere surface is divided into n divisions along θ direction, and $2n$ divisions along ϕ direction. Note: θ is the angle made by a point on the hemisphere with the Z-axis, and ϕ is the angle made with X-axis by the projection of the point on the X-Y plane. Probes are placed at the center of each of the partial areas formed with these divisions.

Parallelepiped Figure 7.45 shows a parallelepiped measurement surface whose sides are parallel to those of the surface of the source. Click on the **Suggest** button to auto fill the measurement surface variables in agreement with the standard. Verify the input you have entered by clicking on the **Check** button. *Coustyx* checks to see if the input variables satisfy the standard requirements. If the standard requirements are not met, a message window pops up to help you make appropriate corrections.

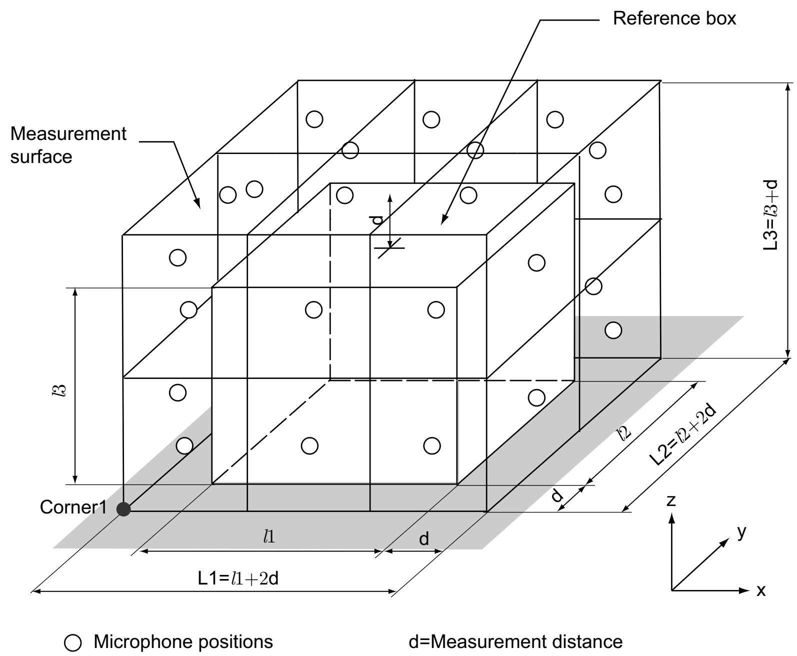


Figure 7.45: ISO9614-1 - Parallelepiped measurement surface with probe positions.

X-Axis Specify the orientation of the X-axis here. See Figure 7.45 for definition.

Y-Axis Specify the orientation of the Y-axis here. See Figure 7.45 for definition.

Corner1 Specify Corner1 of the parallelepiped surface here. See Figure 7.45 for definition. The parallelepiped is constructed using Corner1 as the starting point and $L1$, $L2$, $L3$ as its dimensions along X, Y, Z-axis respectively.

L1 Length of the parallelepiped in X-direction (Figure 7.45). Specify the value of $L1$ such that it satisfies the definition, $L1 = l1 + 2d$, where $l1$ is the reference box dimension in X-direction, and d is the measurement distance. The recommended value for $d = 1\,m$.

L2 Length of the parallelepiped in Y-direction (Figure 7.45). Specify the value of $L2$ such that it satisfies the definition, $L2 = l2 + 2d$, where $l2$ is the reference box dimension in Y-direction, and d is the measurement distance. The recommended value for $d = 1\,m$.

L3 Length of the parallelepiped in Z-direction (Figure 7.45). Specify the value of $L3$ such that it satisfies the definition, $L3 = l3 + d$, where $l3$ is the reference box dimension in Z-direction, and d is the measurement distance. The recommended value for $d = 1\,m$.

N1 Number of subdivisions in X-direction (Figure 7.45). Select the values of $N1$, $N2$, $N3$ such that there is a minimum of one probe position per square metre, and a minimum of 10 positions distributed as uniformly as possible (according to segment area) over the measurement surface.

N2 Number of subdivisions in Y-direction (Figure 7.45). Select the values of $N1$, $N2$, $N3$ such that there is a minimum of one probe position per square metre, and a minimum of 10 positions distributed as uniformly as possible (according to segment area) over the measurement surface.

N3 Number of subdivisions in Z-direction (Figure 7.45). Select the values of $N1$, $N2$, $N3$ such that there is a minimum of one probe position per square metre, and a minimum of 10 positions distributed as uniformly as possible (according to segment area) over the measurement surface.

The sound power level, L_w, from ISO9614-1 standard is computed as follows:

$$L_w = 10\log_{10}\left(\sum_{i=1}^{N}\frac{P_i}{P_0}\right)\,\mathrm{dB} + W$$

where $P_0 = 10^{-12}$ is the reference sound power; N is the total number of measurement positions and segments; W is the weight applied by the filter at the frequency of analysis (Figure 7.31);

P_i is the partial sound power for segment i, that is,

$$P_i = I_{ni}S_i$$

where I_{ni} is the signed magnitude of the normal sound intensity component measured at position i on the measurement surface; S_i is the area of segment i.

7.2.5. Sound Power

Coustyx automatically creates an *ASCII* data file entitled *power.dat* while running an analysis. This file contains acoustic sound power values computed at each analysis frequency. It has five columns. The first column contains analysis frequencies in 'Hertz'. The second and third columns contain the radiated (active) sound power and the reactive sound power respectively. The input power is in the fourth column with the same unit. The units of power are consistent with the material properties - sound speed, and ambient density. The final column consists of the radiation efficiency.

The sound power W through an area S is given by,

$$W = \int \mathbf{I}.d\mathbf{S} \tag{7.1}$$

where $\mathbf{I} = \mathrm{P}\mathbf{V}$ is instantaneous sound intensity in the direction of particle velocity $\mathbf{V}$; and P is the instantaneous pressure at that point. For more details on the definition of sound intensity refer to Section 7.2.5.1.

Acoustic sound power definitions in *Coustyx*.

Radiated (Active) Sound Power Radiated (Active) sound power is the rate at which acoustic energy is radiated from a source. It is computed from the mean active acoustic intensity I_a (refer to Equation 7.11) over a surface S, that is,

$$W_a = \int I_a d\mathrm{S} \tag{7.2}$$

Reactive Sound Power Reactive sound power is a measure of the acoustic energy which keeps transforming back and forth between potential and kinetic energies. The reactive sound power is computed from the amplitude of the reactive sound intensity I_r (refer to Equation 7.12) over a surface area S, that is,

$$W_r = \int I_r d\mathrm{S} \tag{7.3}$$

Note that the reactive sound power values in *Coustyx* are computed at the surface of a structure.

Input Sound Power The input power for a closed surface S is defined as,

$$W = \frac{1}{2}\rho_o c \int v_n^2 dS \tag{7.4}$$

where ρ_o is the medium density, c is the speed of sound in the medium, v_n is the particle normal velocity at the structure surface. The input sound power is used to estimate the radiation efficiency of a structure.

Radiation Efficiency Radiation efficiency is a useful measure to understand the radiation characteristics of structures. It is defined as the ratio of the radiated (active) sound power to the input sound power, that is,

$$\sigma = \frac{W_a}{W_i} \tag{7.5}$$

7.2.5.1. Complex Intensity

This section provides a concrete definition of complex sound intensity and the meaning of the real part and imaginary parts. The derivation follows Fahy [4], with appropriate modifications to account for our convention for harmonic variation, namely $exp(-i\omega t)$.

Let $P(x,t)$ and $V(x,t)$ represent the instantaneous sound pressure, and acoustic particle velocity (in a certain direction) at the location x. They are given as

$$P(x,t) = \mathrm{Re}\left[p(x)\exp(i\phi_p)\exp(-i\omega t)\right] = p(x)\cos(\phi_p - \omega t) = p_r(x)\cos\omega t + p_i(x)\sin\omega t \quad (7.6)$$

$$V(x,t) = \mathrm{Re}\left[v(x)\exp(i\phi_v)\exp(-i\omega t)\right] = v(x)\cos(\phi_v - \omega t) = v_r(x)\cos\omega t + v_i(x)\sin\omega t \quad (7.7)$$

The instantaneous sound intensity is the product of sound pressure and acoustic particle velocity at a point and is given as follows:

$$\begin{aligned} I(x,t) &= P(x,t)\cdot V(x,t) \\ &= p(x)v(x)\cos(\phi_p - \omega t)\cos(\phi_v - \omega t) \\ &= \tfrac{p(x)v(x)}{2} 2\cos(\phi_p - \omega t)\cos(\phi_v - \omega t) \\ &= \tfrac{p(x)v(x)}{2}\left[\cos(\phi_p - \omega t + \phi_v - \omega t) + \cos(\phi_p - \phi_v)\right] \\ &= \tfrac{p(x)v(x)}{2}\left[\cos(2\phi_p - 2\omega t - \phi_r) + \cos\phi_r\right] \\ &= \tfrac{p(x)v(x)}{2}\left[\cos(2(\phi_p - \omega t))\cos\phi_r + \sin(2(\phi_p - \omega t))\sin\phi_r + \cos\phi_r\right] \\ &= \tfrac{p(x)v(x)\cos\phi_r}{2}\left[1 + \cos(2(\phi_p - \omega t))\right] + \tfrac{p(x)v(x)\sin\phi_r}{2}\sin(2(\phi_p - \omega t)) \\ &= \tfrac{p_r(x)v_r(x)+p_i(x)v_i(x)}{2}\left[1 + \cos(2(\phi_p - \omega t))\right] + \tfrac{p_i(x)v_r(x)-p_r(x)v_i(x)}{2}\sin(2(\phi_p - \omega t)) \end{aligned} \quad (7.8)$$

The complex exponential representation of sound intensity is given as

$$\begin{aligned} I(x,t) &= \mathrm{Re}\left\{\tilde{C}(x)\left[1 + \exp(-2i(-\omega t + \phi_p))\right]\right\} \\ &= \mathrm{Re}\left\{(C_r(x) + iC_i(x))\left[1 + \cos(2(\phi_p - \omega t)) - i\sin(2(\phi_p - \omega t))\right]\right\} \\ &= C_r(x)\left[1 + \cos(2(\phi_p - \omega t))\right] + C_i(x)\sin(2(\phi_p - \omega t)) \end{aligned} \quad (7.9)$$

The real part of $\tilde{C}(x)$ is $C_r(x)$ and represents the *mean* active intensity. The imaginary part of $\tilde{C}(x)$ is $C_i(x)$ represents the *amplitude* of the reactive intensity.

$$\begin{aligned} \tilde{C}(x) &= C_r(x) + iC_i(x) \\ &= I_a(x) + iI_r(x) \end{aligned} \quad (7.10)$$

Comparing Equation 7.8 with Equation 7.9, we get the expressions for mean active intensity $I_a(x)$ and amplitude of reactive intensity as follows:

$$I_a(x) = C_r(x) = \frac{p_r(x)v_r(x) + p_i(x)v_i(x)}{2} = \frac{1}{2}\mathrm{Re}(\tilde{p}(x)\tilde{v}^*(x)) \quad (7.11)$$

$$I_r(x) = C_i(x) = \frac{p_i(x)v_r(x) - p_r(x)v_i(x)}{2} = \frac{1}{2}\mathrm{Im}(\tilde{p}(x)\tilde{v}^*(x)) \quad (7.12)$$

where $\tilde{p}(x) = p_r(x) + ip_i$ and $\tilde{v}(x) = v_r(x) + iv_i$.

The complex intensity $\tilde{C}(x)$ is given as

$$\tilde{C}(x) = \frac{1}{2}\tilde{p}(x)\tilde{v}^*(x) \tag{7.13}$$

7.3. Script

The window **Script** shows the script used by *Coustyx* to run the analysis. The script is generated automatically based on the options selected in **Solver Controls**, **Frequency Ranges**, and **Outputs** windows. A sample script is shown in Figure 7.46. Advanced users can directly modify the script to suit their needs. *Coustyx* syntax is presented in Chapter 8.

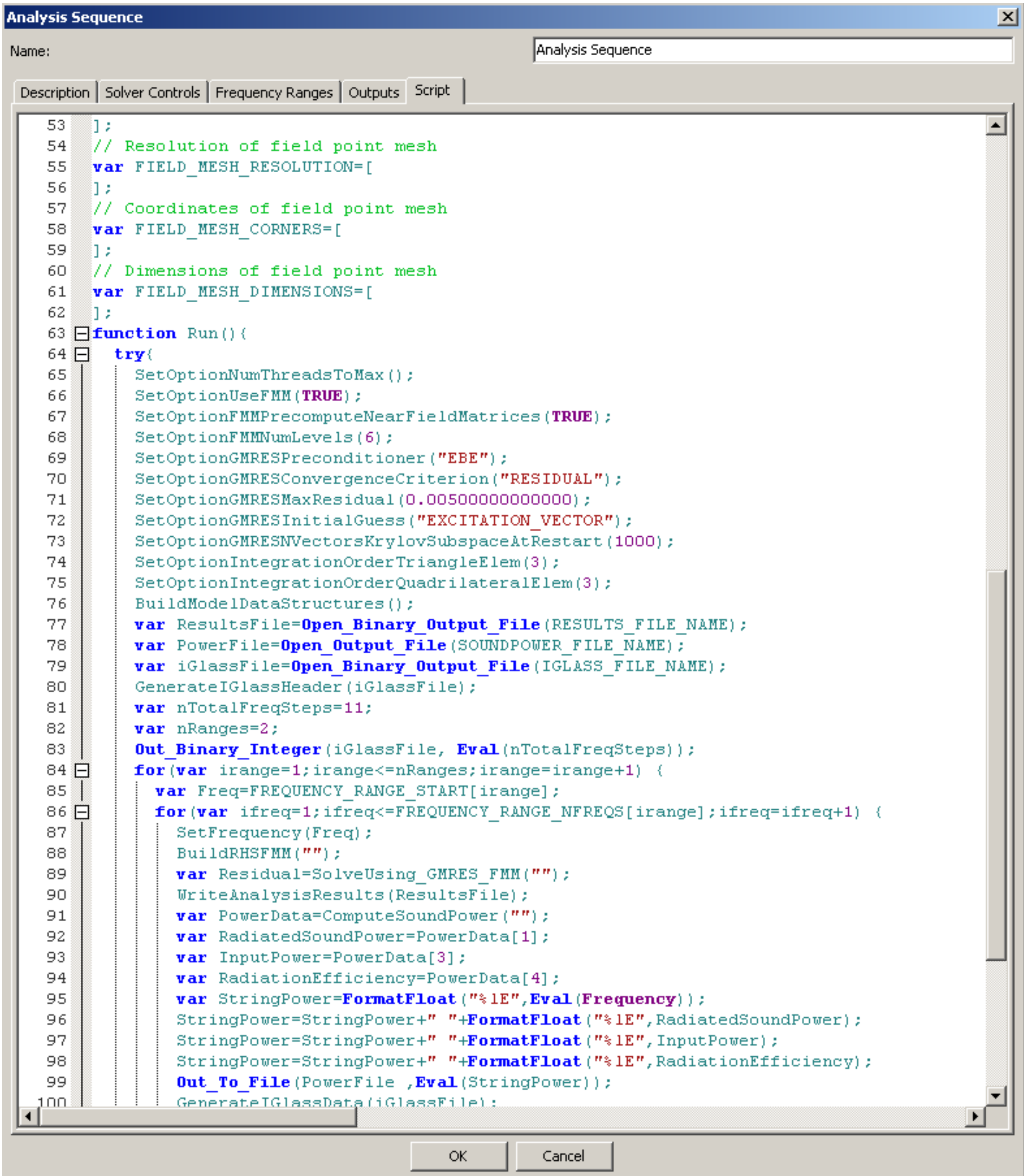


Figure 7.46: Analysis script.

Bibliography

[1] *American National Standard - Specification for Sound Level Meters.* Acoustical Society of America, 1983. Reference no. ANSI S1.4-1983(R2006).

[2] *ISO 3744: Acoustics - Determination of sound power levels of noise sources using sound pressure - Engineering method in an essentially free field over a reflecting plane.* International Organization for Standarization (ISO), 1994. Reference no. ISO 3744:1994(E).

[3] *ISO 3745: Acoustics - Determination of sound power levels of noise sources using sound pressure - Precision methods for anechoic and hemi-anechoic rooms.* International Organization for Standarization (ISO), 2003. Reference no. ISO 3745:2003(E).

[4] F. J. Fahy. *Sound Intensity.* Elsevier Applied Science, 1989. Page 56.

[5] A. D. Pierce. *Acoustics - An introduction to its physical principles and applications.* Acoustical Society of America, 1991. Pages 57–60.

Chapter 8

Language Syntax

A user provides input and instructions to *Coustyx* through a special programming language. This user interaction language consists of comments, constants, variables, function calls, arrays, expressions, and statements.

8.1. Comments

Single line comments are started by the characters '//' Multiple line comments are started with a '/*' and terminated by a '*/'

```
x=x+2; // This is a single line comment
```

```
x=x+2;  /* This is a multiple
           line comment */
```

8.2. Constants

Three type of constants are accepted: Numeric, Boolean, and string constants. Numeric constants follow the same conventions as in FORTRAN. All constants are treated internally as double precision floating point values, regardless of the format in which the user specifies them. Some examples of numeric constants are:

```
0 1.0 0.123e-20 .123 34324.
```

Only two Boolean constants exist:

```
TRUE, FALSE
```

String constants consist of a sequence of ASCII characters enclosed in double quotes:

```
"This is a string"
```

Strings may also contain certain escape sequences:

```
"This is a tab:\t and a newline:\n"
```

The escape sequences that are available are:
`"\b"` : Backspace
`"\t"`: Tab
`"\n"`: Newline
`"\r"`: Carriage return

8.2.1. Built-in Constants

Some of the built-in constants in *Coustyx* are listed below.
PI = 3.14159265359.

i = imaginary unit. For example, a complex number can be represented as $3 + 4 * \mathbf{i}$.

e1, e2 and e3 are the coordinate directions.

Origin = It is the coordinate origin which must be added to all position vectors. For example, a position vector is represented as $\mathbf{Origin} + 2 * \mathbf{e1} + 3 * \mathbf{e2} + 5 * \mathbf{e3}$.

8.3. Variables

Variables are referred to by their names. Variable names may be of any length, but only the first 128 characters are considered significant. The names are case sensitive, must begin with an alphabet or and underscore symbol, and all the following characters may be alphabets, numerals or underscores. Examples of valid variable names:

```
Transmission_Error
Output_Torque
Planet_1
```

All variables must be declared before they can be used. A few variables are pre-defined, and do not need to be declared. Many of these are 'Read-only', which means that their value cannot be changed.

8.3.1. Predefined Variables

Some of the predefined variables in *Coustyx* are listed below. These variables can be directly used in *Coustyx* scripts.
Frequency Analysis frequency in hertz. The value could be set by the function call `SetFrequency(freq)`.

AngularFreq Analysis frequency in radians per sec. The value could be set by the function call `SetAngularFreq(afreq)`.

SoundSpeed Speed of sound set by material properties.

AmbientDensity Density of the acoustic fluid medium set by material properties.

WaveNumber Wavenumber, $k = AngularFreq/SoundSpeed$.

8.4. Function Calls

Functions are called by invoking their name and supplying arguments in parentheses. Functions may be pre-defined, or defined by the user. Examples of function calls:

```
sin(Theta) ln(x) exp(y) Clear()
Set_Window(0,1,0,.75) Set_Viewport(0,1,0,.75)
PartialDiff(Time^2+0.1*Time,Time)
```

Parentheses are mandatory, even if no arguments are passed to the function.

8.5. Expressions:

Expressions may be formed by using constants, variables, function calls and the following symbols: Arithmentic Operators:
`+` :Addition or string concatenation
`-` :Subtraction
`*` :Multiplication
`/` :Division
`^` :To the power of
Parentheses:
`(` :Open parenthesis
`)` :Close parenthesis
Relational operators (RELOPS):
`>` :Greater than
`<` :Less than
`==` : Equal to
`<=` : Less than or equal to
`>=` :Greater than or equal to
`!=` :Not equal to
Boolean operators:
`&` :Boolean AND operator
`|` :Boolean OR operator
`!` :Boolean NOT operator
Array operators:
`[` :Start of an array list or array element extractor
`]` :End an array list or array element extractor

, :Separator for array list and argument list elements
Examples:

```
1+2
1.0092*Pi
[0,1,x^2,3/Pi]
Translate(0.001*e1-0.002*e2)*Rotate(Time*Omega,e3)
[Sin(Theta),Cos(Theta),0]
Body_Frame_Reaction_Vector(Sun_Body)[6]
e1[2]
([0,1,2,3,4,5,6])[2]
```

Note: `[0,1,2,3,4,5,6][2]` is not valid!

```
Transfm_Matrix[1][2]
([[[0,1],
   [0,1]],
  0])[1][2][1]
[[1,0,0,0],
 [0,1,0,0],
 [0,0,1,0],
 [0,0,0,1]]
```

8.6. Statements

All simple statements are terminated by a semi-colon. The statement is not processed until the semicolon is entered. A statement does not have to be contained entirely in one line.

8.6.1. Declaration Statement

The first type of statement is a declaration statement. It simply declares one or more variables for later use. If a variable is used without being formally declared, an error is generated. Examples:

```
var x;
var x,y,z;
```

8.6.2. Expression Statement

The second type of statement is an expression followed by a semicolon:

```
Set_Surface_Pairs(
    [Conformal,[CrankCase_Body     ,"InnerSeal"],
               [OrbitingScroll_Body,"PressureSurface"],
               Friction_Coeff,Separation_Tolerance_Conf,
               1,1,x1,x2,x3,Fixed_Frame
```

```
    ],
    [Conformal,[CrankCase_Body     ,"OuterSeal"],
               [OrbitingScroll_Body,"PressureSurface"],
               Friction_Coeff,Separation_Tolerance_Conf,
               1,1,x1,x2,x3,Fixed_Frame
    ]
);
```

8.6.3. Assignment Statement

The next type of statement is an assignment:

Symbolic assignment:

```
Planet_3_Runout_Error:=0.001*e1+0.00003*e2;
```

Evaluated assignment:

```
Planet_1_Runout_Error =0.001*e1+0.00003*e2;
```

8.6.4. Declaration with Assignment

A value can be assigned to a variable at the same time that it is being declared:

```
var x=1,y,z=3;
```

Arrays are allocated using the 'Dim()' function:

```
var x=Dim(20);
var x=Dim(3,5);
```

8.6.5. Compound Statement

A compound statement is formed by enclosing a sequence of zero or more statements in braces. Each of these statements may be a simple statement or a compound statement. Examples:

```
{}

{{}{}{}}

{Out("Hello");Out("World");}

{
  var x=1;
  var y=2;
  Out(x+y);
  Out(Eval(x+y));
}
```

8.6.6. Symbolic Form and Evaluation of Expressions

Expressions are manipulated by *Coustyx* in symbolic form unless it is explicitly told to evaluate them. Example:

```
var y:=1.0;
var x:=2*y+3;
Out("x=",x);
Output:
x=x
```

Example:

```
var y:=1.0;
var x:=2*y+3;
Out("x=",Eval(x));
```

Output:

```
x=5
```

The subsitute function Subst() can be used to substitute a variable by the symbolic expression it contains. Example:

```
var y:=1.0;
var x:=y;
Out("x=",Subst(x));
```

Output:

```
x=y
```

Example:

```
var y:=1.0;
var x:=2*y+3;
Out("x=",Subst(x));
```

Output:

```
x=((2*y)+3)
```

Evaluation of an expression can also be caused by using the = assignment operator instead of the := assignment operator. In that case, evaluation of the expression on the right hand side takes place before it is assigned to variable on the left hand side. Example:

```
var y:=1.0;
var x=2*y+3;
Out("x=",Subst(x));
```

Output:

```
x=5
```

Example:

```
var y:=1.0;
var x:=y;
Out("x=",Subst(Subst(x)));
```

Output:

```
x=y
```

The Subst() function does nothing if its' argument is not a simple variable. Example:

```
var y:=1.0;
var x:= 2*y+3;
Out("x=",Subst(Subst(x)));
```

Output:

```
x=((2*y)+3)
```

Circular definitions could occur, in which case evaluation will create a runtime error. Example:

```
var y;
var x:=2*y+3;
y:=2*x;
Out("x=",Eval(x));
```

Output:

```
The variable: x has been
defined in terms of itself.
```

8.6.7. if Statement

An `if` statement comes in two forms:

```
if ( <exp> ) <statement>
```

and

```
if (<exp>) <statement> else <statement>
```

Here `<exp>` is a Boolean valued expression and `<statement>` is any simple or compound statement. Examples:

```
if(i==10) Out("Value(10)=",Eval(val[10]));


if(j>2) {
  x[j]=x[j-1];
}

if(j>2) {
  x[j]=x[j-1];
} else {
  x[j]=1.0;
}
```

The second form of the `if` statement can lead to an ambiguity:

```
var j=3;
if(j>2) if(j==10) Out("OK"); else Out("Not OK");
```

The ambiguity is resolved by binding the else part to the innermost if part. So the above statement is equivalent to:

```
var j=3;
if(j>2) {
  if(j==10) {
    Out("OK");
  } else {
    Out("Not OK");
  }
}
```

In such a situation, it is recommended that braces be used to explicitly resolve the ambiguity.

8.6.8. for Statement

The `for` statement takes the following form:

```
for ( <assign_decl_exp> ; <exp> ; <assign_exp> ) <statement>
```

Here <assign_decl_exp> is either a simple expression, or a declaration (with or without an assignment). It is executed once at the beginning of the `for` loop. <exp> is any simple Boolean valued expression. It is executed before each iteration. If it evaluates to TRUE, then the simple or compound statement in <statement> is executed, otherwise the loop terminates. <assign_exp> is an expression with or without an assignment. It is executed at the end of each iteration. It is usually used to increment some counter. Example:

```
var i,x=Dim(10);
for(i=1;i<=10;i=i+1) x[i]=0.0;
```

It is preferable to declare the loop variable inside the `for` statement as shown below. This limits the visibility of the variable to the `<statement>` part of the `for` statement. If this variable is referred to anywhere else in the program, an error will be generated.

```
var x=Dim(10);
for(var i=1;i<=10;i=i+1) x[i]=0.0;
```

```
var i;
function Initialize(){
  i=1;
}
function Done(){
  return i>=10;
}
function Increment(){
  i=i+1;
}
function DoSomething(){
  Out("Hi ",Eval(i));
}
for(Initialize();!Done();Increment()) {
  DoSomething();
}
```

8.6.9. while Statement

The `while` statement is a simpler alternative to the `for` statement:

```
while ( <exp> ) <statement>
```

`<exp>` is a Boolean valued expression, and `<statement>` is a simple or compound statement. The expression `<exp>` is evaluated first. If its value is `TRUE`, then `<statement>` is executed. Otherwise the while statement is terminated. This process is repeated until `<exp>` evaluates to `FALSE`.

```
var i;
function Done(){
  return i>=10;
}
function Increment(){
  i=i+1;
}
function DoSomething(){
  Out("Hi ",Eval(i));
```

```
}
i=1;
while(!Done()) {
  DoSomething();
  Increment();
}
```

8.6.10. do-while Statement

The `do-while` statement is similar to the while statement except that it evaluates its conditional expression after executing its statement.

```
do <statement> while ( <exp> ) ;
```

First `<statement>` is executed. Then the Boolean valued expression in `<exp>` is evaluated. If it evaluates to `FALSE`, the do-while statement is terminated. otherwise the process is repeated. Hence the do-while statement always executes its body at least once.

```
var i;
function Done(){
  return i>=10;
}
function Increment(){
  i=i+1;
}
function DoSomething(){
  Out("Hi ",Eval(i));
}
i=1;
do {
  DoSomething();
  Increment();
} while (!Done());
```

8.6.11. break Statement

The `break` statement is used to exit out of the innermost loop in a `for`, `while`, `do-while` or `switch` statement.

Example:

```
var i;
function Initialize(){
  i=1;
}
function Done(){
```

```
  return i>=10;
}
function Increment(){
  i=i+1;
}
function DoSomething(){
  Out("Hi ",Eval(i));
}
for(Initialize();!Done();Increment()) {
  if(i==5) break;
  DoSomething();
}
Out("I'm done.");
```

Output:

```
Hi 1
Hi 2
Hi 3
Hi 4
I'm done.
```

8.6.12. continue Statement

The `continue` statement is used to jump to the end of the loop in a `for`, while or do-while statement.

Example:

```
var i;
function Initialize(){
  i=1;
}
function Done(){
  return i>=10;
}
function Increment(){
  i=i+1;
}
function DoSomething(){
  Out("Hi ",Eval(i));
}
for(Initialize();!Done();Increment()) {
  if(i==5) continue;
  DoSomething();
}
Out("I'm done.");
```

Output:

```
Hi 1
Hi 2
Hi 3
Hi 4
Hi 6
Hi 7
Hi 8
Hi 9
Hi 10
I'm done.
```

8.6.13. switch Statement

The `switch` statement is used to selectively execute a part of a compound statement. It takes the form:

```
switch ( <exp> ) <compound_stmt>
```

The simple expression `<exp>` is evaluated first. The value of this expression is then compared with the values of expressions in case statements that occur in the compound statement. A case statement is of the form:

```
case <exp> : <statement>
```

If the value matches that of a particular case, then execution jumps to the statement part of that case.

```
var x=2;
switch(x) {
case 1:
  Out("One");
  break;
case 2:
  Out("Two");
  break;
case 3:
  Out("Three");
  break;
}
```

Otherwise, execution jumps to a default statement, which is of the form:

```
default : <statement>
var x=4;
switch(x) {
case 1:
```

```
  Out("One");
  break;
case 2:
  Out("Two");
  break;
case 3:
  Out("Three");
  break;
default:
  Out("Unknown");
}
```

If the expression does not match that of any case, and no default statement is found in the compound statement, then a runtime error is generated. Example:

```
for(var month=1; month <=12; month =month+1) {
  var quarter;
  switch(month) {
  case 1:
  case 2:
  case 3:
    quarter="Winter";
    break;
  case 4:
  case 5:
  case 6:
    quarter="Spring";
    break;
  case 7:
  case 8:
  case 9:
    quarter="Summer";
    break;
  case 10:
  case 11:
  case 12:
    quarter="Fall";
    break;
  default:
    quarter="Invalid";
  }
  Out("Month:",Eval(month),", Quarter=",Eval(quarter));
}
```

Output:

```
Month:1, Quarter=Winter
```

```
Month:2, Quarter=Winter
Month:3, Quarter=Winter
Month:4, Quarter=Spring
Month:5, Quarter=Spring
Month:6, Quarter=Spring
Month:7, Quarter=Summer
Month:8, Quarter=Summer
Month:9, Quarter=Summer
Month:10, Quarter=Fall
Month:11, Quarter=Fall
Month:12, Quarter=Fall
```

8.6.14. Variable Scope (Visibility) Rules

A variable may be declared using a declaration statement anywhere in a program. A variable that is declared within a compound statement is visible to all subsequent statements within that compound statement. It is not visible to statements that precede the declaration, or to statements that are outside of the compound statement.

```
// The variable x is not visible to statements here.
{
  // The variable x is not visible to statements here.
  var x;
  // the variable x is visible to all statements here.
}
// The variable x is not visible to statements here.
```

A variable can be declared only once in a compound statement.

```
{
  var x=1;
  var x=2; //This will generate an Error!
  Out("x=",Eval(x));
}
```

However, it is possible that a variable with the same name might have been declared in an outer context. In such a case, the variable declared in the innermost context is the only one visible.

```
var x=1;
{
  var x=2;
  Out("x is = ",Eval(x));
}
```

Output:

```
x is = 2
```

Variables declared in the initialization part of a `for` statement will be visible only within the `for` statement:

```
var y=Dim(10);
for(var j=1;j<=10;j=j+1) {
  var k=11-j
  y[k]=j*j;
}
j=10;  //This an Error. j is not visible here.
k=1;   // This is also an Error.
```

It is good programming practice to limit the scope of variables to as small a part of the code as possible. For example, in the following piece of code, a programming error will cause an infinite loop that is difficult to debug. No error message is generated because there is no syntax error:

```
var x1=Dim(10);
var x2=Dim(20);

var i1,i2;
for(i1=1;i1<=10;i1=i1+1) {
  x1[i1]=1.0;
}
for(i2=1;i2<=20;i1=i2+1) {
  x2[i2]=1.0;
}
```

But if the scope of the loop variables had been limited, as in the following piece of code, then the problem would be caught right away, and an error message will be generated:

```
var x1=Dim(10);
var x2=Dim(20);

for(var i1=1;i1<=10;i1=i1+1) {
  x1[i1]=1.0;
}
for(var i2=1;i2<=20;i1=i2+1) {
  x2[i2]=1.0;
}
```

8.6.15. Function Definition

Functions may be defined anywhere using a declaration of the form:
`function IDENTIFIER ( <parameterlist> ) <statement>`

The identifier IDENTIFIER is the name assigned to the function. `<statement>` is a simple or compound statement that forms the body of the function. Here `<parameterlist>` is a list of zero or more parameter declarations separated by commas. Each parameter may be an in parameter, an out parameter, or an inout parameter. Accordingly, the parameter declarations are of the form:

```
in IDENTIFIER
out IDENTIFIER
inout IDENTIFIER
```

Here IDENTIFIER is the name of the parameter. The value of an `in` parameter cannot be modified in the body of the function. The value of an `out` variable is initially undefined, and must be set in the function body. The value of an `inout` variable is already defined, and may also be modified in the function body. The return statement is used to transfer control out of a function. The return statement can optionally return a value to the calling routine. The return statement is of one of the following two forms:

```
return ;
return <exp>;
```

The first form of the return statement returns the value 0, and the second form returns a copy of the expression `<exp>` Example:

```
function Factorial(in n){
  if(n==1) {
    return 1;
  } else {
    return Eval(n*Factorial(n-1));
  }
}

Out(Factorial(4));
Out(Factorial(5));
```

Output:

```
24
120
```

Example:

```
function Binary(in n){
  if(n==1) {
    return "1";
  } else if(n==0) {
    return "0";
  } else {
    var n1=int(n/2);
    var n2=n-2*n1;
    return Eval(Binary(n1)+Binary(n2));
  }
```

```
}

Out(Binary(3));
Out(Binary(13));
Out(Binary(6876));
```

Output:

```
11
1101
1101011011100
```

Like variables, a function once declared inside a compound statement, is visible to all subsequent statements inside that compound statement. It is not visible to statements outside that compound statement. Example:

```
// Sample routine to expand the function f into its Taylor series
// of order n, in variable x, about its current value.
function Taylor(in f, in x, in n){
  // define a local function:
  function Factorial(in n){
    if(n==1) {
      return 1;
    } else {
      return Eval(n*Factorial(n-1));
    }
  }
  // The first term of the series:
  var lastder:=Subst(f);
  var TaylorSeries=f;
  for(var i=1;i<=n;i=i+1) {
    // Obtain the ith order derivative of f wrt x:
    lastder:=TotalDiff(Subst(lastder),Subst(x));
    // If this derivative is non-zero at the current value of x:
    if(Eval(lastder)!=0) {
      TaylorSeries:=Subst(TaylorSeries)+
          Eval(lastder)*(Subst(x)-Eval(x))^Eval(i)/
          Factorial(i);
    }
  }
  return Subst(TaylorSeries);
}

var theta=0;
Out(Taylor(sin(theta),theta,2));
Out(Taylor(sin(theta),theta,4));
Out(Taylor(cos(theta),theta,4));
```

```
var t=1;
Out(Taylor(t^0.5,t,3));
Output:
(0+((1*((theta-0)^1))/1))
((0+((1*((theta-0)^1))/1))+((-1*((theta-0)^3))/6))
((1+((-1*((theta-0)^2))/2))+((1*((theta-0)^4))/24))
(((1+((0.5*((t-1)^1))/1))+
   ((-0.25*((t-1)^2))/2))+((0.375*((t-1)^3))/6))
```

Scope of returned values: When a function returns an expression, the expression should not be dependent on any variable that is not visible to the calling routine.

```
function g(){
  var n=1;
  return Eval(n);
}

function f(){
  var n=1;
  return n;
}

Out(Eval(g()));  // This is OK
Out(Eval(f()));  // This will generate an error. the
                 // variable n is not visible here
```

Functions may be defined in a separate file. The name of the file should be the same as that of the function, with the extension '.clx'. Example: Contents of file hex.clx

```
function hex(in x){
  var Table=["0","1","2","3","4","5","6","7","8","9",
             "A","B","C","D","E","F"];
  if(x<0) throw "Invalid input to hex() routine.";
  if(x-int(x)!=0) throw "non-integer input to hex() routine.";
  if(x<16) {
    return Eval(Table[x+1]);
  } else {
    var n1=int(x/16);
    var n2=x-16*n1;
    return Eval(hex(n1)+hex(n2));
  }
}
```

Input to *Coustyx*:

```
Out(hex(3));
Out(hex(13));
Out(hex(6876));
```

Output:

```
3
D
1ADC
```

8.6.16. try-catch and throw Statements

The `try-catch` and `throw` statements form a exception handling mechanism. They offer an alternative to `goto` statements when trying to gracefully recover from an error. A `try-catch` statement is of one of the two following forms:

```
try <statement1> catch () <statement2>
try <statement1> catch ( IDENTIFIER ) <statement2>
```

The statement `<statement1>` is executed during the normal course of execution. If while executing this part of code, an exception is thrown, then control gets transferred to the statement `<statement2>`. The exception may be thrown anywhere inside `<statement1>`, or even inside functions called by it. These exceptions are thrown by the throw statement which takes one of the following two forms:

```
throw ;
throw <exp> ;
```

here `<exp>` is any expression. Just like in the return statement, this expression must not be dependent on any variables not visible to the routine that will 'catch' this thrown exception. When this statement is executed, control is transferred to the `<statement2>` part of the innermost try-catch statement that encloses this throw statement. The IDENTIFIER in the try-catch statement is assigned the expression `<exp>`. Example:

```
function Factorial(in n){
  if(n<0) throw "Factorial of negative number attempted.";
  if(n>75) throw "Overflow.";
  if(n==1) {
    return 1;
  } else {
    return Eval(n*Factorial(n-1));
  }
}

try{
  Out("Factorial(4)=",Factorial(4));
  Out("Factorial(5)=",Factorial(5));
  Out("Factorial(4.5)=",Factorial(4.5));
} catch (message) {
  Out(Eval("Error:"+message));
}
```

Output:

```
Factorial(4)=24
Factorial(5)=120
Error:Factorial of negative number attempted.
```

If there is no `try-catch` statement that can catch an exception, then the exception will be passed on to *Coustyx*, which will handle it in the same way it handles its internally generated error messages. The exception can also be re-thrown after being caught.

```
try{
  Out("Factorial(4)=",Factorial(4));
  Out("Factorial(5)=",Factorial(5));
  Out("Factorial(4.5)=",Factorial(4.5));
} catch (message) {
  Out(Eval("Error:"+message));
  throw(Subst(message));
}
```

8.6.17. Statement Label

Any statement can be assigned a label or IDENTIFIER as follows. This label is visible to all statements contained in the compound statement of which this statement is a part. Unlike variables and functions, this label is also visible to statements preceding the label, provided they are contained in the same compound statement. IDENTIFIER : `<statement>`

8.6.18. Goto Statement

A `goto` statement transfers control to a statement that has been labeled as shown above. The goto statement can be used to transfer control to any labeled statement, as long as the label is visible. However, the `goto` statement cannot be used to transfer control out of a function body: `goto IDENTIFIER ;`

Example:

```
{
  var x=1;
  var y=2;
  if(x==0) goto exitlabel;
  var z=3;
exitlabel:
  Out("Done");
}
```

Example:

```
{
  var x=1;
  var y=2;
```

```
  if(x==0) goto exitlabel; // This is not correct. The label
                           // exitlabel is not visible from here.
  var z=3;
  {
    var z1=2;
  exitlabel:
    Out("Done");
  }
}
```

Example:

```
{
  var x=1;
  {
    var y=2;
    if(x==0) goto exitlabel;  // This is OK
  }
  var z=3;
exitlabel:
  Out("Done");
}
```

8.7. End of Input

The input Session is terminated by an end of file marker, or by the special symbol: `End`

8.8. Grammar

The syntactical elements of the language are formally specified by the rules described in Table 8.1 and Table 8.2.

8.9. Regular Functions Syntax

Some of the functions that are stored in symbolic form and can be differentiated are listed below. You can use these functions in *Coustyx* scripts.

Sin or sin Obtain sine of an angle. The angle is assumed to be in radians. Example: Sin($\pi/2$)=1.

Cos or cos Obtain cosine of an angle. The angle is assumed to be in radians. Example: Cos(π)=-1.

Table 8.1: Coustyx Language Grammar

Non-Terminal Symbol		Expansion
$\epsilon \rightarrow$		(empty)
input $\rightarrow$		ϵ stmt_list_toplevel
stmt_list_toplevel $\rightarrow$		ϵ
	|	stmt_list_toplevel statement
compound_stmt $\rightarrow$		{ stmt_list }
stmt_list $\rightarrow$		ϵ
	|	stmt_list statement
statement $\rightarrow$		assign_decl_exp ;
	|	compound_stmt
	|	function_defn
	|	`if` (exp)statement
	|	`if` (exp)statement else statement
	|	`break` ;
	|	`continue` ;
	|	`return` ;
	|	`return` exp ;
	|	`for` (assign_decl_exp ; exp ; assign_exp) statement
	|	`while` (exp) statement
	|	`do` statement `while` (exp) ;
	|	`switch` (exp) compound_stmt
	|	`case` exp : statement
	|	`goto` IDENTIFIER ;
	|	IDENTIFIER : statement
	|	`default` : statement
	|	`throw` ;
	|	`throw` exp ;
	|	`try` statement `catch` () statement
	|	`try` statement `catch` (IDENTIFIER) statement
function_defn $\rightarrow$		`function` IDENTIFIER (parameter_list) statement
parameter_list $\rightarrow$		ϵ
	|	parameter_list_1
parameter_list_1 $\rightarrow$		parameter
	|	parameter_list_1 , parameter
parameter $\rightarrow$		`in` IDENTIFIER
	|	`out` IDENTIFIER
	|	inout IDENTIFIER
assign_decl_exp $\rightarrow$		assign_exp
	|	decl_list
decl_list $\rightarrow$		var identdecl_list
identdecl_list $\rightarrow$		identdecl_list , identdecl
	|	identdecl

Table 8.2: Coustyx Language Grammar (contd.)

<table>
<tr><th>Non-Terminal Symbol</th><th></th><th>Expansion</th></tr>
<tr><td>identdecl →</td><td></td><td>IDENTIFIER</td></tr>
<tr><td></td><td>|</td><td>IDENTIFIER = exp</td></tr>
<tr><td></td><td>|</td><td>IDENTIFIER := exp</td></tr>
<tr><td>assign_exp →</td><td></td><td>ϵ</td></tr>
<tr><td></td><td>|</td><td>addr = exp</td></tr>
<tr><td></td><td>|</td><td>addr := exp</td></tr>
<tr><td></td><td>|</td><td>exp</td></tr>
<tr><td>addr →</td><td></td><td>addr [exp]</td></tr>
<tr><td></td><td>|</td><td>IDENTIFIER</td></tr>
<tr><td>exp →</td><td></td><td>or_list</td></tr>
<tr><td>or_list →</td><td></td><td>or_list | and_list</td></tr>
<tr><td></td><td>|</td><td>and_list</td></tr>
<tr><td>and_list →</td><td></td><td>and_list & binary</td></tr>
<tr><td></td><td>|</td><td>binary</td></tr>
<tr><td>binary →</td><td></td><td>binary relop binary</td></tr>
<tr><td></td><td>|</td><td>binary + binary</td></tr>
<tr><td></td><td>|</td><td>binary - binary</td></tr>
<tr><td></td><td>|</td><td>binary * binary</td></tr>
<tr><td></td><td>|</td><td>binary / binary</td></tr>
<tr><td></td><td>|</td><td>binary ^ binary</td></tr>
<tr><td></td><td>|</td><td>unary</td></tr>
<tr><td>relop →</td><td></td><td><</td></tr>
<tr><td></td><td>|</td><td><=</td></tr>
<tr><td></td><td>|</td><td>></td></tr>
<tr><td></td><td>|</td><td>>=</td></tr>
<tr><td></td><td>|</td><td>==</td></tr>
<tr><td></td><td>|</td><td>!=</td></tr>
<tr><td>unary →</td><td></td><td>addr</td></tr>
<tr><td></td><td>|</td><td>aoterm</td></tr>
<tr><td></td><td>|</td><td>[list]</td></tr>
<tr><td></td><td>|</td><td>+ unary</td></tr>
<tr><td></td><td>|</td><td>- unary</td></tr>
<tr><td></td><td>|</td><td>! unary</td></tr>
<tr><td></td><td>|</td><td>floating_point_constant</td></tr>
<tr><td></td><td>|</td><td>string_constant</td></tr>
<tr><td></td><td>|</td><td>TRUE</td></tr>
<tr><td></td><td>|</td><td>FALSE</td></tr>
<tr><td>aoterm →</td><td></td><td>aoterm [exp]</td></tr>
<tr><td></td><td>|</td><td>IDENTIFIER (list)</td></tr>
<tr><td></td><td>|</td><td>(exp)</td></tr>
<tr><td>list →</td><td></td><td>ϵ</td></tr>
<tr><td></td><td>|</td><td>list_1</td></tr>
<tr><td>list_1 →</td><td></td><td>list_1 ; exp</td></tr>
<tr><td></td><td>|</td><td>exp</td></tr>
</table>

Tan or tan Obtain tangent of an angle. The angle is assumed to be in radians. Example: Tan($\pi/4$)=1.

Asin or asin Obtain the arc sine (inverse function of sine) of a number between -1 and +1. The output is in radians. Example: Asin(0.707107)=0.785398 (=$\pi/4$).

Acos or acos Obtain the arc cosine (inverse function of cosine) of a number between -1 and +1. The output is in radians. Example: Acos(0.707107)=0.785398 (=$\pi/4$).

Atan or atan Obtain the (2-quadrant) arc tangent (inverse function of tangent) of a number. The output is in radians. Example: Atan(1)=0.785398 (=$\pi/4$).

atan2(y,x) Obtain the 4-quadrant arc tangent of the real arguments x and y. $-\pi \leq \mathbf{atan2(y,x)} \leq \pi$.

Exp or exp Obtain the exponential of a number. Example: Exp(1)=2.71828 (=e^1).

Ln or ln Obtain logarithm to the base e (natural logarithms). Example: Ln(10)=2.30259, Ln(exp(1))=1.

int Obtain the integer value of a number. Example: int(4.926)=4, int(-4.926)=4.

abs Obtain the absolute value of a real number or the magnitude of a complex number. Example: abs(-10.958)=10.958, abs(-3+4*i)=5.

mag2(x,y) Obtains the magnitude $\sqrt{x^2+y^2}$ for real arguments x and y. Example: mag2(3,-4)=5.

real, imag Obtain real and imaginary values of a complex number respectively. Example: real(3+4*i)=3, imag(3+4*i)=4.

8.10. Special Functions Syntax

Given below is the syntax for calling special functions that are available in *Coustyx*. These functions are evaluated numerically. Use these functions in *Coustyx* scripts.

8.10.1. Associated Legendre Function

Plm(l, m1, m2, z, result)

```
Description:
  Computes Associated Legendre functions of degree l and order m (not normalized)
Inputs:
  integer: l - degree
  integer: m1 - begin order
  integer: m2 - end order
  double:  z - argument (must be in [-1,1])
Output:
```

```
  double array: result
  result contains Plm(z) for degree l, and all orders [m1,m2]
```

plm(l, m1, m2, z, result)

```
Description:
  Computes Associated Legendre functions of degree l and order m (normalized)
Inputs:
  integer: l - degree
  integer: m1 - begin order
  integer: m2 - end order
  double:  z - argument (must be in [-1,1])
Output:
  double array: result
  result contains plm(z) for degree l, and all orders [m1,m2]
```

8.10.2. Legendre Polynomials

LegendreP(maxorder,x,result)

```
Description:
  Computes array of Legendre Polynomials
Inputs:
  integer: maxorder - degree of the Legendre Polynomial
      All degrees from [0,maxorder] will be computed.
  double: x - argument to the Legendre Polynomial (must be in [-1,1])
Output:
  double array: result
  result contains Pn(z) for degrees n in [0,maxorder]
```

8.10.3. Spherical Harmonic

Ylm(l, m1, m2, theta, phi, result)

```
Description:
  Computes Spherical Harmonics of degree l and order m
Inputs:
  integer: l - degree
  integer: m1 - begin order
  integer: m2 - end order
  double: theta - theta in Spherical polar coordinates [0,PI]
  double: phi - phi in Spherical polar coordinates [0,2*PI], any branch
               is fine.
Output:
  complex array: result
  result contains Ylm(theta,phi) for degree l, and all orders [m1,m2]
```

8.10.4. Spherical Hankel Function of First Kind

h1n(n, nterms, z, result)

```
Description:
  Computes Spherical Hankel function of the first kind and order n, n+1.. n+(nterms-1)
Inputs:
  integer: n - order of the Spherical Hankel function
  integer: nterms - number of terms to be computed.
               All orders from [n, n+1, ...n+nterms-1] will be
               computed.
  complex: z - argument to the Spherical Hankel function
Output:
  complex array: result contains h1n(z)
```

8.10.5. Spherical Hankel Function of Second Kind

h2n(n, nterms, z, result)

```
Description:
  Computes Spherical Hankel function of the second kind and order n
Inputs:
  integer: n - order of the Spherical Hankel function
  integer: nterms - number of terms to be computed.
               All orders from [n, n+1, ...n+nterms-1] will be
               computed.
  complex: z - argument to the Spherical Hankel function
Output:
  complex array: result contains h2n(z)
```

8.10.6. Cylindrical Hankel Function of First Kind

H1n(n, nterms, z, result)

```
Description:
  Computes Cylindrical Hankel function of first kind, integer order n
Inputs:
  integer: n - order of the Cylindrical Hankel function
          For integer n, n can be +ve, 0 or -ve
  integer: nterms - number of terms to be computed.
               All orders from [n, n+1, ...n+nterms-1] will be
               computed.
  complex: z - argument to the Cylindrical Hankel function
Output:
```

```
  complex array: result contains H1n(z)
```

H1nu(n, nterms, z, result)

```
Description:
  Computes Cylindrical Hankel function of first kind, fractional order n
Inputs:
  integer: n - order of the Cylindrical Hankel function
          For fractional n, n must be +ve
  integer: nterms - number of terms to be computed.
               All orders from [n, n+1, ...n+nterms-1] will be
               computed.
  complex: z - argument to the Cylindrical Hankel function
Output:
  complex array: result contains H1nu(z)
```

8.10.7. Cylindrical Hankel Function of Second Kind

H2n(n, nterms, z, result)

```
Description:
  Computes Cylindrical Hankel function of second kind, integer order n
Inputs:
  integer: n - order of the Cylindrical Hankel Function
          For integer n, n can be +ve, 0 or -ve
  integer: nterms - number of terms to be computed.
               All orders from [n, n+1, ...n+nterms-1] will be
               computed.
  complex: z - argument to the Cylindrical Hankel function
Output:
  complex array result contains H2n(z)
```

H2nu(n, nterms, z, result)

```
Description:
  Computes Cylindrical Hankel function of second kind, fractional order n
Inputs:
  integer: n - order of the Cylindrical Hankel Function
          For fractional n, n must be +ve
  integer: nterms - number of terms to be computed.
               All orders from [n, n+1, ...n+nterms-1] will be
               computed.
  complex: z - argument to the Cylindrical Hankel function
Output:
  complex array result contains H2nu(z)
```

8.10.8. Spherical Bessel Function

jn(n, nterms, z, result)

```
Description:
  Computes Spherical Bessel function of order n
Inputs:
  integer: n - order of the Spherical Bessel function
  integer: nterms - number of terms to be computed.
               All orders from [n, n+1, ...n+nterms-1] will be
               computed.
  complex: z - argument to the Spherical Bessel function
Output:
  complex array: result contains jn(z)
```

8.10.9. Cylindrical Bessel Function

Jn(n, nterms, z, result)

```
Description:
  Computes Cylindrical Bessel function of integer order n
Inputs:
  integer: n - order of the Cylindrical Bessel function
          For integer n, n can be +ve, 0 or -ve
  integer: nterms - number of terms to be computed.
               All orders from [n, n+1, ...n+nterms-1] will be
               computed.
  complex: z - argument to the Cylindrical Bessel function
Output:
  complex array: result contains Jn(z)
```

Jnu(n, nterms, z, result)

```
Description:
  Computes Cylindrical Bessel function of fractional order n
Inputs:
  integer: n - order of the Cylindrical Bessel function
          For fractional n, n must be +ve
  integer: nterms - number of terms to be computed.
               All orders from [n, n+1, ...n+nterms-1] will be
               computed.
  complex: z - argument to the Cylindrical Bessel function
Output:
  complex array: result contains Jnu(z)
```

Chapter 9

Tutorial – Gear Box Radiation

This tutorial is created to outline the steps required to compute radiated noise from a gearbox housing. Detailed steps are given on how to create and perform acoustic analysis for 'Multidomain Model', and 'Indirect Model'.

9.1. Introduction

Follow the procedure outlined below for general noise radiation prediction using Coustyx software. The two steps involved in the prediction of noise radiated by a gearbox housing are: (a) determination of housing vibration by experiments or analysis, here we do finite element analysis; (b) prediction of radiated noise based on the vibration using Coustyx (boundary element method).

FEA The Finite Element Analysis (FEA) is used to build the mesh and to estimate the structural vibration, which is used as the input velocity for acoustic analysis in Coustyx.

1. Build a finite element model of the gearbox housing from the geometry. Compute mode shapes and natural frequencies of the structure from FEA modal analysis.
2. Estimate the forces transmitted to the gearbox housing through bearings or from other interactions with the components in a gearbox. Transform the forces into the frequency domain to obtain force amplitudes as a function of frequency.

   Generally, the forces on the gearbox housings are from the gear mesh excitation which are transmitted through the bearings. The time domain bearing loads (forces and moments) can be computed from a Calyx analysis or any other similar analysis. The bearing loads can then be transformed into the frequency domain to obtain the bearing load amplitudes as a function of frequency.
3. Compute the housing structural response at each frequency of analysis using a FEA software. The gearbox housing frequency response is computed by modal superposition.

Coustyx Acoustic radiation problem is then solved using Coustyx by importing the FEA structure mesh and loading the structural response for the gearbox housing.

1. Build a new Coustyx model by importing the FEA structure mesh.
2. Load the frequency response data from the FEA analysis into the Coustyx model. This response is used as the velocity boundary condition for the acoustic analysis.
3. Set the Coustyx analysis parameters and run acoustic analysis to compute acoustic metrics such as radiated sound power, radiation efficiency, pressure levels at field points, etc., at each frequency.

9.2. Problem Description

The example gearbox housing analyzed in this tutorial is from a gear noise test rig developed at NASA Lewis Research Center [1]. The details of the gear box are shown in Figure 9.1. The

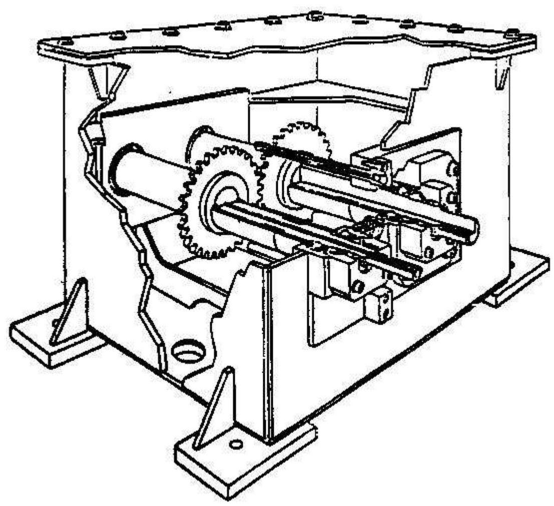


Figure 9.1: Detail of the NASA Lewis gearbox.

structure is a rectangular box of overall dimensions 0.2794 m x 0.3048 m x 0.29845 m, or 11 inches x 12 inches x 11.75 inches. The top plate is made of 1.588 mm or 1/16-inch aluminum and the other five surfaces are made of 12.7 mm or 1/2-inch thick steel. Note that all the dimensions considered in this tutorial are in S.I units. The four corners of the bottom surface are clamped rigidly to the ground. The housing supports two shafts through bearings, which are mounted on the four holes in the structure.

To simulate the forces on the gearbox housing transmitted from the gears through shafts and bearings, an in-plane force F_y of unit amplitude in frequency domain is applied to one of the shafts and an equal but opposite force is applied on the second shaft. The direction y is assumed to be along the line connecting the centers of the two shafts. The noise radiation from the resulting housing vibration is computed in the frequency range 100–1000 Hz.

9.3. Finite Element Analysis

The finite element analysis (FEA) of the gearbox housing is required prior to solving the acoustic radiation problem for the following reasons:

- The gearbox structure mesh generated in FEA is imported into Coustyx to build the model to solve the acoustic problem.
- The structural response due to forces on the housing at each frequency in the frequency domain is loaded into the Coustyx model. These values are used as the input velocity boundary condition.

Coustyx can read FEA data from NASTRAN, ABAQUS, and ANSYS softwares.

9.3.1. FE Mesh Modeling

In the FE model, plate elements are used to discretize the box surfaces. The shafts are connected to the housing using rigid elements. A concentrated mass of 10 grams is attached to center of the top plate to account for the mass of the stinger and the moving part of the shaker.

The FE model used for the structural analysis shown in Figure 9.2 has 974 nodes, 943 quadrilateral plate elements, 1 concentrated mass, and 4 rigid elements.

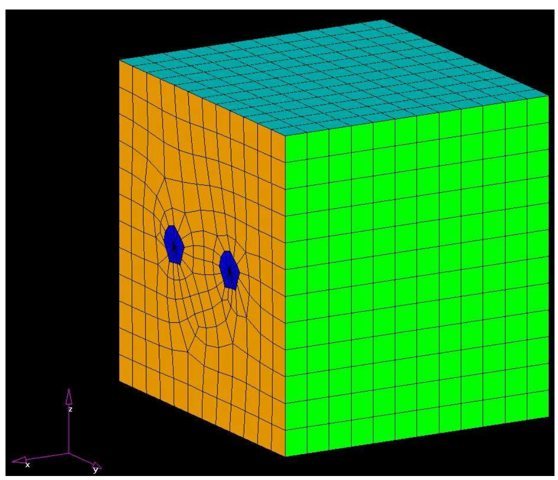


Figure 9.2: FE model of the NASA Lewis gearbox.

The physical properties of the materials used in FEA analysis are given in Table 9.1.

9.3.2. Natural Frequencies and Normal Modes

The natural frequencies and mode shapes are extracted by solving the FE eigenvalue problem of the gearbox housing. The eigenvectors are normalized with respect to the mass matrix to get the structure normal modes. Since the top plate is relatively more flexible compared to

Table 9.1: Physical properties of materials used in FEA.

Material	Elastic Modulus, Pa	Shear Modulus, Pa	Poisson's Ratio	Density, kg/m^3
Steel	2.034E+11	7.823E+10	0.3	7850
Aluminum	7.31E+10	2.748E+10	0.33	2700

the remaining five surfaces, several free vibration mode shapes of the gearbox housing resemble classic plate modes of the top surface. Mode 1 is a (1,1) mode of the top plate. Similarly Mode 2 is a (1,2) plate mode, Mode 3 is a (2,1) plate mode, Mode 4 is a (2,2) plate mode and Mode 5 is a (1,3) plate mode respectively. In a (m, n) plate mode m and n represent the number of half-wave lengths along the x and y directions respectively.

Table 9.2: Gearbox natural frequencies from FEA.

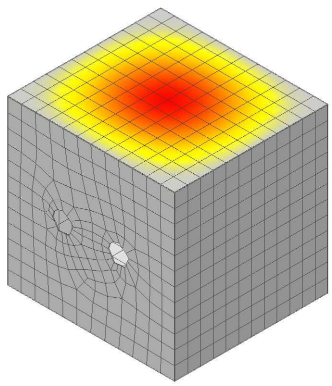
Mode Number	Frequency, Hz
1	157
2	329
3	366
4	505
5	543
6	647
7	741
8	760
9	789
10	797
11	829
12	907
13	936
14	962

Table 9.2 lists some of the natural frequencies of the gearbox housing. The first five modes are purely classical modes of the top plate. For some of the higher modes, the side and the bottom surfaces of the gearbox undergo deformation as well. Figure 9.3 shows the first five free vibration modes of the gearbox housing.

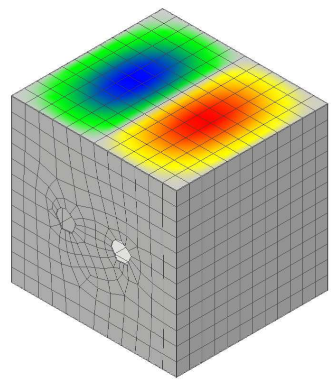
9.3.3. Forced Response - Modal Superposition

Forced response analysis is performed using modal superposition in the frequency range of 100–1000 Hz. In modal superposition, FEA computes the modal basis and a subset of the computed modes are used to compute the forced response via superposition.

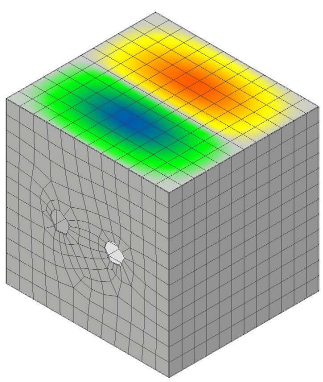
The loads applied on the housing are: an in-plane force of unit amplitude in frequency domain is applied in $+y$ direction to one of the shafts and an equal but opposite force is applied to the



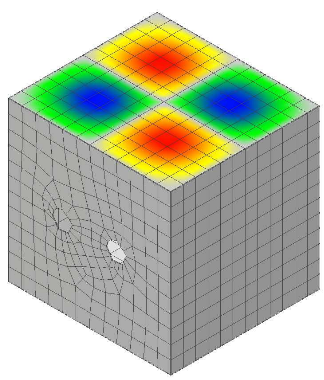
(a) Mode 1: 157 Hz



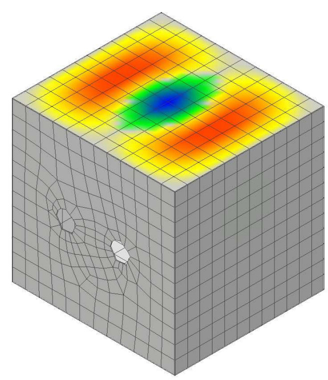
(b) Mode 2: 329 Hz



(c) Mode 3: 366 Hz



(d) Mode 4: 505 Hz



(e) Mode 5: 543 Hz

Figure 9.3: Surface normal velocity distribution for the first five free vibration modes.

other. Note that $+y$ is in the direction of a line connecting the centers of both the shafts. The radiation from the resulting housing vibration is computed in the frequency range 100–1000 Hz with a frequency resolution of 15 Hz.

The output punch file contains the nodal velocities or accelerations at each frequency. This data is loaded into Coustyx model and is later applied as the velocity boundary condition to predict noise generated by the gearbox housing in this frequency range.

9.4. Coustyx MultiDomain Model

Coustyx MultiDomain model is created by importing the FE mesh. The frequency response data from the FEA analysis is loaded into Coustyx and is applied as a structure velocity boundary condition on the gearbox housing. The analysis parameters are then set and the acoustic analysis is run to compute radiation predictions.

9.4.1. Problem Setup

Follow the steps to setup Coustyx model and perform acoustic analysis on the gearbox housing. Open Coustyx from the start menu of your computer.

9.4.1.1. Create a New Model

- In the main menu select: **File** → **New Model**. The window in Figure 9.4 will then appear.
- Choose the model type: 'Multidomain Model' and select model units: meter-kilogram-second (SI units). Note that the selection of model units is consistent with the unit of length in the structure mesh. Click 'OK' to proceed.

9.4.1.2. Import FE Structure Mesh

- In Coustyx model main menu select: **Model** → **Structures**.
- Right–click on **Structures** and select: **Import** → **Nastran Bulk Data (.bdf) File...** (as shown in Figure 9.5) to import mesh from Nastran bulk data format. The FE meshes from Abaqus and Ansys data formats can be imported by selecting **Abaqus (.inp) File** or **Ansys Results (.rst) Files** respectively.
- Select the appropriate FE structure data file to be imported from the browser and click 'Open'.

9.4.1.3. Load Frequency Response Data

- In Coustyx model main menu select: **Model** → **Structures** → **Structmesh_0** or *<Struct Mesh Name>*.
- Right-click on **Structmesh_0** or *<Struct Mesh Name>* and select: **Load Freq Response Data** → **Nastran Punch File...** (as shown in Figure 9.6). The other valid data formats from which frequency response data can be loaded into Coustyx are **Nastran OP2 File** and **Ansys rst File**.

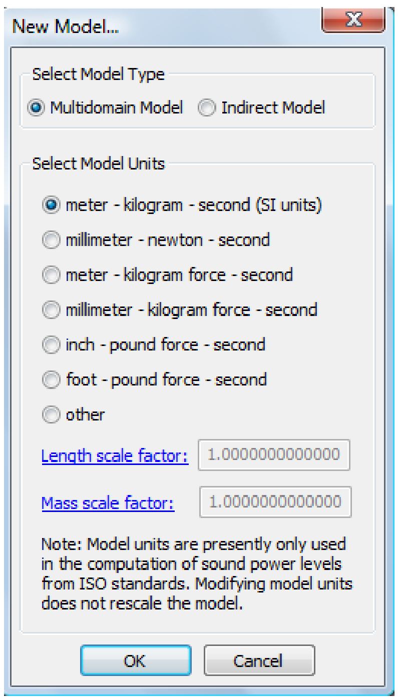


Figure 9.4: New Model Selection Window.

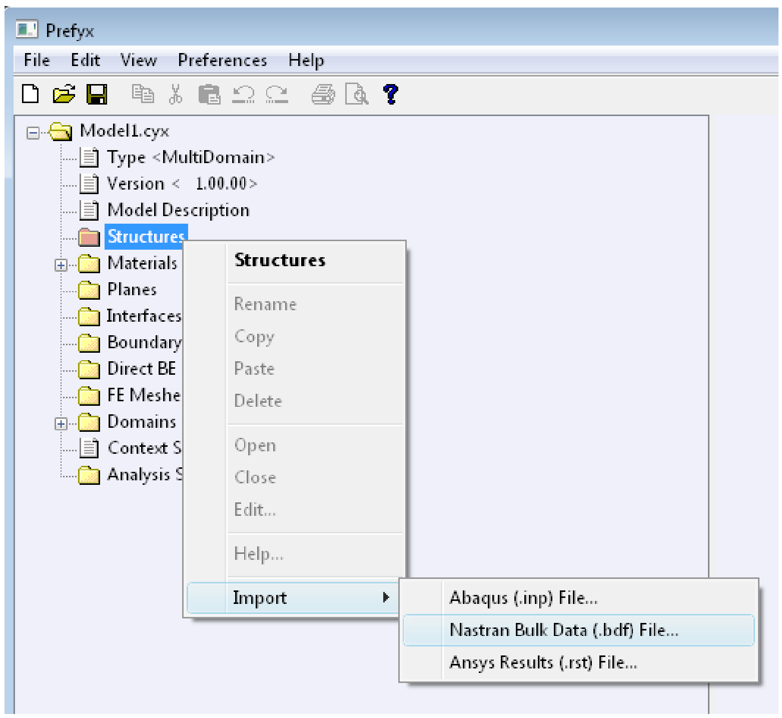


Figure 9.5: Import a finite element structure mesh.

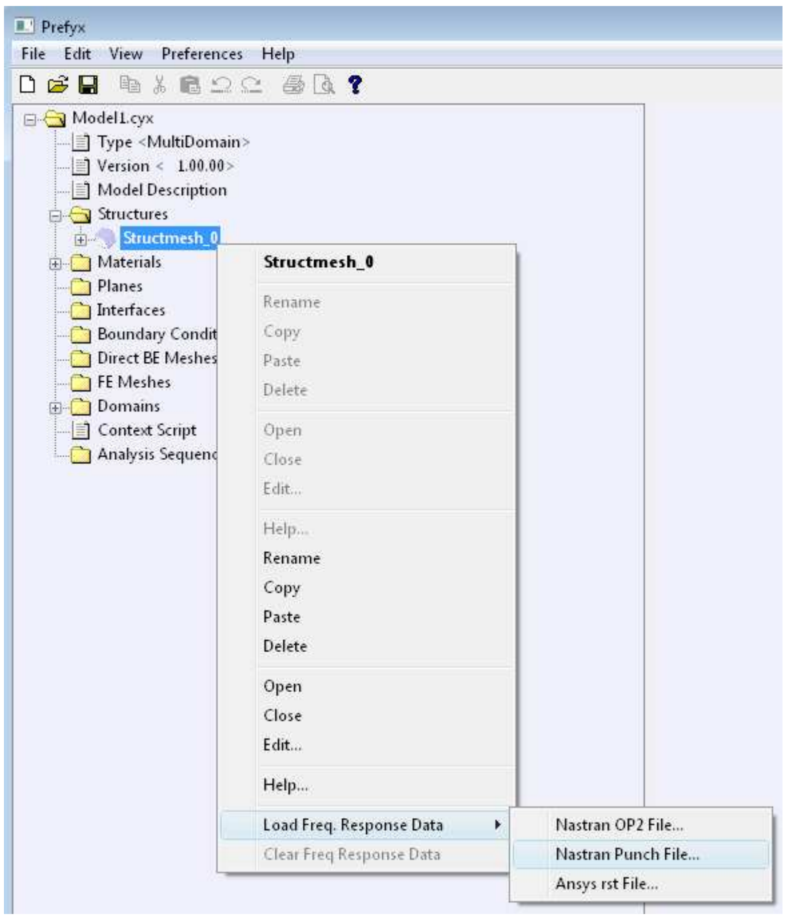


Figure 9.6: Load frequency response data into Coustyx.

- Select the appropriate frequency response data file to be loaded from the browser and click ‘Open’.

9.4.1.4. Generate BE Mesh

- Select: **Model** → **Structures** → **Structmesh_0** or <*Struct Mesh Name*>.
- Right-click on **Structmesh_0** or <*Struct Mesh Name*> and select: **Open** to view the structure mesh in the GUI. The structure mesh will appear as shown in Figure 9.7.

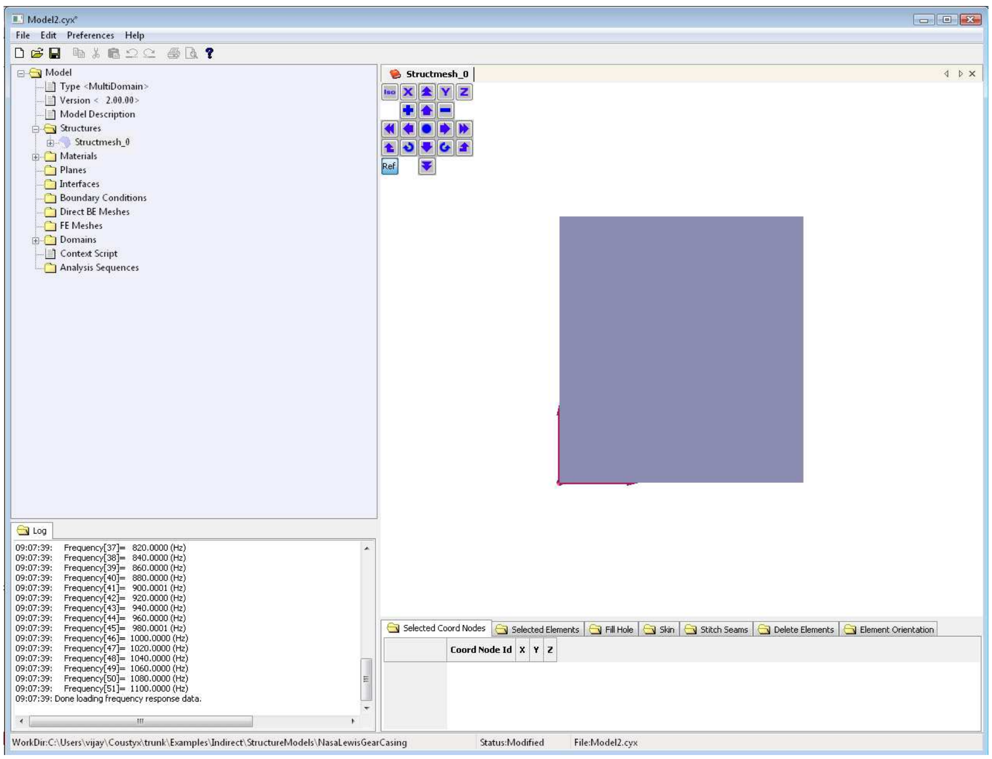


Figure 9.7: Structure mesh opened in Coustyx GUI.

- Move the cursor into the GUI window with the structure mesh and observe the cursor change to *move-cursor* style (or to the shape of ‘+’). To manipulate the view:
  - Use the GUI control panel tools shown in Figure 9.8 to *zoom* and *rotate* the model.
  - Hold down the left-click button and move the mouse to rotate the model in the GUI.
  - Hold down the right-click button and move the mouse to move the model in the GUI.
  - Move and rotate the model to see the holes on one of the side surfaces on the structure.

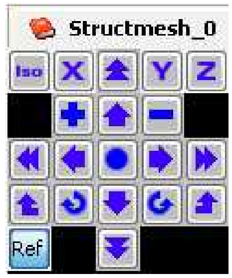


Figure 9.8: Coustyx GUI control panel tools.

- Display element edges of the structure mesh.
  - Move the cursor into the GUI window of the structure mesh.
  - Press and hold the *shift-key* to observe the cursor change from *move-cursor* style (or '+' shape) to an *arrow* style.
  - Right-click on the mesh while holding down the *shift-key* to view a pop-up context menu shown in Figure 9.9 and select: **Select All Displayed Elements**.

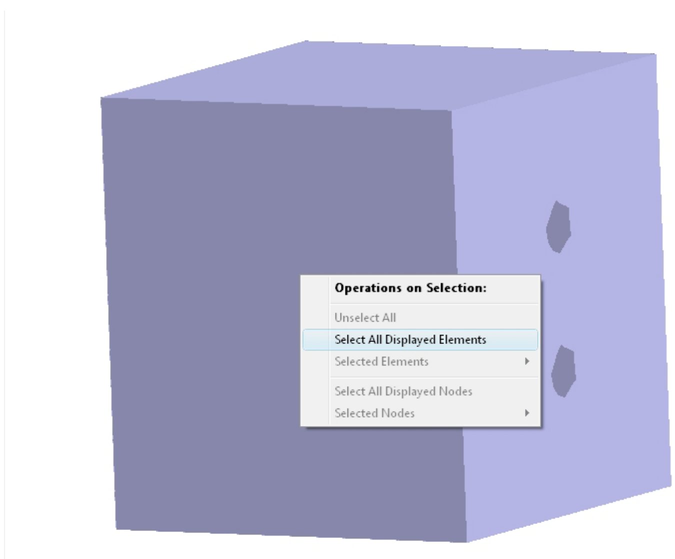


Figure 9.9: Select all displayed elements in the GUI.

  - Again right-click on the mesh while holding down the *shift-key* and select: **Selected Elements** → **Display Style** to view a pop-up window shown in Figure 9.10.
  - Pick the option **Show Edges** and click 'OK'.

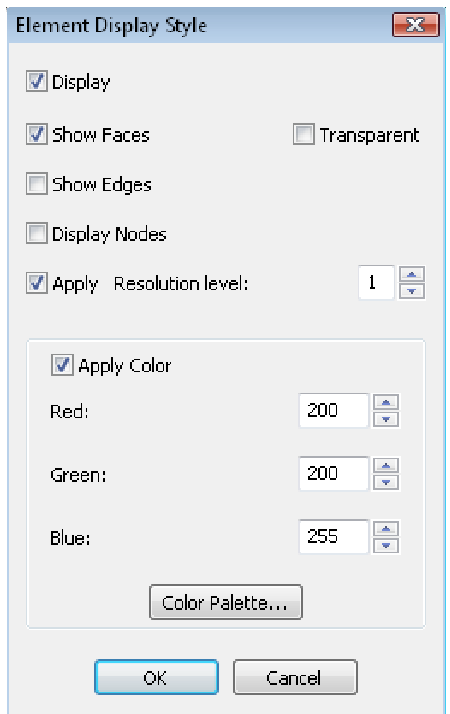


Figure 9.10: Element Display Style Window.

  - To unselect all the elements right-click on the mesh again while holding down the *shift-key* and select: **Unselect All**.

- Create seams at the hole edges to avoid skinning the interior surface of the gearbox housing.

  - Select the tabbed window **Skin** from the series of tabs located below the structure mesh.
  - Move the cursor to the structure mesh in GUI.
  - Left-click on the elements around the edge of a hole while holding the *shift-key*. Make sure to select elements with nodes on the hole edge as shown in Figure 9.11.

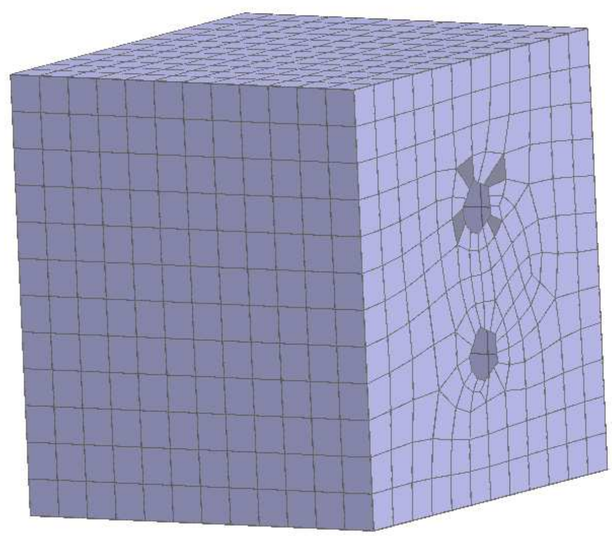


Figure 9.11: Select elements for creating a seam.

  - Right-click on the mesh while holding down the *shift-key* to view the context menu and select: **Selected Elements → Display Connected Nodes** as shown in Figure 9.12.
  - Left-click on the displayed nodes while holding the *shift-key* to pick the nodes to be part of the seam. Make sure to pick nodes in a specific direction.
  - Pick the nodes until you see a circular seam following the edge of the hole as shown in Figure 9.13.
  - From the tabbed windows located below the structure mesh select: **Skin → Accept Seam**.

- Create seams around all the four holes in the gearbox housing following the instructions given above.

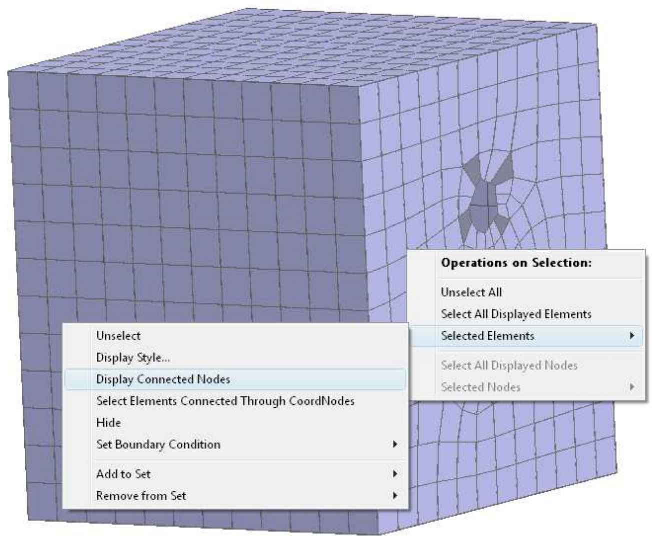


Figure 9.12: Display connected nodes for creating a seam.

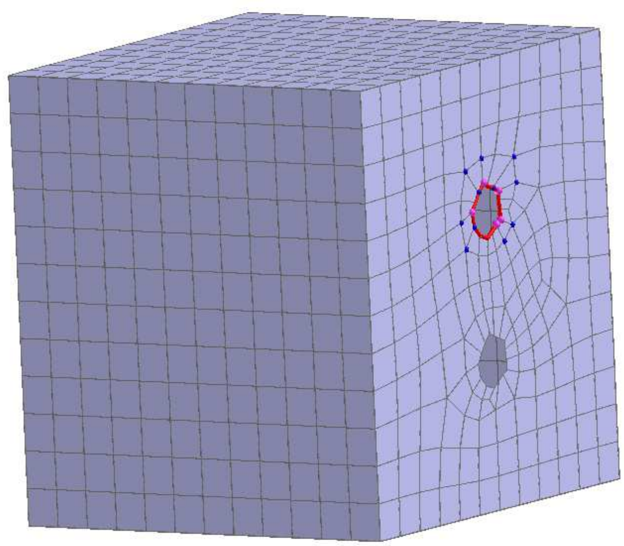


Figure 9.13: Pick nodes to create a seam.

- Right-click on the mesh while holding down the *shift-key* and select: **Unselect All** to unselect all elements.

- Skin the finite element structure mesh to generate a boundary element mesh for Coustyx.

  - Left-click on any element on the exterior surface of the gearbox housing mesh while holding the *shift-key*. Make sure you select only one element.
  - From the tabbed windows located below the structure mesh select: **Skin** → **Create Skin**.
  - Once the skin is created select: **Skin** → **Create Mesh From Skin** to generate a boundary element mesh.

- To verify the creation of boundary element mesh, from the main model menu select: **Model** → **Direct BE Meshes** → **NewMeshCreatedFromSkin**. Right-click on **NewMeshCreatedFromSkin** and select: **Open** to view the boundary element mesh created from skinning the FE structure mesh.

9.4.1.5. Define Material Properties

- In the main model menu select: **Model** → **Materials** → **Air**. Right-click on **Air** and select **Edit...**. Figure 9.14 will appear.

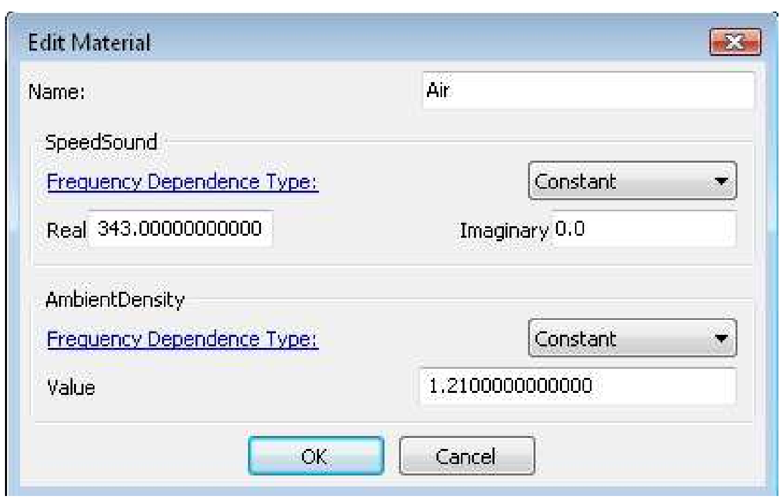


Figure 9.14: Edit material properties.

- Type-in the name of the material as 'Air'.

- Define **SpeedSound** as a constant with value 343 m/s. The unit is consistent with the unit of length (m) in the structure mesh.

- Define **AmbientDensity** as a constant with value 1.21 *kg/m*^{3}. The unit is consistent with the unit of length (m) in the structure mesh.

9.4.1.6. Fill Holes

- Open the Coustyx BE mesh from the main model menu by selecting: **Model** → **Direct BE Meshes** → **NewMeshCreatedFromSkin**. Right-click on **NewMeshCreatedFromSkin** and select: **Open** to view the boundary element mesh in the GUI.
- Select the tabbed window **Fill Hole** from the series of tabs located below the BE mesh.
- Follow the instructions given earlier in Section 9.5.1.4 to display element edges in the mesh.
- Left-click on the elements around the edge of a hole while holding the *shift-key*. Make sure to select elements with nodes on the hole edge (similar to the Figure 9.9).
- Right-click on the mesh while holding down the *shift-key* to view the context menu and select: **Selected Elements** → **Display Connected Nodes** (similar to the Figure 9.12).
- Left-click on the displayed nodes while holding the *shift-key* to pick the nodes on the edge of the hole in a specific direction.
- Pick the nodes until a unique closed loop is identified by the appearance of triangle elements filling the hole as shown in Figure 9.15.

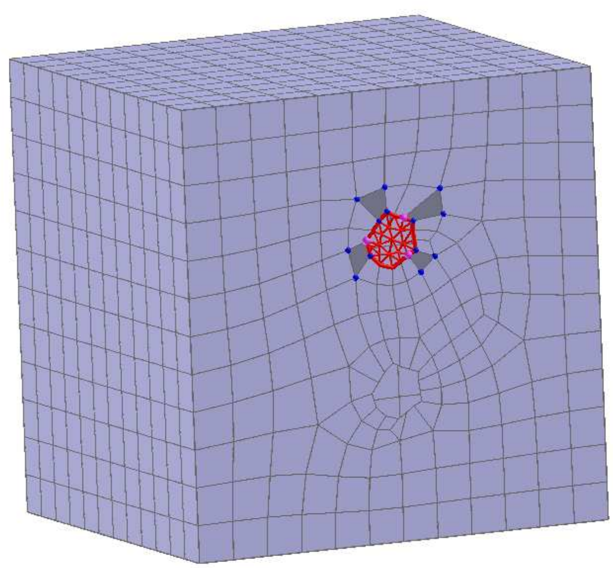


Figure 9.15: Fill hole using triangle elements.

- The elements created to fill the holes are automatically added to a new set created with the name 'Hole_1' in: **Fill Hole → New Set Name**.
- From the tabbed windows located below the structure mesh select: **Fill Hole → Fill Hole**.
- Repeat the above instructions to fill all the four holes in the gearbox housing.

9.4.1.7. Define Boundary Conditions

- Edit the existing default boundary condition.
  - In the Coustyx main model menu select: **Model → Boundary Conditions → Default**.
  - Right-click on **Default** and select: **Edit...** to make changes to the default boundary conditions applied to all the boundary elements. The window in Figure 9.16 will appear.

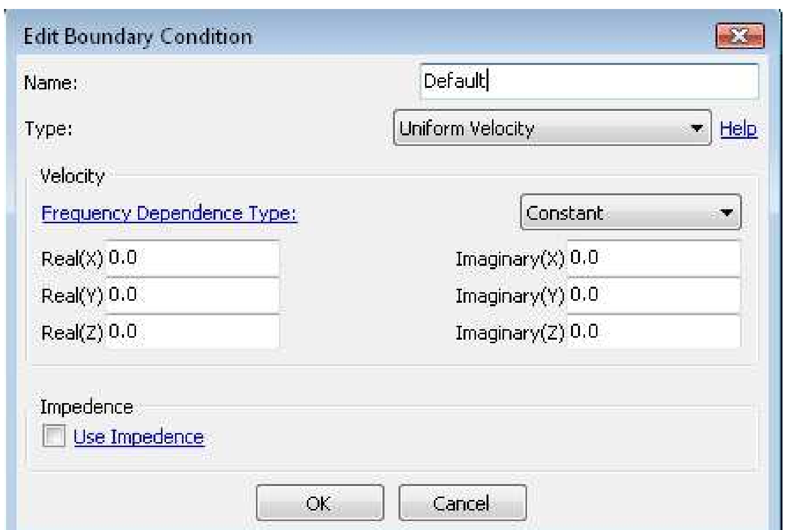


Figure 9.16: Edit boundary conditions window.

  - Type-in the new name 'Structural Velocity BC'.
  - Select **Structure Velocity** from the drop-down menu for 'Type'. The window in Figure 9.17 will appear.
  - Fill the **Structure Name** with **Structmesh_0** or <*Struct mesh name*>. For the current model there is no structure interface, so leave **Structure Interface Name** blank.
  - Select 'Choose Default Options' as interpolation options for mismatched meshes.

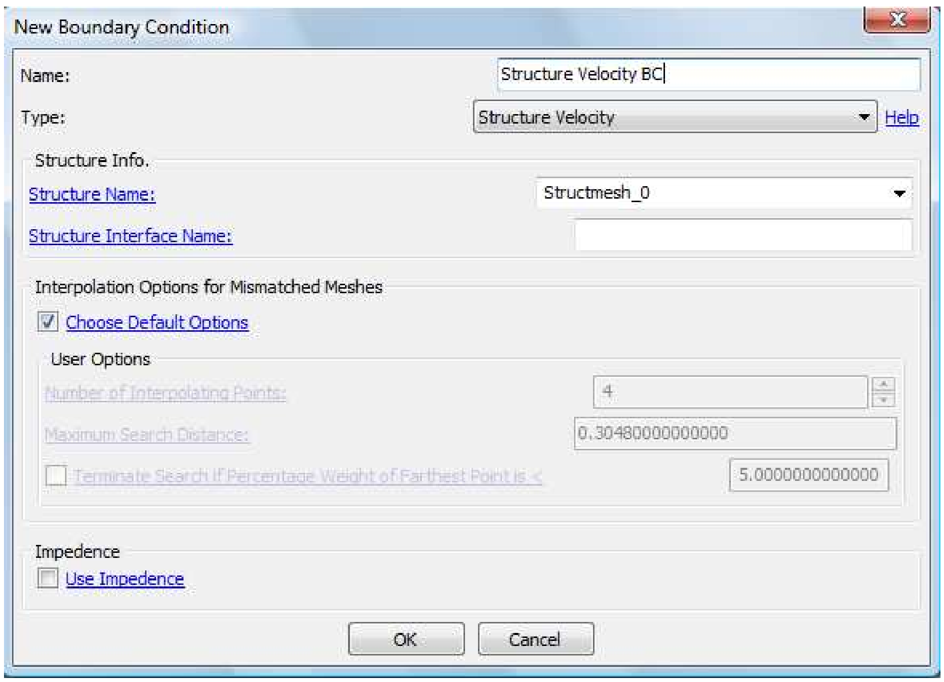


Figure 9.17: Edit structure velocity boundary condition window.

  - Click 'OK' to save the boundary condition.

- Create a new rigid boundary condition.
  - In the main model menu select: **Model** → **Boundary Conditions**.
  - Right-click on **Boundary Conditions** and select **New...**.
  - In the **New Boundary Condition** window type-in the new name 'Rigid BC'.
  - Select **Uniform Normal Velocity** from the drop-down menu for 'Type'.
  - Enter zero constant values for the real and imaginary values of the normal velocity.
  - Click 'OK' to save the boundary condition.

9.4.1.8. Apply Boundary Conditions

The boundary conditions defined earlier are applied to the elements in the Coustyx BE mesh before running acoustic analysis.

- Apply structure velocity boundary condition to all the elements in the Coustyx BE model.
  - Select: **Model** → **Direct BE Meshes** → **NewMeshCreatedFromSkin**. Right-click on **NewMeshCreatedFromSkin** and select **Open** to view the boundary element mesh in the GUI.
  - Right-click on the mesh while holding down the *shift-key* to view the context menu and select: **Select All Displayed Elements**.
  - Again right-click on the mesh while holding down the *shift-key* and select: **Selected Elements** → **Set Boundary Condition** → **Structure Velocity BC** (as shown in Figure 9.18). If the boundary condition **Structure Velocity BC** is inactive, it implies that it has already been applied over the selected elements.

- Apply rigid boundary conditions on all the elements created to fill holes. Note that we don't have structure velocities for these as they are newly created in Coustyx and not present in the original structure. Since the elements filling the holes are conveniently added to sets named 'Hole_1', 'Hole_2' and so on, we can apply the boundary conditions on them through these sets.
  - In the main model menu select: **Model** → **Direct BE Meshes** → **NewMeshCreatedFromSkin**→ **Sets** → **Holes_1**.
  - Right-click on **Holes_1** and select: **Elements** → **Set Boundary Condition** → **Rigid BC** as shown in Figure 9.19.
  - Repeat the above for other holes as well.

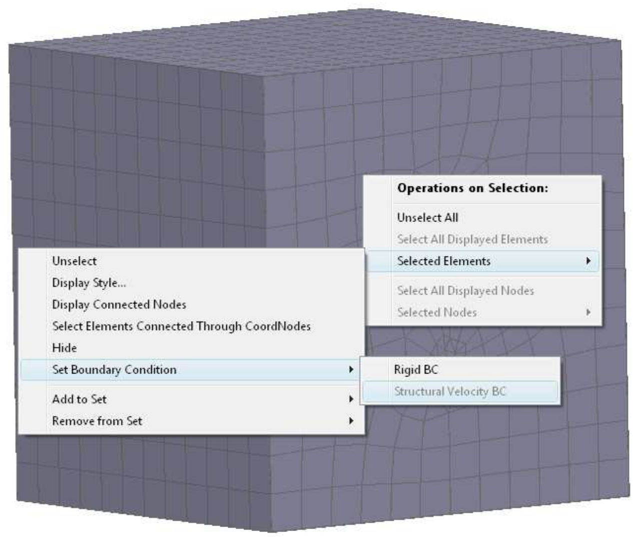


Figure 9.18: Apply boundary conditions through selected elements.

9.4.1.9. Domains

- Specify the type of acoustic problem by selecting: **Model** → **Domains** → **Domain1** or *<Domain Name>* → **Boundedness**. The radiation from a gearbox housing is an exterior or unbounded acoustic problem. Right-click on **Boundedness** and select: **Boundedness** → **Unbounded** as shown in Figure 9.20.

- Choose the fluid medium around the gearbox housing by selecting: **Model** → **Domains** → **Domain1** or *<Domain Name>* → **Material**. Right-click on **Material** and select: **Material** → **Air**.

- To set 'the side of the mesh on which the domain is' you need to first check the direction of mesh normals.

  – Open Coustyx BE mesh in the GUI. Select: **Model** → **Direct BE Meshes** → **NewMeshesCreatedFromSkin** and right-click on **NewMeshesCreatedFromSkin** and select **Open**.

  – Move the cursor to the BE mesh in GUI.

  – Right-click on the mesh while holding down the *shift-key* to view the context menu and select: **Select All Displayed Elements**.

  – From the tabbed windows located below the mesh select: **Element Orientation** to view the direction of element normals (refer to Figure 9.21). The green arrow indicates the positive direction and the red arrow indicates the negative direction of

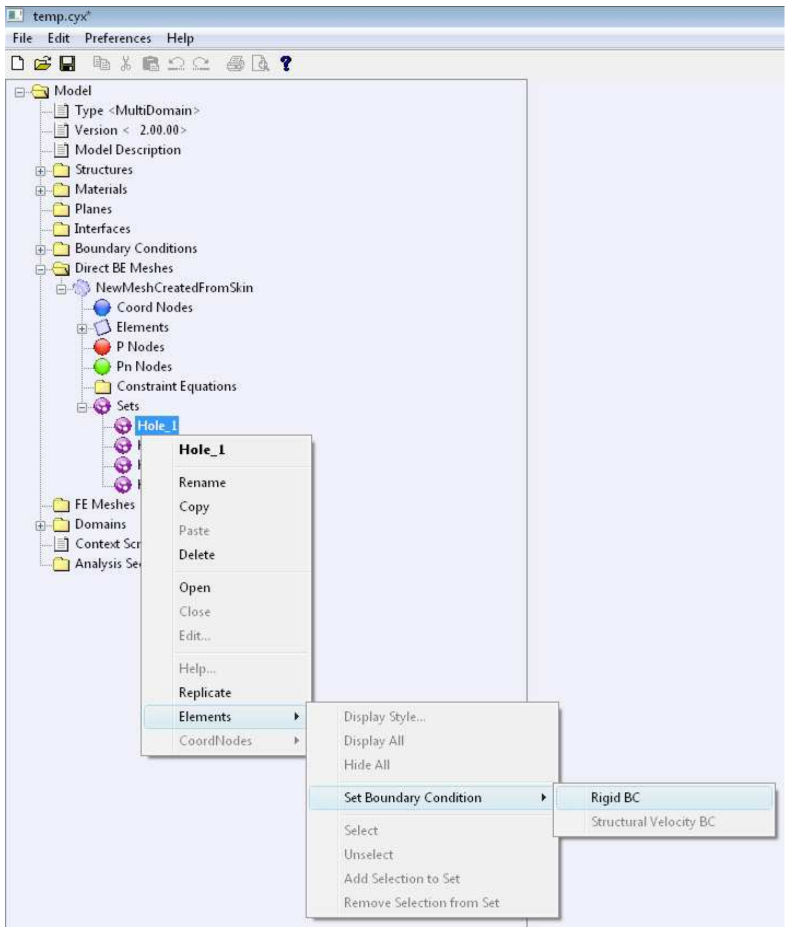


Figure 9.19: Apply boundary conditions through sets.

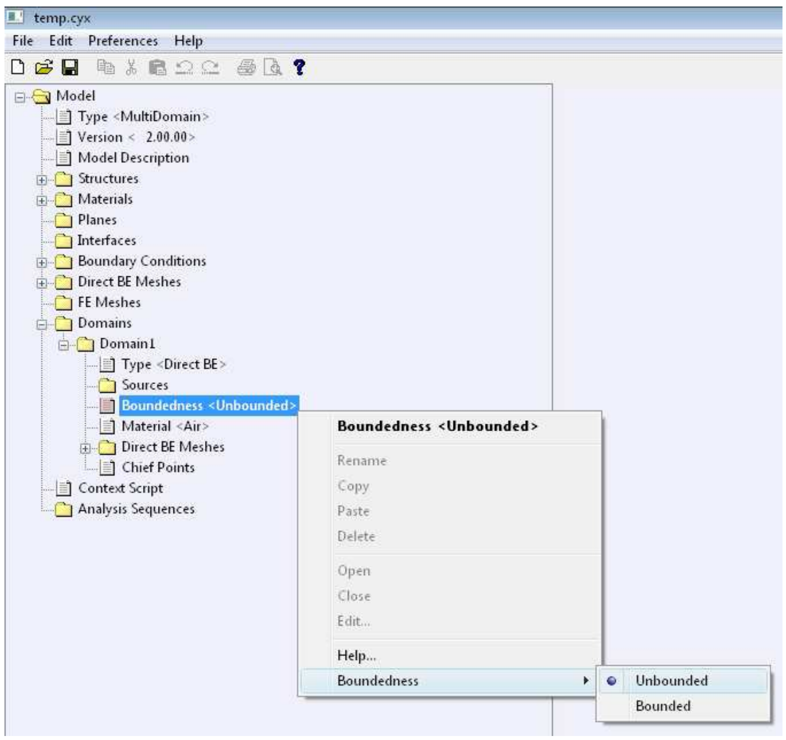


Figure 9.20: Specifying boundedness of the domain.

the normal. Here, all the element normals are coming out of the mesh. By definition the positive side of the mesh is defined as the side with positive element normals.

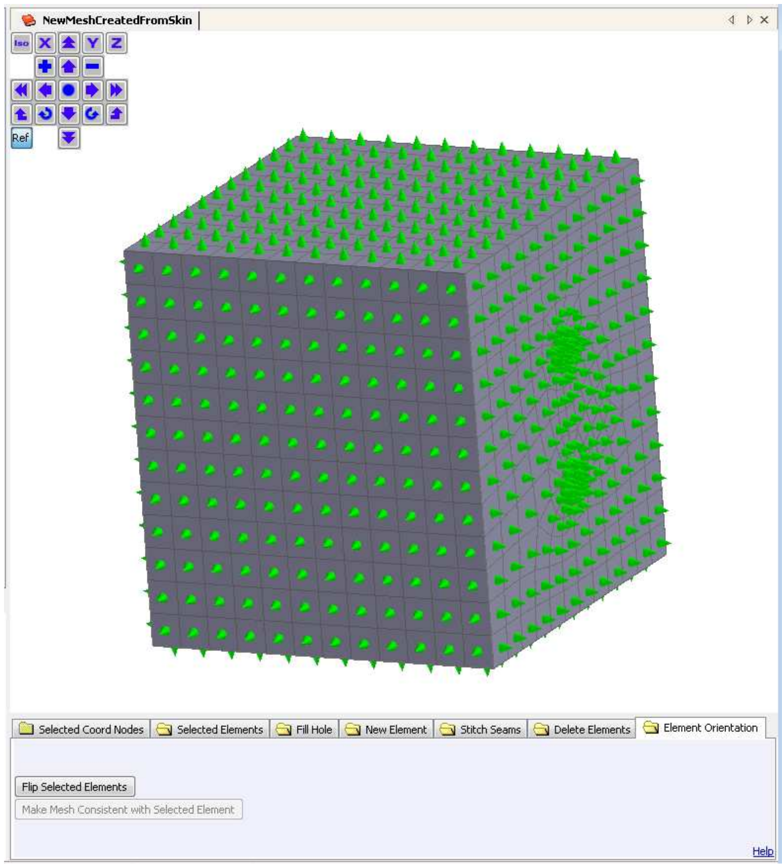


Figure 9.21: Element orientations.

– Since we are interested in the radiation problem the domain is on the positive side of the mesh.

- Select: **Model** → **Domains** → **Domain1** or <*Domain Name*> → **Direct BE Meshes** → **NewMeshesCreatedFromSkin** → **Side of Mesh on which Domain is**. Right-click on **Side of Mesh on which Domain is** and select: **Side of Mesh on which Domain is** → **Positive** as shown in Figure 9.22.

9.4.2. Run Acoustic Analysis

Coustyx analysis parameters are set in ‘Analysis Sequences’, which are then ‘Run’ to solve the acoustics radiation problem.

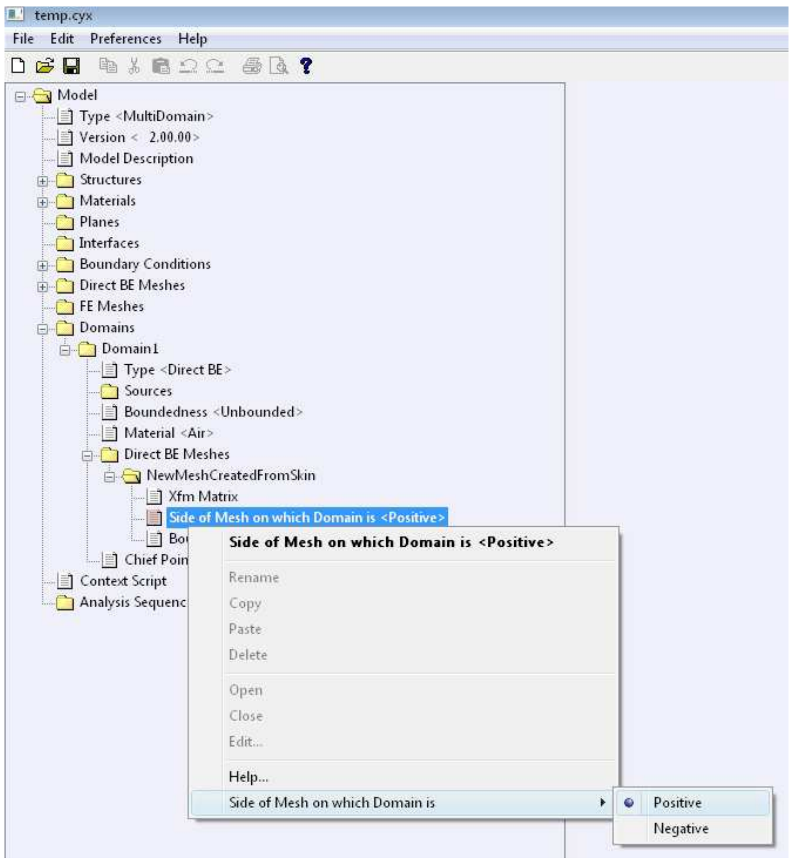


Figure 9.22: Set the side of the mesh on which the domain is.

- Select: **Model** → **Analysis Sequences** and right-click to create a new analysis sequence by selecting **New...**.
- Select **Solver Controls** tab to set solver parameters. Refer to Figure 9.23.
  - Ensure the default solver options are satisfactory.
  - Set **Initial Guess** → **Previous Solution** from the drop-down menu.
- Move onto **Frequency Ranges** tab to specify analysis frequencies. Refer to Figure 9.24.
  - Enter the starting frequency to be 100 Hz in the table under 'Start (Hz)'.
  - Enter a value of 15 Hz for the frequency resolution under 'Delta (Hz)'.
  - Enter the final frequency to be 1000 Hz in the table under 'End (Hz)'.
- Now move onto **Outputs** tab where output file names are specified. Ensure the default settings in **Outputs** tab are satisfactory.
- Click 'OK' to save the new analysis sequence.
- To edit the analysis parameters any time, select: **Model** → **Analysis Sequences** → **Analysis Sequence**. Right-click and select **Edit**.
- To start acoustic analysis, select: **Model** → **Analysis Sequences** → **Analysis Sequence**. Right-click and select **Run** to perform acoustic analysis on the gearbox housing with the applied vibrations for the desired frequencies.

9.4.3. Post-processing/Outputs

Coustyx creates the following output files based on the choices made in **Outputs** tab in **Analysis Sequence**.

9.4.3.1. results.dat

A binary results file is saved by Coustyx for later use. When the model is re-run Coustyx directly uses these results if the checksum of the model matches with the checksum in the results file. This file can't be interpreted by the user and is only for Coustyx use.

9.4.3.2. sensors.dat

The pressure and particle velocity at the sensor locations are written into this *ASCII*-text file. Since we didn't add any sensors to the gearbox housing radiation problem, this file is empty.

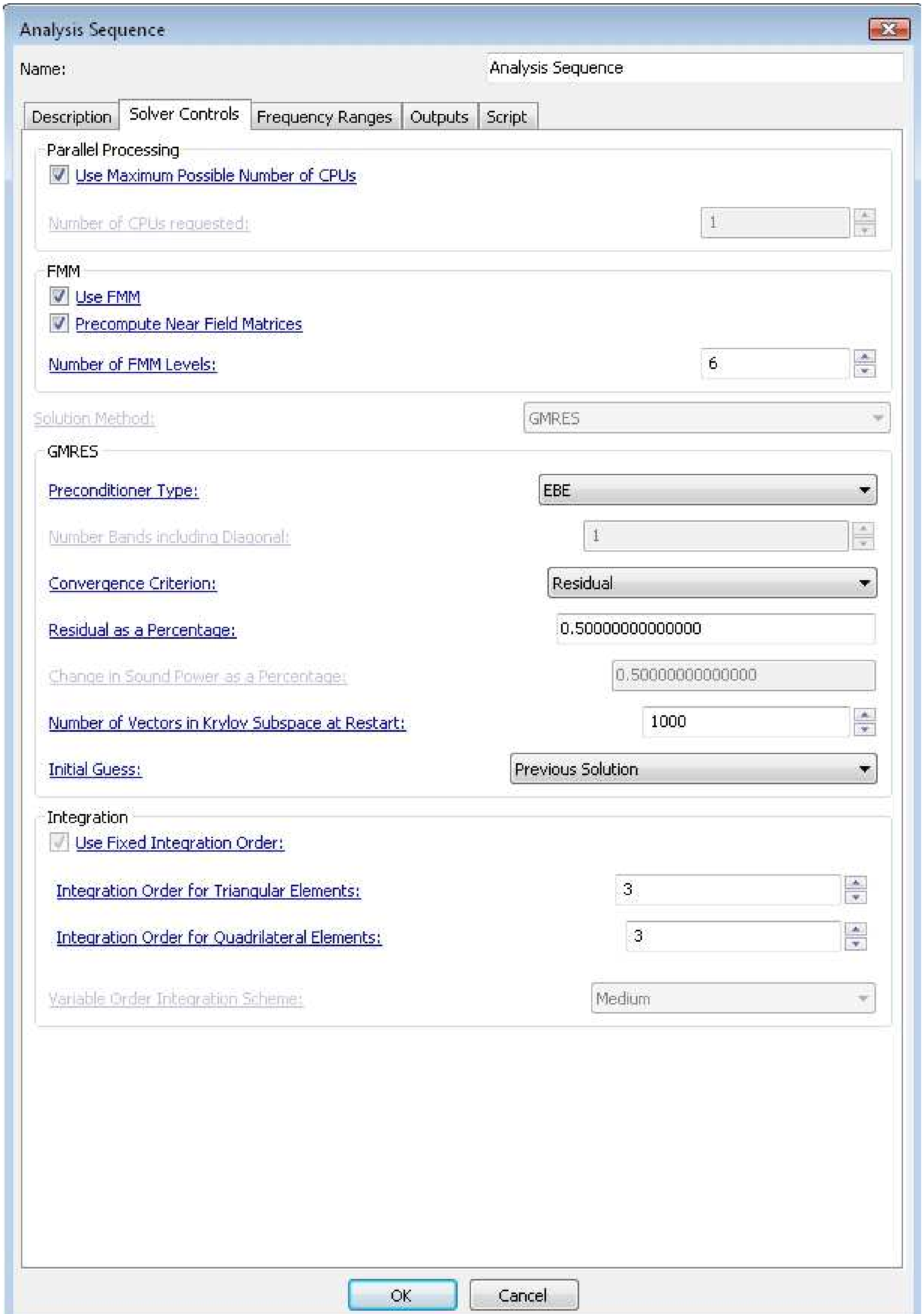


Figure 9.23: Analysis solver controls.

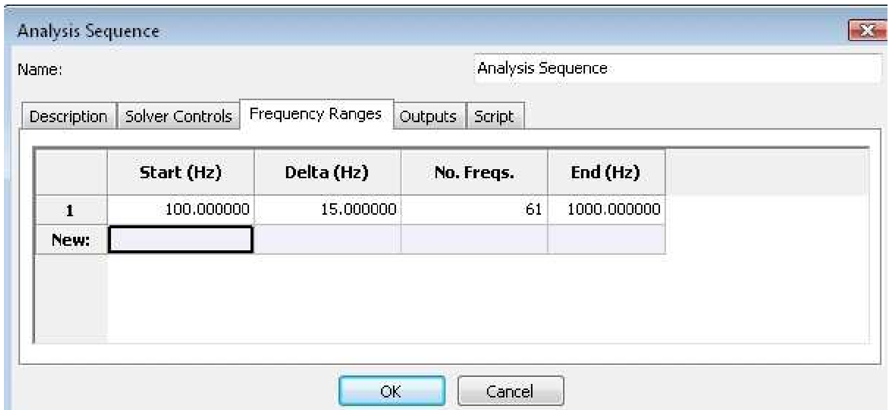


Figure 9.24: Set analysis frequencies.

9.4.3.3. power.dat

This *ASCII*-text file contains acoustic power values computed at each analysis frequency. Each file has five columns. The first column contains analysis frequencies in 'Hertz'. The second and third columns contain radiated (active) sound power and reactive sound power respectively. The input power is written to the fourth column. All the power units will be consistent with the material properties - sound speed, and ambient density, defined earlier; here the unit is 'Watt'. The fifth column consists of the radiation efficiency of the gearbox housing.

The radiated sound power and radiation efficiency are plotted against the analysis frequency in Figure 9.25 and Figure 9.26 using matlab plot command. The sound power radiated (Figure 9.25) has peaks corresponding to the structural vibration modes that have a non-zero net volume velocity.

9.4.3.4. iglass.igl

IGlass files are post-processing data files created by Coustyx to visualize the acoustic analysis results. Refer to Figure 9.27.

- Double-click on 'iglass.igl' file to open it.
- Click on the **Attribs** tab on the top-left of the iglass viewer.
- Select: **Attribs** → **Attribute** → **Pressure** to view the pressure distribution on the surface of the gearbox housing.
- Click on the **View** tab on the top-left of the iglass viewer.
- Press the slider under **View** → **Phase**, to start animation. This activates the animation of the wave propagation on the housing surface.
- To view the results for different frequencies press the slider under **View** → **Frequency**.

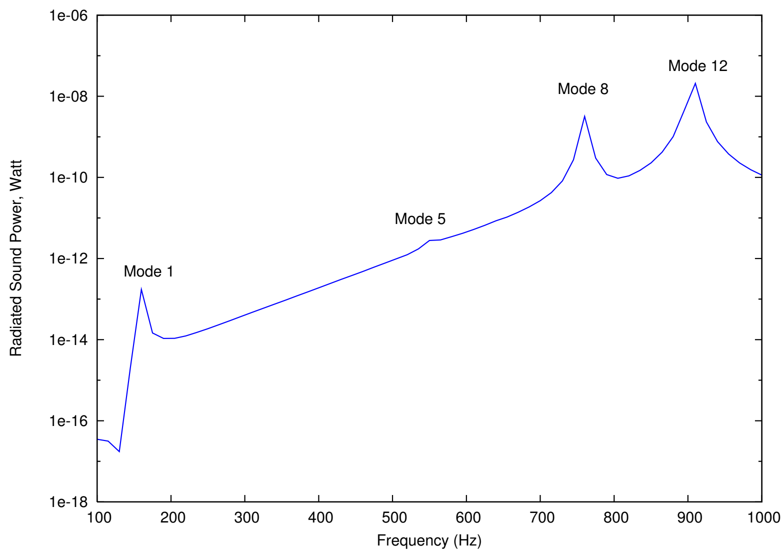


Figure 9.25: Sound power from the forced vibration response.

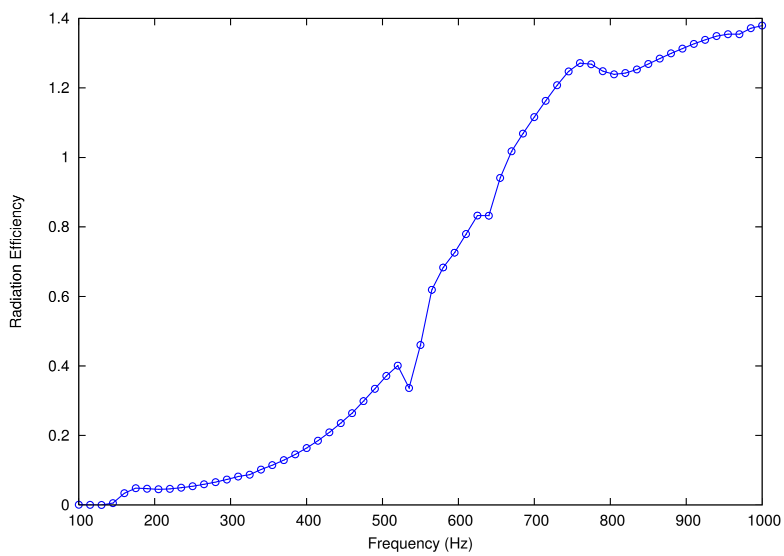


Figure 9.26: Radiation efficiency of the gearbox forced vibration.

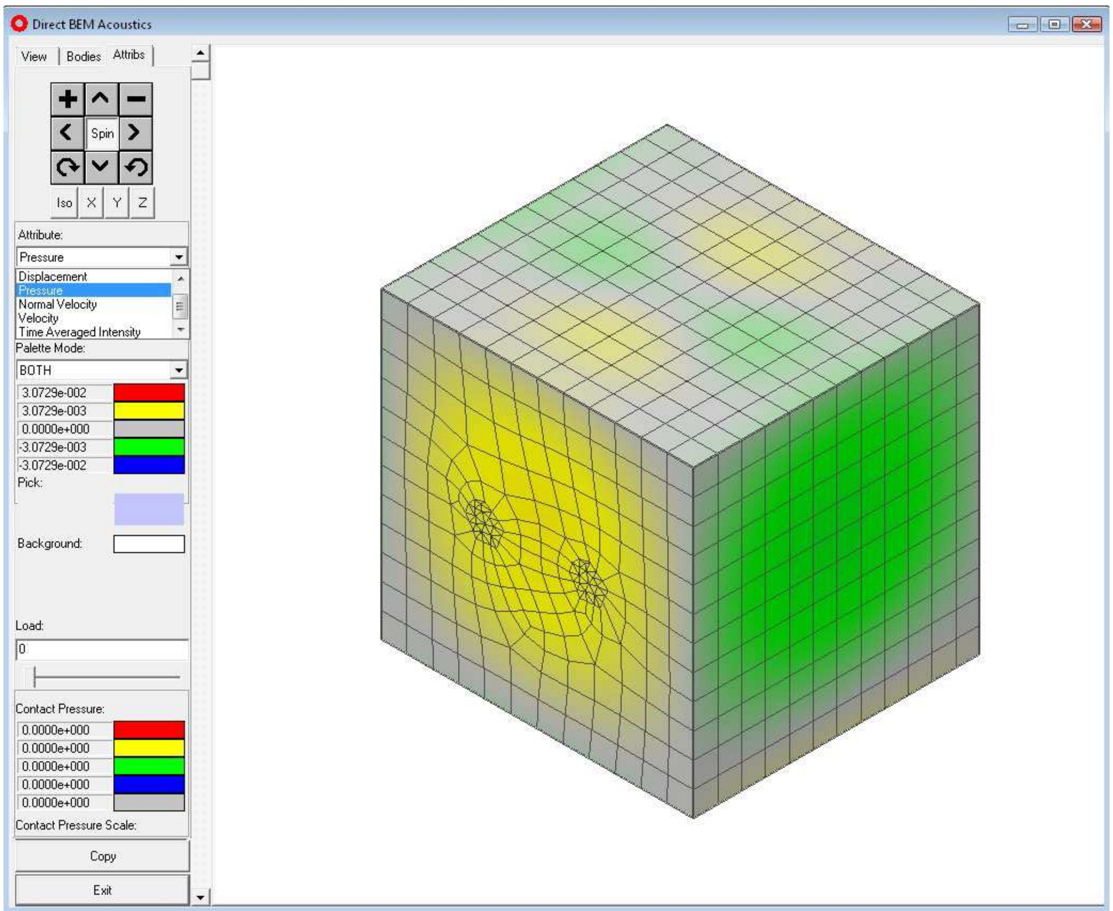


Figure 9.27: IGlass viewer showing sound pressure distribution at 760 Hz.

9.5. Coustyx Indirect Model

Coustyx Indirect model is created by importing the FE mesh. The frequency response data from the FEA analysis is loaded into Coustyx and is applied as a structure velocity boundary condition on the gearbox housing. The analysis parameters are then set and the acoustic analysis is run to compute radiation predictions.

9.5.1. Problem Setup

Follow the steps to setup Coustyx model and perform acoustic analysis on the gearbox housing. Open Coustyx from the start menu of your computer.

9.5.1.1. Create a New Model

- In the main menu select: **File** → **New Model**. The window in Figure 9.28 will then appear.
- Choose the model type: 'Indirect Model' and select model units: meter-kilogram-second (SI units). Note that the selection of model units is consistent with the unit of length in the structure mesh. Click 'OK' to proceed.

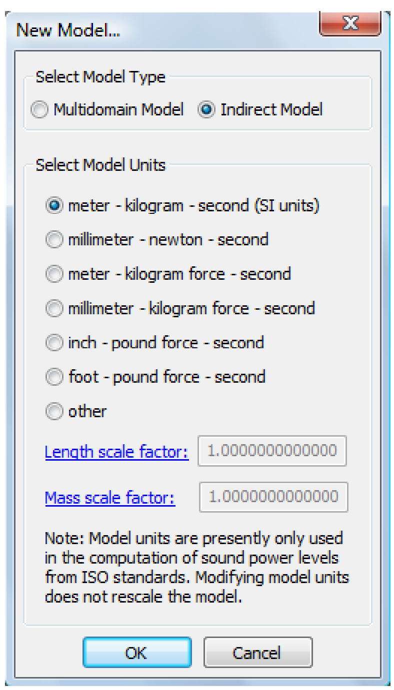


Figure 9.28: New Model Selection Window.

9.5.1.2. Import FE Structure Mesh

- In Coustyx model main menu select: **Model** → **Structures**.
- Right–click on **Structures** and select: **Import** → **Nastran Bulk Data (.bdf) File...** (as shown in Figure 9.29) to import mesh from Nastran bulk data format. The FE meshes from Abaqus and Ansys data formats can be imported by selecting **Abaqus (.inp) File** or **Ansys Results (.rst) Files** respectively.

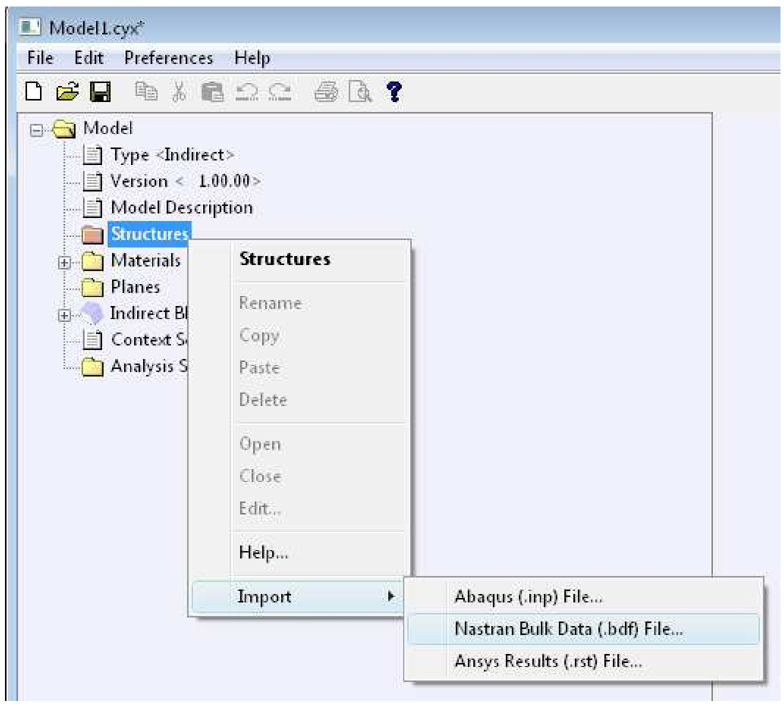


Figure 9.29: Import a finite element structure mesh.

- Select the appropriate FE structure data file to be imported from the browser and click ‘Open’.

9.5.1.3. Load Frequency Response Data

- In Coustyx model main menu select: **Model** → **Structures** → **Structmesh_0** or <*Struct Mesh Name*>.
- Right-click on **Structmesh_0** or <*Struct Mesh Name*> and select: **Load Freq Response Data** → **Nastran Punch File...** (as shown in Figure 9.30). The other valid data formats from which frequency response data can be loaded into Coustyx are **Nastran OP2 File** and **Ansys rst File**.

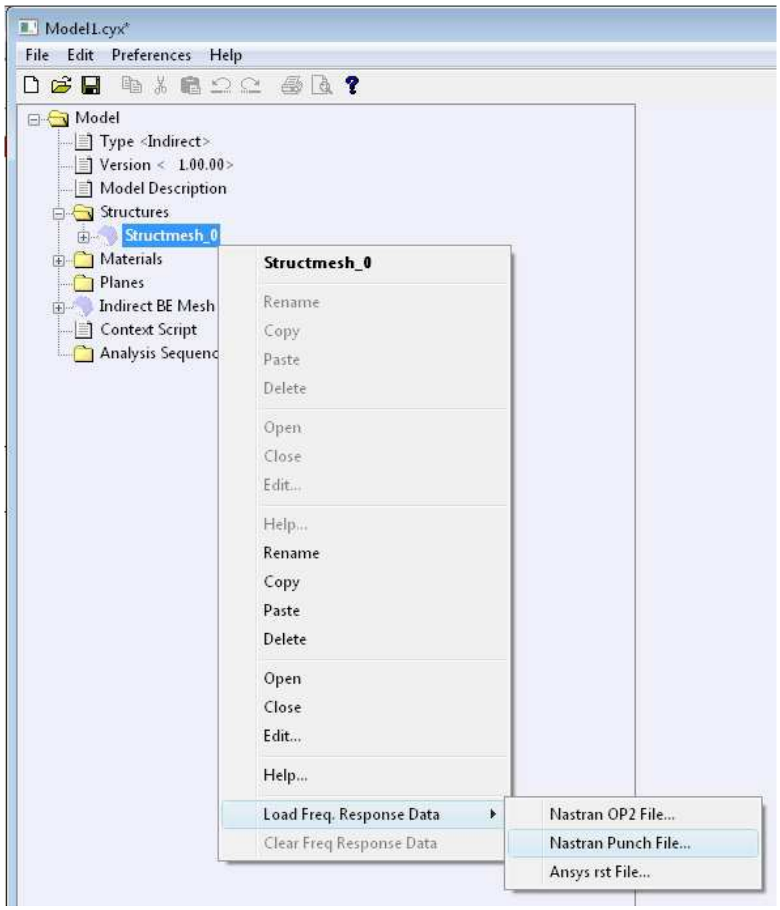


Figure 9.30: Load frequency response data into Coustyx.

- Select the appropriate frequency response data file to be loaded from the browser and click 'Open'.

9.5.1.4. Generate BE Mesh

- Select: **Model** → **Structures** → **Structmesh_0** or <*Struct Mesh Name*>.
- Right-click on **Structmesh_0** or <*Struct Mesh Name*> and select: **Open** to view the structure mesh in the GUI. The structure mesh will appear as shown in Figure 9.31.

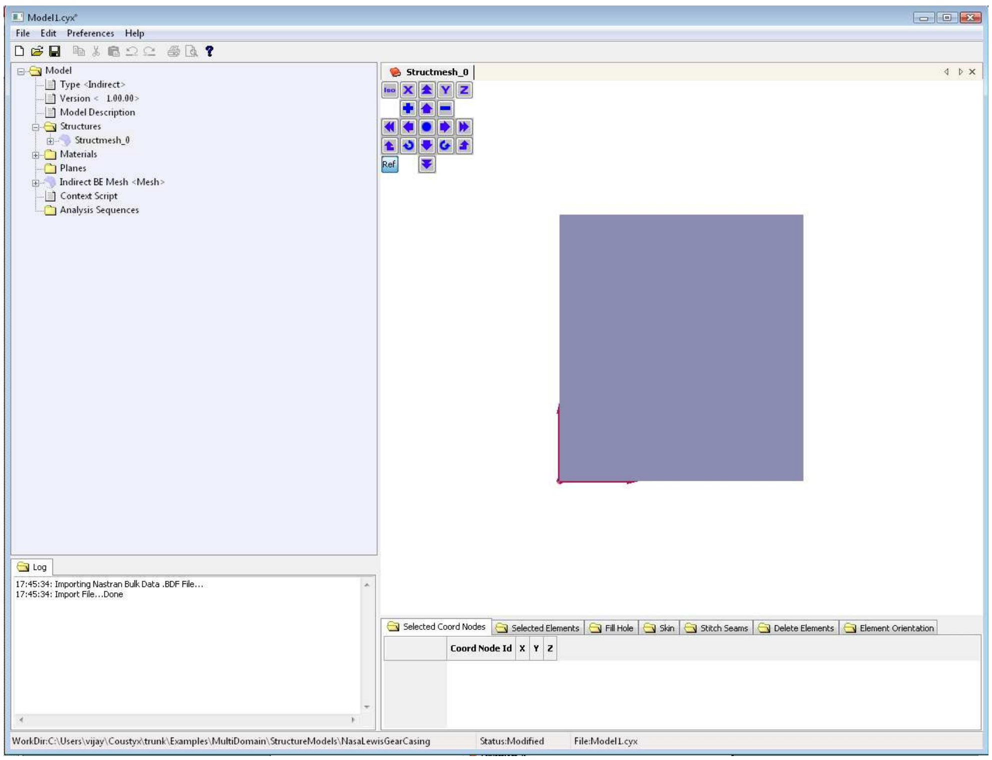


Figure 9.31: Structure mesh opened in Coustyx GUI.

- Move the cursor into the GUI window with the structure mesh and observe the cursor change to *move-cursor* style (or to the shape of '+'). To manipulate the view:
  - Use the GUI control panel tools shown in Figure 9.32 to *zoom* and *rotate* the model.
  - Hold down the left-click button and move the mouse to rotate the model in the GUI.
  - Hold down the right-click button and move the mouse to move the model in the GUI.
  - Move and rotate the model to see the holes on one of the side surfaces on the structure.

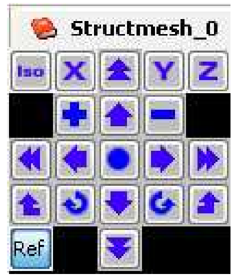


Figure 9.32: Coustyx GUI control panel tools.

- Display element edges of the structure mesh.
  - Move the cursor into the GUI window of the structure mesh.
  - Press and hold the *shift-key* to observe the cursor change from *move-cursor* style (or '+' shape) to an *arrow* style.
  - Right-click on the mesh while holding down the *shift-key* to view a pop-up context menu shown in Figure 9.33 and select: **Select All Displayed Elements**.

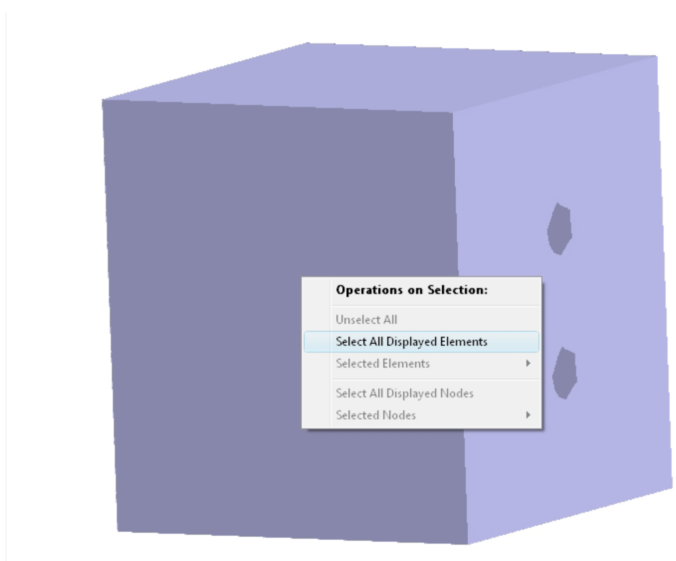


Figure 9.33: Select all displayed elements in the GUI.

  - Again right-click on the mesh while holding down the *shift-key* and select: **Selected Elements → Display Style** to view a pop-up window shown in Figure 9.34.
  - Pick the option **Show Edges** and click 'OK'.

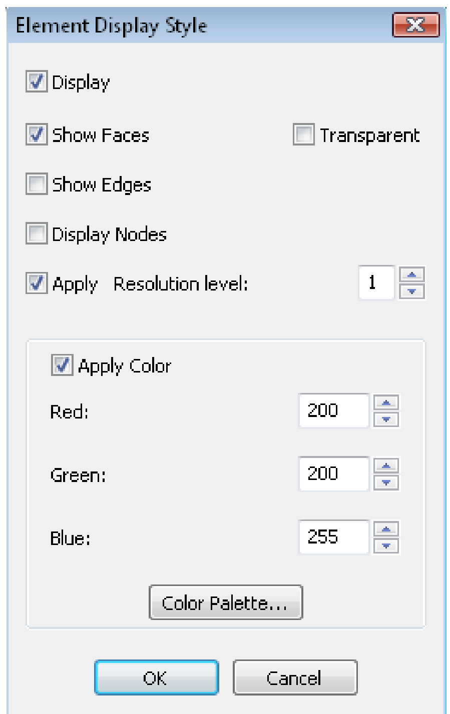


Figure 9.34: Element Display Style Window.

- To unselect all the elements right-click on the mesh again while holding down the *shift-key* and select: **Unselect All**.

- Create seams at the hole edges to avoid skinning the interior surface of the gearbox housing.
  - Select the tabbed window **Skin** from the series of tabs located below the structure mesh.
  - Move the cursor to the structure mesh in GUI.
  - Left-click on the elements around the edge of a hole while holding the *shift-key*. Make sure to select elements with nodes on the hole edge as shown in Figure 9.35.

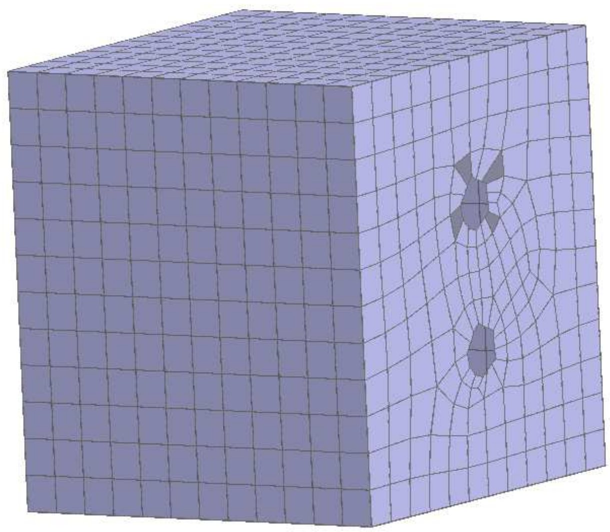


Figure 9.35: Select elements for creating a seam.

  - Right-click on the mesh while holding down the *shift-key* to view the context menu and select: **Selected Elements** → **Display Connected Nodes** as shown in Figure 9.36.
  - Left-click on the displayed nodes while holding the *shift-key* to pick the nodes to be part of the seam. Make sure to pick nodes in a specific direction.
  - Pick the nodes until you see a circular seam following the edge of the hole as shown in Figure 9.37.
  - From the tabbed windows located below the structure mesh select: **Skin** → **Accept Seam**.

- Create seams around all the four holes in the gearbox housing following the instructions given above.

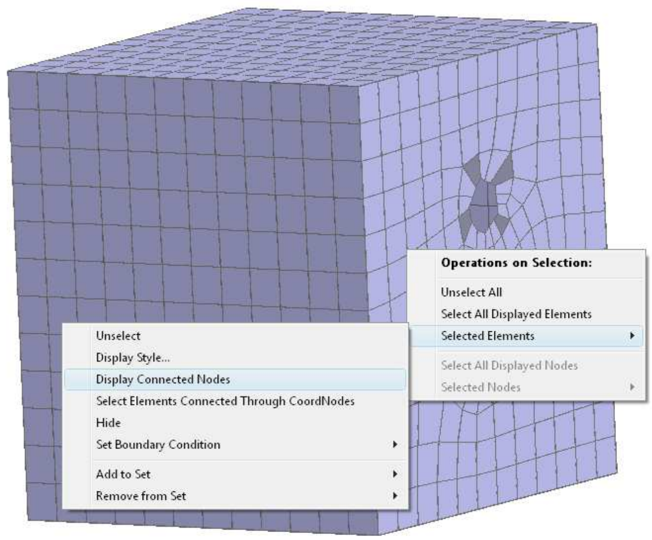


Figure 9.36: Display connected nodes for creating a seam.

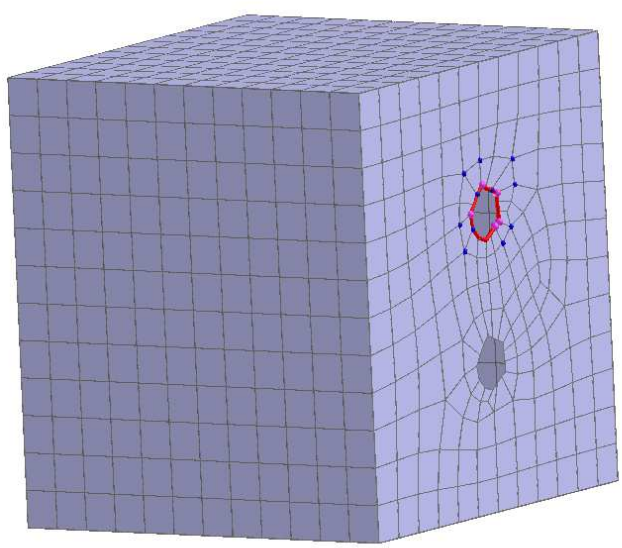


Figure 9.37: Pick nodes to create a seam.

- Right-click on the mesh while holding down the *shift-key* and select: **Unselect All** to unselect all elements.
- Skin the finite element structure mesh to generate a boundary element mesh for Coustyx.
  - Left-click on any element on the exterior surface of the gearbox housing mesh while holding the *shift-key*. Make sure you select only one element.
  - From the tabbed windows located below the structure mesh select: **Skin** → **Create Skin**.
  - Once the skin is created select: **Skin** → **Create Mesh From Skin** to generate a boundary element mesh.
- To verify the creation of boundary element mesh, from the main model menu select: **Model** → **Indirect BE Mesh**. Right-click on it and select: **Open** to view the boundary element mesh created from skinning the FE structure mesh.

9.5.1.5. Define Material Properties

- In the main model menu select: **Model** → **Materials** → **Air**. Right-click on **Air** and select **Edit...**. Figure 9.38 will appear.

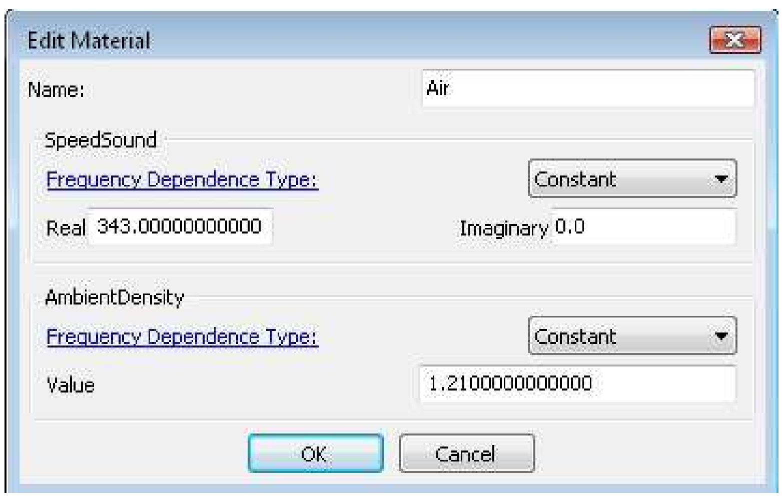


Figure 9.38: Edit material properties.

- Type-in the name of the material as ‘Air’.
- Define **SpeedSound** as a constant with value 343 m/s. The unit is consistent with the unit of length (m) in the structure mesh.
- Define **AmbientDensity** as a constant with value 1.21 kg/m^3. The unit is consistent with the unit of length (m) in the structure mesh.

9.5.1.6. Fill Holes

- Open the Coustyx BE mesh from the main model menu by selecting: **Model → Indirect BE Mesh**. Right-click on it and select: **Open** to view the boundary element mesh in the GUI.
- Select the tabbed window **Fill Hole** from the series of tabs located below the BE mesh.
- Follow the instructions given earlier in Section 9.5.1.4 to display element edges in the mesh.
- Left-click on the elements around the edge of a hole while holding the *shift-key*. Make sure to select elements with nodes on the hole edge (similar to the Figure 9.33).
- Right-click on the mesh while holding down the *shift-key* to view the context menu and select: **Selected Elements → Display Connected Nodes** (similar to the Figure 9.36).
- Left-click on the displayed nodes while holding the *shift-key* to pick the nodes on the edge of the hole in a specific direction.
- Pick the nodes until a unique closed loop is identified by the appearance of triangle elements filling the hole as shown in Figure 9.39.

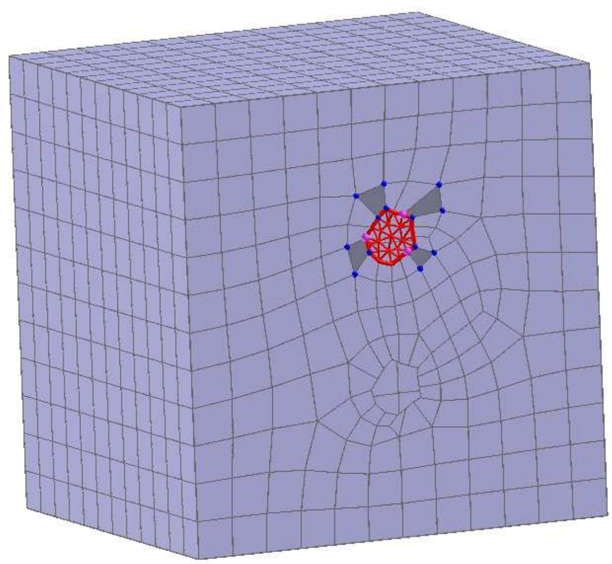


Figure 9.39: Fill hole using triangle elements.

- The elements created to fill the holes are automatically added to a new set created with the name 'Hole_1' in: **Fill Hole → New Set Name**.

- From the tabbed windows located below the structure mesh select: **Fill Hole** → **Fill Hole**.
- Repeat the above instructions to fill all the four holes in the gearbox housing.

9.5.1.7. Define Boundary Conditions

- Edit the existing default boundary condition.
  - In the Coustyx main model menu select: **Model** → **Indirect BE Mesh** → **Boundary Conditions** → **Default**.
  - Right-click on **Default** and select: **Edit...** to make changes to the default boundary conditions applied to all the boundary elements. The window in Figure 9.40 will appear.

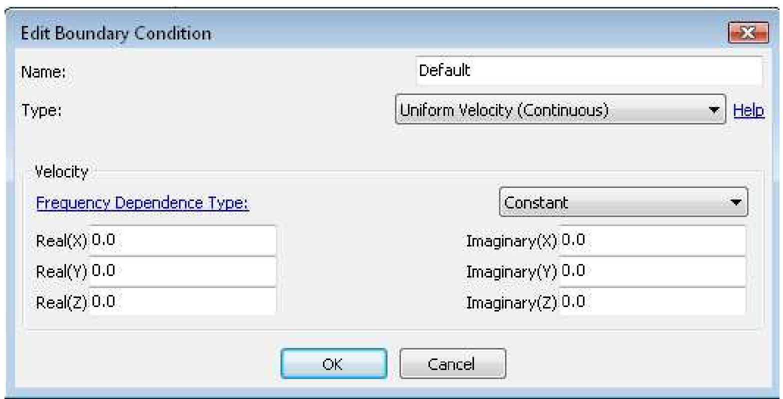


Figure 9.40: Edit boundary conditions window.

  - Type-in the new name 'Structural Velocity BC'.
  - Select **Structure Velocity (Continuous)** from the drop-down menu for 'Type'. The window in Figure 9.41 will appear.
  - Fill the **Structure Name** with **Structmesh_0** or <*Struct mesh name*>. For the current model there is no structure interface, so leave **Structure Interface Name** blank.
  - Select 'Choose Default Options' as interpolation options for mismatched meshes.
  - Click 'OK' to save the boundary condition.
- Create a new rigid boundary condition.
  - In the main model menu select: **Model** → **Boundary Conditions**.
  - Right-click on **Boundary Conditions** and select **New...**.
  - In the **New Boundary Condition** window type-in the new name 'Rigid BC'.

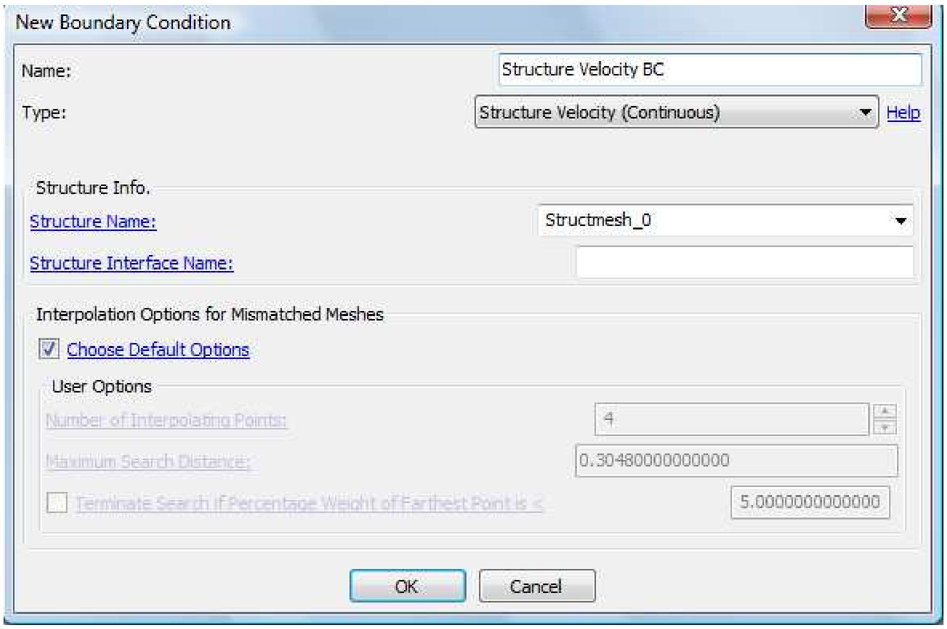


Figure 9.41: Edit structure velocity boundary condition window.

- Select **Uniform Normal Velocity (Continuous)** from the drop-down menu for 'Type'.
- Enter zero constant values for the real and imaginary values of the normal velocity.
- Click 'OK' to save the boundary condition.

9.5.1.8. Apply Boundary Conditions

The boundary conditions defined earlier are applied to the elements in the Coustyx BE mesh before running acoustic analysis.

- Apply structure velocity boundary condition to all the elements in the Coustyx BE model.
  - Select: **Model** → **Indirect BE Mesh**. Right-click on it and select **Open** to view the boundary element mesh in the GUI.
  - Right-click on the mesh while holding down the *shift-key* to view the context menu and select: **Select All Displayed Elements**.
  - Again right-click on the mesh while holding down the *shift-key* and select: **Selected Elements** → **Set Boundary Condition** → **Structure Velocity BC** (as shown in Figure 9.42). If the boundary condition **Structure Velocity BC** is inactive, it implies that it has already been applied over the selected elements.

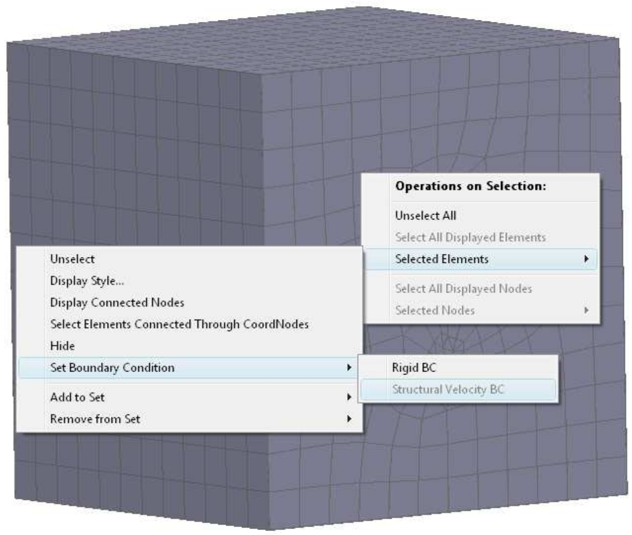


Figure 9.42: Apply boundary conditions through selected elements.

- Apply rigid boundary conditions on all the elements created to fill holes. Note that we don't have structure velocities for these as they are newly created in Coustyx and not present in the original structure. Since the elements filling the holes are conveniently added to sets named 'Hole_1', 'Hole_2' and so on, we can apply the boundary conditions on them through these sets.
  - In the main model menu select: **Model** → **Indirect BE Mesh** → **Sets** → **Holes_1**.
  - Right-click on **Holes_1** and select: **Elements** → **Set Boundary Condition** → **Rigid BC** as shown in Figure 9.43.
  - Repeat the above for other holes as well.

9.5.2. Run Acoustic Analysis

Coustyx analysis parameters are set in 'Analysis Sequences', which are then 'Run' to solve the acoustics radiation problem.

- Select: **Model** → **Analysis Sequences** and right-click to create a new analysis sequence by selecting **New...**.
- Select **Solver Controls** tab to set solver parameters. Refer to Figure 9.44.
  - Ensure the default solver options are satisfactory.
  - Set **Initial Guess** → **Previous Solution** from the drop-down menu.

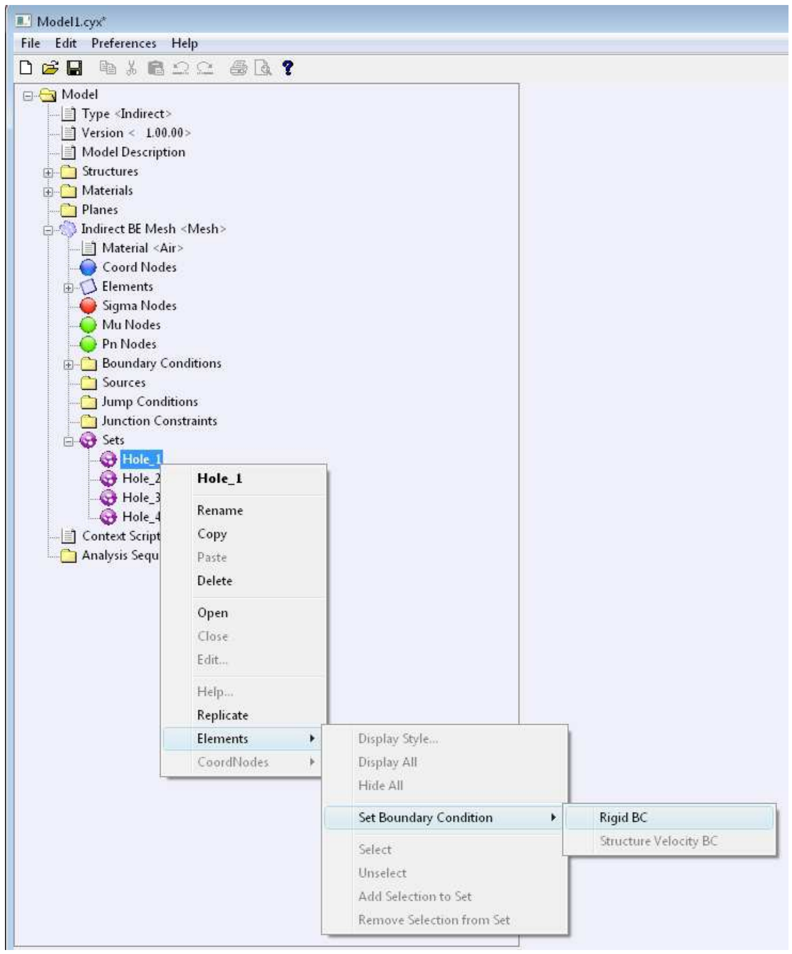


Figure 9.43: Apply boundary conditions through sets.

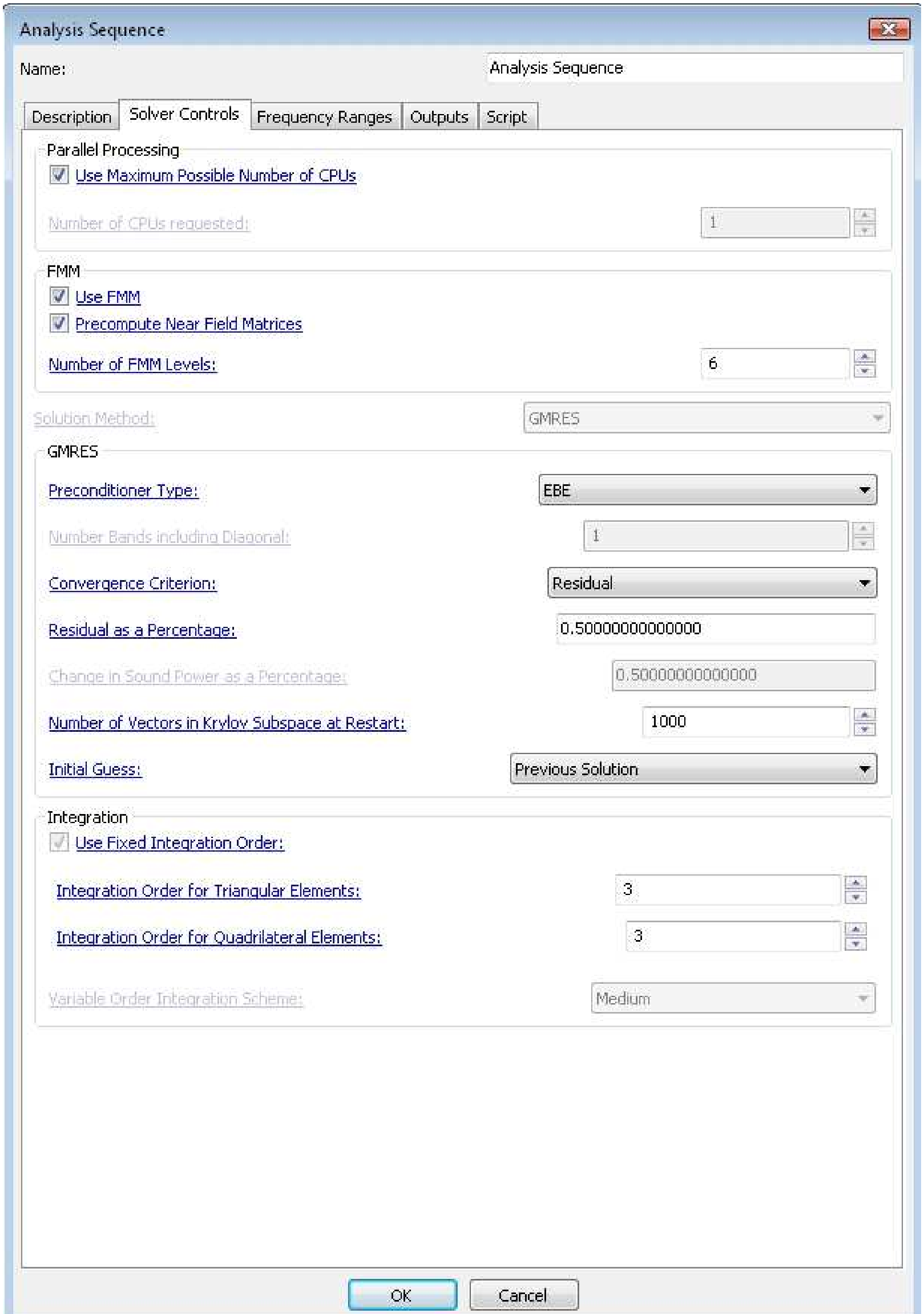


Figure 9.44: Analysis solver controls.

- Move onto **Frequency Ranges** tab to specify analysis frequencies. Refer to Figure 9.45.
  - Enter the starting frequency to be 100 Hz in the table under 'Start (Hz)'.
  - Enter a value of 15 Hz for the frequency resolution under 'Delta (Hz)'.
  - Enter the final frequency to be 1000 Hz in the table under 'End (Hz)'.

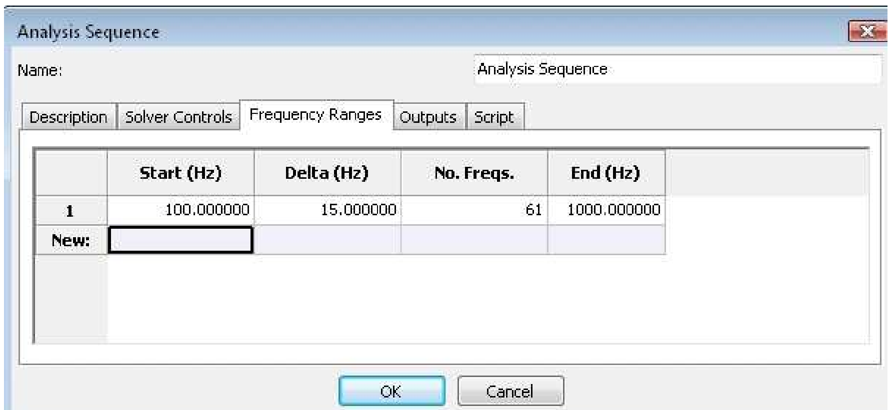


Figure 9.45: Set analysis frequencies.

- Now move onto **Outputs** tab where output file names are specified. Ensure the default settings in **Outputs** tab are satisfactory.
- Click 'OK' to save the new analysis sequence.
- To edit the analysis parameters any time, select: **Model** → **Analysis Sequences** → **Analysis Sequence**. Right-click and select **Edit**.
- To start acoustic analysis, select: **Model** → **Analysis Sequences** → **Analysis Sequence**. Right-click and select **Run** to perform acoustic analysis on the gearbox housing with the applied vibrations for the desired frequencies. Note that Coustyx solves both the interior and exterior domains simultaneously for the 'Indirect Model'.

9.5.3. Post-processing/Outputs

Coustyx creates the following output files based on the choices made in **Outputs** tab in **Analysis Sequence**.

9.5.3.1. results.dat

A binary results file is saved by Coustyx for later use. When the model is re-run Coustyx directly uses these results if the checksum of the model matches with the checksum in the results file. This file can't be interpreted by the user and is only for Coustyx use.

9.5.3.2. sensors.dat

The pressure and particle velocity at the sensor locations are written into this *ASCII*-text file. Since we didn't add any sensors to the gearbox housing radiation problem, this file is empty.

9.5.3.3. power.dat

This *ASCII*-text file contains acoustic power values computed at each analysis frequency. Each file has five columns. The first column contains analysis frequencies in 'Hertz'. The second and third columns contain radiated (active) sound power and reactive sound power respectively. The input power is written to the fourth column. All the power units will be consistent with the material properties - sound speed, and ambient density, defined earlier; here the unit is 'Watt'. The fifth column consists of the radiation efficiency of the gearbox housing.

The radiated sound power and radiation efficiency are plotted against the analysis frequency in Figure 9.46 and Figure 9.47 using matlab plot command. The sound power radiated (Figure 9.46) has peaks corresponding to the structural vibration modes that have a non-zero net volume velocity.

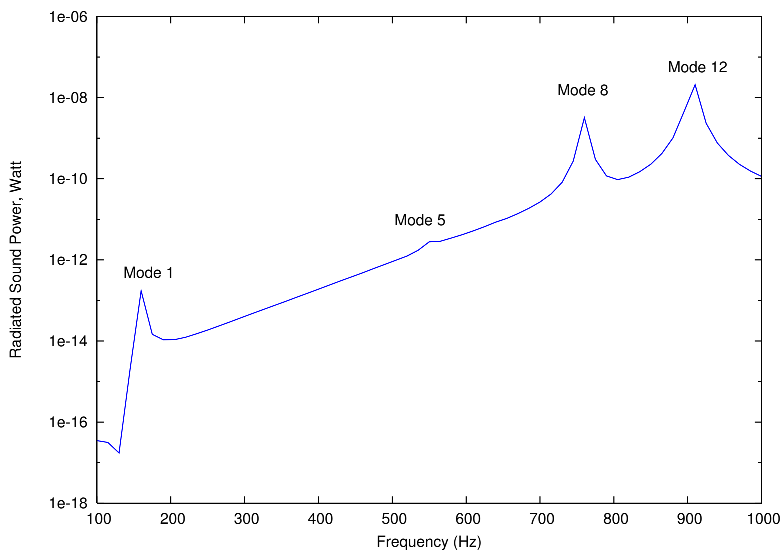


Figure 9.46: Sound power from the forced vibration response.

9.5.3.4. iglass.igl

IGlass files are post-processing data files created by Coustyx to visualize the acoustic analysis results. Refer to Figure 9.48.

- Double-click on 'iglass.igl' file to open it.
- Click on the **Attribs** tab on the top-left of the iglass viewer.

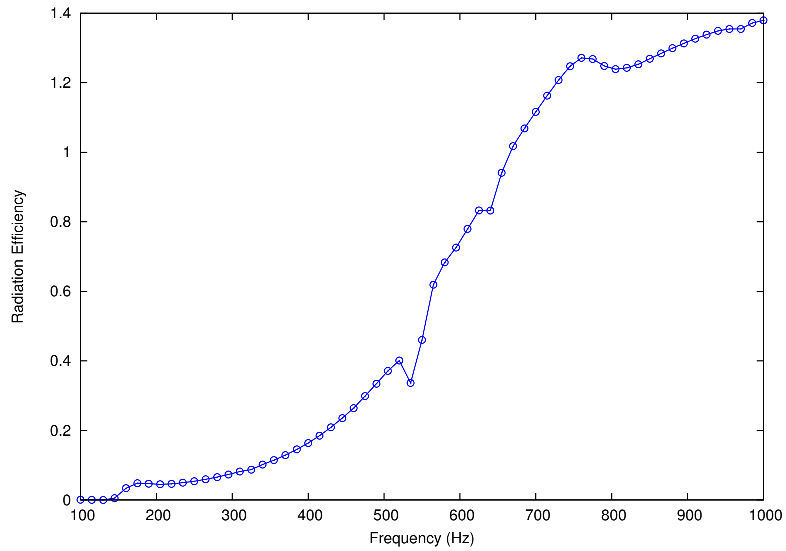


Figure 9.47: Radiation efficiency of the gearbox forced vibration.

- Select: **Attribs** → **Attribute** → **Surface Pressure Plus** to view the pressure distribution on the exterior surface of the gearbox housing.
- Click on the **View** tab on the top-left of the iglass viewer.
- Press the slider under **View** → **Phase**, to start animation. This activates the animation of the wave propagation on the housing surface.
- To view the results for different frequencies press the slider under **View** → **Frequency**.

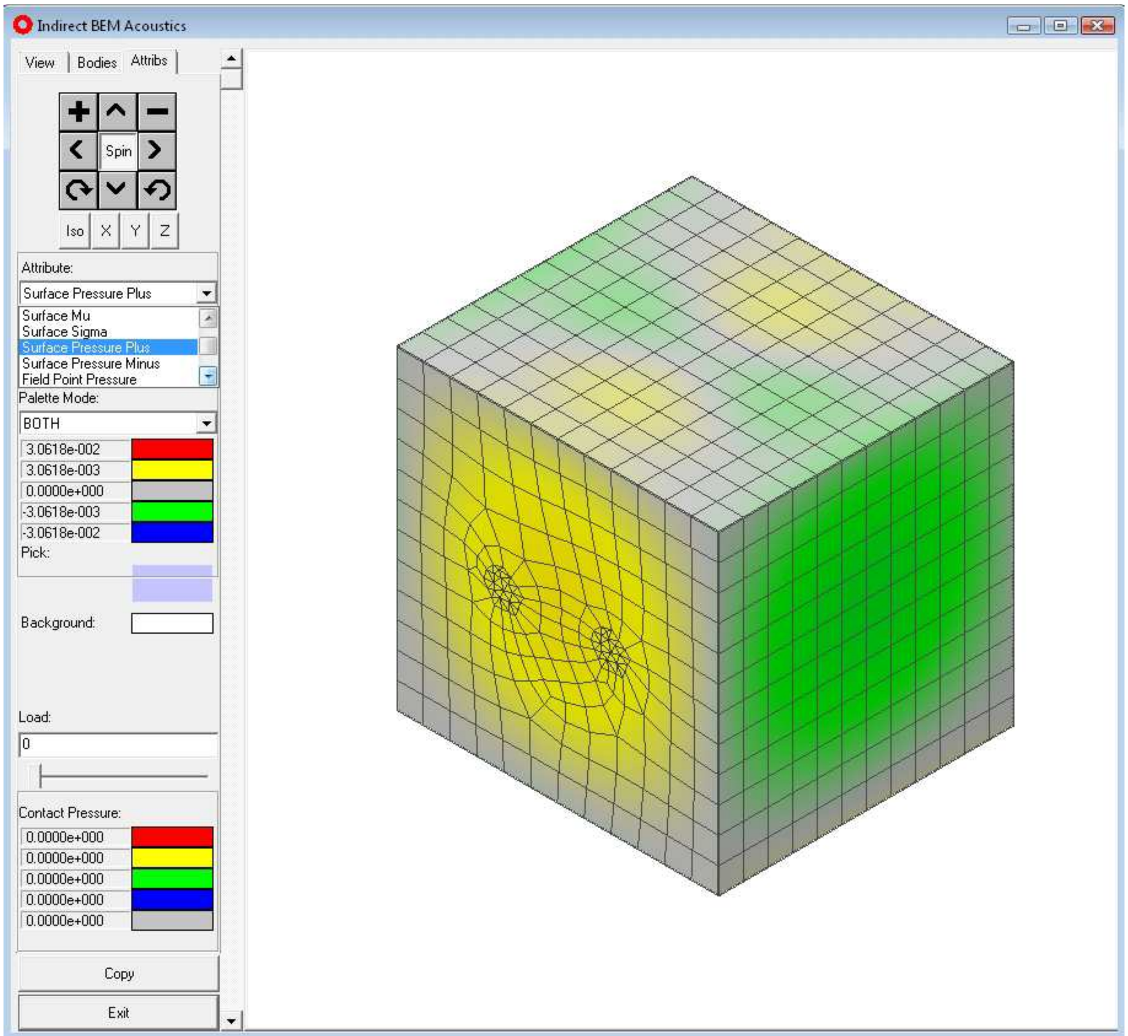


Figure 9.48: IGlass viewer showing sound pressure distribution on the exterior surface of the housing at 760 Hz.

Bibliography

[1] A. Seybert, T. W. Wu, and X. F. Wu. Experimental validation of finite element and boundary element methods for predicting structural vibration and radiated noise. Technical report, NASA Contractor Report 4561, 1994.