

---

# ***COUSTYX***

## **User's Manual**

---

**Advanced Numerical Solutions**  
**August 2022**



# CONTENTS

---

List of Figures	xiii
List of Tables	xxiii
Preface	xxv
<b>1 Introduction</b>	<b>1</b>
<b>2 Installing Coustyx</b>	<b>3</b>
2.1 Software Installation	3
2.1.1 License Agreement	3
2.1.2 Select Installation Folder	3
2.1.3 Confirm Installation	5
2.2 Dongle Device Driver Installation	5
2.3 License Key Installation	5
2.3.1 Local License Type	5
2.3.1.1 If you have a local dongle	5
2.3.1.2 If you are using <i>Coustyx</i> without a dongle	5
2.3.1.3 If you are using a Demo Model	5
2.3.2 Network Dongle	10
2.3.3 Ansol Server	10
2.3.4 Altair HyperWorks	13
2.3.5 Modify Path Environment Variable	13
2.4 Running Coustyx	13
2.4.1 Running from the Command (DOS) Prompt	13
<b>3 Conventions in Coustyx</b>	<b>17</b>

3.1	Time Dependence	17
3.2	Units	17
3.2.1	Model Units	17
3.2.2	How to choose model units?	18
3.3	Frequency Dependence Type	20
3.3.1	Constant	21
3.3.2	Table	21
3.3.2.1	Import from File	22
3.3.2.2	Import Options	22
3.3.3	Script	24
<b>4</b>	<b>Getting Started</b>	<b>27</b>
4.1	Main Menu Features	27
4.2	Operations on Mesh Viewer Window	31
4.2.1	GUI Control Panel Tools	35
4.2.2	Rotate, Pan & Selection Operations	36
4.2.3	Operations on Selection in GUI	36
4.2.3.1	Operations on Displayed Elements	36
4.2.3.2	Operations on Displayed Nodes	39
4.2.3.3	Operations on Displayed Faces	40
4.2.4	Selected Coord Nodes	42
4.2.5	Selected Elements	42
4.2.6	Fill Hole	43
4.2.7	Skin	43
4.2.8	New Element	43
4.2.9	Stitch Seams	43
4.2.10	Delete Elements	43
4.2.11	Merge Nodes	46
4.2.12	Split Pn Nodes / Split Sigma Nodes	46
4.2.13	Element Orientation	46
4.3	Model Setup	46
4.3.1	New Model/Open Model File	46
4.3.1.1	MultiDomain Model	46
4.3.1.2	Indirect Model	46
4.3.2	Structures	51
4.3.3	Materials	52
4.3.3.1	Speed of Sound	52
4.3.3.2	Ambient Density	53
4.3.4	Planes	54
4.3.5	Interfaces	55
4.3.6	Boundary Conditions	55
4.3.7	Direct BE Meshes	56
4.3.7.1	Coord Nodes	56
4.3.7.2	Elements	56
4.3.7.3	P Nodes	57
4.3.7.4	Pn Nodes	57



4.3.7.5	Constraint Equations	58
4.3.7.6	Sets	58
4.3.8	Indirect BE Mesh	58
4.3.8.1	Material	58
4.3.8.2	Coord Nodes	58
4.3.8.3	Elements	60
4.3.8.4	Sigma Nodes	60
4.3.8.5	Mu Nodes	60
4.3.8.6	Pn Nodes	60
4.3.8.7	Boundary Conditions	60
4.3.8.8	Sources	62
4.3.8.9	Jump Conditions	62
4.3.8.10	Junction Constraints	62
4.3.8.11	Sets	63
4.3.9	Domains	63
4.3.9.1	Sources	64
4.3.9.2	Boundedness	64
4.3.9.3	Material	65
4.3.9.4	Direct BE Meshes	65
4.3.9.5	Chief Points	68
4.3.10	Context Script	69
4.3.11	Analysis Sequences	69
4.4	Sets	71
4.4.1	Add elements, coordinate nodes, or faces to a Set	76
4.5	Acoustic Sources	77
4.5.1	Monopole	77
4.5.2	Plane Wave	78
4.5.3	Cylindrical	78
4.5.4	Dipole	80
4.5.5	Quadrupole	81
4.5.6	User Defined	82
<b>5</b>	<b>Pre-processing Features</b>	<b>85</b>
5.1	Importing FE Data	85
5.1.1	Importing Structure Mesh	86
5.1.1.1	Abaqus *.odb file	86
5.1.1.2	Abaqus *.inp files	87
5.1.1.3	Nastran Bulk Data *.bdf files	87
5.1.1.4	Nastran OP2 *.op2 files	87
5.1.1.5	Ansys Results *.rst files	87
5.1.1.6	I-DEAS Universal *.unv files	87
5.1.1.7	RadioSS Bulk Data *.fem files	87
5.1.1.8	Polygon File Format *.ply	89
5.1.2	Loading Frequency Response Data	89
5.1.2.1	Abaqus odb File	89
5.1.2.2	Nastran OP2 File	89

5.1.2.3	Nastran Punch File	89
5.1.2.4	Ansys rst File	89
5.1.2.5	Ansys rfrq File	90
5.1.2.6	I-DEAS Universal File	90
5.1.3	Loading Natural Mode Data	90
5.1.3.1	Copy to Frequency Response Data	90
5.2	Exporting Mesh	96
5.3	Extract Active Nodes	96
5.3.1	Frequency Response Data File	97
5.3.2	Natural Mode Data File	97
5.3.3	Mismatched Mesh	98
5.4	Forced Response Analysis using Modal Superposition	98
5.4.1	Load Natural Mode Data	99
5.4.2	Modify Modal Damping	99
5.4.3	Apply Loads	102
5.4.4	Perform Modal Superposition	104
5.5	Fill Hole	104
5.5.1	Delaunay Triangulation Optimization Parameters	108
5.5.2	Fill Parameters	108
5.5.3	Procedure to Fill Hole	108
5.6	Skin	110
5.6.1	Free Edges in a FE Mesh	110
5.6.2	Create Seam	110
5.6.2.1	Procedure to Create Seams	113
5.6.3	Create Skin	113
5.6.4	Create Mesh from Skin	115
5.6.5	Treatment of Bad Elements	117
5.7	New Element	122
5.7.1	Procedure to Create a New Element	122
5.8	Stitch Seams	123
5.8.1	Set Parameters	123
5.8.2	Procedure to Stitch Seams	125
5.9	Delete Elements	125
5.10	Merge Nodes	129
5.11	Split Pn Nodes / Split Sigma Nodes	130
5.12	Element Orientation	131
<b>6</b>	<b>Boundary Conditions</b>	<b>137</b>
6.1	Impedance Definition in Coustyx	137
6.2	Interpolation Options for Mismatched Meshes	139
6.3	Structural Acoustic Coupling	140
6.4	Interfaces	141
6.4.1	Transparent Interface	143
6.4.2	Perforated Interface	143
6.4.2.1	Sullivan and Crocker	143
6.4.2.2	Transfer Impedance	144

6.4.2.3	Uniform Perforated Interface	144
6.4.2.4	Non-uniform Perforated Interface	144
6.4.3	Uniform Arbitrary Interface	144
6.4.4	Non-uniform Arbitrary Interface	146
6.5	Multi-Domain Model BCs	146
6.5.1	Dummy BC	148
6.5.2	Interface BC	148
6.5.2.1	Interface Name	148
6.5.2.2	Domain is on the Positive Side of this Interface	150
6.5.3	Uniform Pressure BC	150
6.5.4	Non-uniform Pressure BC	150
6.5.5	Uniform Normal Velocity BC	150
6.5.5.1	Use Impedance	150
6.5.6	Uniform Velocity BC	152
6.5.6.1	Use Impedance	152
6.5.7	Non-uniform Normal Velocity BC	152
6.5.7.1	Use Impedance	153
6.5.8	Structure Velocity BC	154
6.5.8.1	Structure Name	154
6.5.8.2	Structure Interface Name	154
6.5.8.3	Interpolation Options for Mismatched Meshes	154
6.5.8.4	Use Impedance	154
6.5.9	Arbitrary Uniform BC	154
6.5.10	Arbitrary Non-uniform BC	155
6.6	Indirect BE Model BCs	156
6.6.1	Transparent BC	157
6.6.2	Anechoic Termination BC	157
6.6.3	Perforated BC	158
6.6.3.1	Sullivan and Crocker	159
6.6.3.2	Transfer Impedance	160
6.6.3.3	Use Structure Velocity	160
6.6.3.4	Uniform Perforated BC	160
6.6.3.5	Non-uniform Perforated BC	160
6.6.4	Uniform Pressure (Continuous) BC	160
6.6.5	Non-uniform Pressure (Continuous) BC	160
6.6.6	Uniform Normal Velocity (Continuous) BC	162
6.6.7	Uniform Velocity (Continuous) BC	163
6.6.8	Non-uniform Normal Velocity (Continuous) BC	163
6.6.9	Structure Velocity (Continuous) BC	165
6.6.9.1	Structure Name	165
6.6.9.2	Structure Interface Name	165
6.6.9.3	Interpolation Options for Mismatched Meshes	165
6.6.10	Discontinuous BC	165
6.6.10.1	Don't care	166
6.6.10.2	Uniform Pressure	166
6.6.10.3	Non-uniform Pressure	166

6.6.10.4	Uniform Normal Velocity	166
6.6.10.5	Uniform Velocity	169
6.6.10.6	Non-uniform Normal Velocity	169
6.6.10.7	Structure Velocity	169
6.6.11	Uniform Arbitrary BC	170
6.6.12	Non-uniform Arbitrary BC	171
6.7	Applying BCs	173
6.7.1	Apply BCs directly to Elements	173
6.7.2	Apply BCs through Sets	173
References		179
<b>7</b>	<b>Analysis Sequences</b>	<b>181</b>
7.1	Sequence Type	182
7.2	Inputs	187
7.2.1	Solver Controls	187
7.2.1.1	Parallel Processing	188
7.2.1.2	Formulation Type	188
7.2.1.3	FMM	188
7.2.1.4	Solution Method	189
7.2.1.5	GMRES	191
7.2.1.6	Integration	196
7.2.2	Frequency Ranges (Frequency Filter)	198
7.2.2.1	Coustyx Results File	199
7.2.2.2	ASCII File	200
7.2.2.3	Frequency Response Data File	200
7.2.2.4	Natural Mode Data File	201
7.2.2.5	Structure Frequency Response Data	202
7.2.2.6	Structure Natural Mode Data	202
7.2.2.7	Octave-Band Center Frequencies	203
7.2.2.8	1/3 Octave-Band Center Frequencies	206
7.2.2.9	1/6 Octave-Band Center Frequencies	206
7.2.2.10	1/12 Octave-Band Center Frequencies	207
7.2.2.11	1/24 Octave-Band Center Frequencies	207
7.2.3	Just-In-Time Loading	207
7.2.4	Diffuse Sound Field	207
7.2.4.1	Input Power Spectral Density	207
7.2.4.2	Uncorrelated Plane Waves	211
7.3	Outputs	213
7.3.1	General Options	213
7.3.1.1	Sound Spectrum Type	213
7.3.1.2	FMM for Field Point Calculations	215
7.3.1.3	Reference Pressure	215
7.3.1.4	Reference Power	215
7.3.1.5	Weighting Filters	216
7.3.2	Binary Results	216
7.3.2.1	Create a Binary Results File	216

7.3.3	Sensors	218
7.3.3.1	Create an Ascii Sensors File	218
7.3.3.2	Update Domains	218
7.3.3.3	Import From File	218
7.3.3.4	Panel Selection	218
7.3.3.5	Save Pressure in Decibels	220
7.3.3.6	Save Overall Sound Pressure Level	220
7.3.3.7	Save Sound Pressure Level in Octave Band	220
7.3.4	Sound Power	220
7.3.4.1	Create a Sound Power File	221
7.3.4.2	Domain Name	221
7.3.4.3	Save Power in Decibels	221
7.3.4.4	Save Overall Sound Power Level	221
7.3.4.5	Save Sound Power Level in Octave Band	221
7.3.4.6	Default	221
7.3.4.7	Field Mesh	223
7.3.4.8	ISO Standards	223
7.3.5	Muffler Transmission Loss	244
7.3.5.1	File Name	244
7.3.5.2	Inlet Pipe	244
7.3.5.3	Outlet Pipe	244
7.3.5.4	Si/So	244
7.3.6	Acoustic Field Response	246
7.3.6.1	Response Files	247
7.3.6.2	Field Point Grids	247
7.3.6.3	Output Variables	251
7.3.7	Structure Response	254
7.4	Script	254
7.4.1	Model Setup Commands	258
7.4.2	Solver Options Setup Commands	261
7.4.3	Pre-Processing Commands	265
7.4.4	Post-Processing Commands	274
References		283
<b>8</b>	<b>Language Syntax</b>	<b>285</b>
8.1	Comments	285
8.2	Constants	285
8.2.1	Built-in Constants	286
8.3	Variables	286
8.3.1	Predefined Variables	286
8.4	Function Calls	287
8.5	Expressions:	287
8.6	Statements	288
8.6.1	Declaration Statement	288
8.6.2	Expression Statement	288
8.6.3	Assignment Statement	288

8.6.4	Declaration with Assignment	288
8.6.5	Compound Statement	289
8.6.6	Symbolic Form and Evaluation of Expressions	289
8.6.7	if Statement	290
8.6.8	for Statement	291
8.6.9	while Statement	292
8.6.10	do-while Statement	292
8.6.11	break Statement	293
8.6.12	continue Statement	294
8.6.13	switch Statement	294
8.6.14	Variable Scope (Visibility) Rules	296
8.6.15	Function Definition	297
8.6.16	try-catch and throw Statements	300
8.6.17	Statement Label	301
8.6.18	Goto Statement	301
8.7	End of Input	302
8.8	Grammar	302
8.9	IO Functions Syntax	302
8.10	Regular Functions Syntax	306
8.11	Special Functions Syntax	306
8.11.1	Associated Legendre Function	306
8.11.2	Legendre Polynomials	307
8.11.3	Spherical Harmonic	307
8.11.4	Spherical Hankel Function of First Kind	307
8.11.5	Spherical Hankel Function of Second Kind	308
8.11.6	Cylindrical Hankel Function of First Kind	308
8.11.7	Cylindrical Hankel Function of Second Kind	308
8.11.8	Spherical Bessel Function	309
8.11.9	Cylindrical Bessel Function	309
<b>A</b>	<b>Tutorial – Gear Box Radiation</b>	<b>311</b>
A.1	Introduction	311
A.2	Problem Description	312
A.3	Finite Element Analysis	312
A.3.1	FE Mesh Modeling	313
A.3.2	Natural Frequencies and Normal Modes	313
A.3.3	Forced Response - Modal Superposition	314
A.4	Coustyx MultiDomain Model	314
A.4.1	Problem Setup	314
A.4.1.1	Create a New Model	314
A.4.1.2	Import FE Structure Mesh	316
A.4.1.3	Load Frequency Response Data	316
A.4.1.4	Generate BE Mesh	316
A.4.1.5	Define Material Properties	322
A.4.1.6	Fill Holes	323
A.4.1.7	Define Boundary Conditions	323

A.4.1.8	Apply Boundary Conditions	325
A.4.1.9	Domains	326
A.4.2	Run Acoustic Analysis	328
A.4.3	Post-processing/Outputs	332
A.4.3.1	results.dat	332
A.4.3.2	sensors.dat	332
A.4.3.3	power.dat	332
A.4.3.4	iglass.cgl	335
A.5	Coustyx Indirect Model	335
A.5.1	Problem Setup	335
A.5.1.1	Create a New Model	335
A.5.1.2	Import FE Structure Mesh	335
A.5.1.3	Load Frequency Response Data	335
A.5.1.4	Generate BE Mesh	339
A.5.1.5	Define Material Properties	342
A.5.1.6	Fill Holes	343
A.5.1.7	Define Boundary Conditions	343
A.5.1.8	Apply Boundary Conditions	345
A.5.2	Run Acoustic Analysis	346
A.5.3	Post-processing/Outputs	349
A.5.3.1	results.dat	349
A.5.3.2	sensors.dat	349
A.5.3.3	power.dat	349
A.5.3.4	iglass.cgl	349
References		353





## LIST OF FIGURES

---

2.1	Select path window.	4
2.2	Windows installer error message.	4
2.3	Install License Key window.	6
2.4	Local License Key window.	7
2.5	License Features window.	8
2.6	Invalid or expired License status.	9
2.7	Network Dongle window.	11
2.8	Ansol Server window.	12
2.9	Altair HyperWorks License.	14
2.10	Set path variable.	15
3.1	Units model tree member.	18
3.2	Edit units dialog box.	19
3.3	Boundary Condition Table.	21
3.4	Edit Boundary Condition Table.	22
3.5	Table Boundary Condition example.	23
3.6	Table import options window.	24
3.7	Script to define frequency dependent acoustic variable for the boundary condition.	25
4.1	Main menu items	32

4.2	Preferences dialog box - Common parameters	32
4.3	Preferences dialog box - 3D Viewer parameters	33
4.4	Definition of feature angle between two connected elements. Note: This definition is applicable only for 2D elements.	33
4.5	Coustyx with Mesh Viewer Window.	34
4.6	GUI Control Panel Tools	35
4.7	GUI operations on elements available through the context menu, activated by the right mouse button with the <i>shift-key</i> held down.	37
4.8	Element Display Style Window.	38
4.9	GUI operations available for selected nodes.	40
4.10	Nodes Display Style Window.	41
4.11	GUI operations on faces available through the context menu, activated by the right mouse button with the <i>shift-key</i> held down.	41
4.12	Fill Hole	44
4.13	Create Skin	45
4.14	Stitch Seams	45
4.15	Coustyx model tree structure.	47
4.16	New model dialog box.	48
4.17	Variable Nodes in Indirect BEM	49
4.18	Variable Nodes in Indirect BEM	50
4.19	Create New Material	52
4.20	Edit Material Properties	53
4.21	Effect of complex speed of sound, $c = c_r + jc_i$ , on the pressure variation with distance from a point source.	53
4.22	Creating a new plane.	54
4.23	Descriptions for different types of planes. Note arrows represent velocity vectors at those points.	55
4.24	Selecting Material	59
4.25	Create Boundary Conditions	61
4.26	Create New Source	62
4.27	Junction constraints	63
4.28	Choosing Boundedness	64
4.29	Transformation Matrix Edit Dialog Box.	66
4.30	Rotational Matrix Edit Dialog Box.	66
4.31	Side of Mesh on Domain Function	67

4.32	Chief Points Edit Dialog Box.	68
4.33	Import Chief Points from a File.	69
4.34	Context Script	70
4.35	Creating a new set, a group of elements and nodes.	72
4.36	Menu options for elements in a Set.	73
4.37	Menu options for coordinate nodes in a Set.	74
4.38	Menu options for faces in a Set.	75
4.39	Add elements to a set.	76
4.40	Monopole	77
4.41	Description of a plane wave acoustic source.	78
4.42	Plane Wave	79
4.43	Description of a cylindrical wave acoustic source.	79
4.44	Cylindrical	80
4.45	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 $+V$ .	81
4.46	Dipole	82
4.47	Quadrupole Dialog Box	83
4.48	User Defined Acoustic Source	84
5.1	Importing a finite element structure mesh.	86
5.2	Odb Reader Import Requirements Dialog Box.	87
5.3	Odb Reader Import Options Dialog Box.	88
5.4	Odb File Upgrade Dialog Box.	89
5.5	Loading frequency response data into <i>Coustyx</i> model.	90
5.6	Select frequencies to load dialog box.	91
5.7	Loading natural mode data into <i>Coustyx</i> model.	92
5.8	Select natural frequencies to load dialog box.	93
5.9	Copy the mode data to the frequency response data.	94
5.10	Alert message displayed before copying mode data to frequency response data.	95
5.11	Export mesh.	95
5.12	Extract active nodes menu.	96
5.13	Extract active nodes & Create new mesh dialog with frequency response data file as the source.	97
5.14	Extract active nodes & Create new mesh dialog with natural mode data file as the source.	98
5.15	Extract active nodes & Create new mesh dialog with mismatched mesh as the source.	99

5.16	Damping ratios edit dialog box.	100
5.17	Select modes by frequency range.	101
5.18	Edit selected modal damping ratios using Frequency dependent values.	102
5.19	Edit selected modal damping ratios using Rayleigh damping coefficients.	103
5.20	Apply nodal force and moment.	103
5.21	Perform modal superposition.	105
5.22	Participating modes selection.	106
5.23	Coustyx GUI - Fill Hole.	107
5.24	Fill Hole Parameters.	107
5.25	Display Connected Nodes	109
5.26	Mesh with free edges shown in <i>blue</i> (color).	111
5.27	Create Seams.	112
5.28	Select Elements for Creating a Seam.	113
5.29	Display Connected Nodes for Selected Elements.	114
5.30	Choose Nodes for Creating Seams.	114
5.31	Accept Seam.	115
5.32	Create Skin.	116
5.33	Press <b>Skin</b> → <b>Create Skin</b> to check for incompatible elements.	117
5.34	Select all Bad Elements.	118
5.35	Add Bad Elements to a Set.	119
5.36	Display Set of Bad Elements.	120
5.37	Display Connected Nodes of Selected Element.	121
5.38	Display Connected Elements of Selected Node.	121
5.39	Create new element window.	123
5.40	New elements coordinate node connectivity.	124
5.41	Stitch Seam Dialog Panel	124
5.42	Select Elements.	126
5.43	Display Connected Nodes for Elements.	126
5.44	Seam 1.	127
5.45	Seam 2.	127
5.46	Transparent elements shown for inspection before Stitching Seams.	128
5.47	Complete Stitched Seam.	128
5.48	Delete element window.	129

5.49	Merge Nodes window.	129
5.50	Split Pn Nodes (Split Sigma Nodes) window showing duplicate nodes along edges. Note: During skinning, <i>Coustyx</i> automatically creates duplicate nodes for most common cases, such as geometry discontinuities, edges, corners, junctions etc.	130
5.51	After adding elements to Group 1. Notice that the elements belonging to Group 1 are displayed in red.	131
5.52	After adding elements to Group 2. Notice that the elements belonging to Group 2 are displayed in blue.	132
5.53	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.	132
5.54	Before splitting a selected node between Group 1 and Group 2.	133
5.55	After splitting a selected node between Group 1 and Group 2.	133
5.56	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).	133
5.57	After splitting the shared node among Group 1 (Element A), Group 2 (Element B) and unselected elements (Elements C and D).	134
5.58	Element orientations window.	135
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.	138
6.2	Specific acoustic impedance $Z/\rho_0 c = R/\rho_0 c + jX/\rho_0 c$ for a foam of 1-inch thickness measured using $e^{-j\omega t}$ convention.	138
6.3	Interpolation for mismatched meshes.	139
6.4	Coupling type.	141
6.5	New Interface.	142
6.6	Transparent Interface.	143
6.7	Uniform Perforated Interface	145
6.8	Non-Uniform Perforated Interface	145
6.9	Uniform Arbitrary Interface	146
6.10	Sample script for the function <code>GetAlphaBetaGamma()</code> .	147
6.11	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 <i>MultiDomain</i> model are defined with respect to the domain normal.	147
6.12	New Boundary Condition.	148
6.13	Dummy Boundary Condition.	149
6.14	Interface Boundary Condition.	149
6.15	Uniform Pressure BC.	150

6.16	Non-Uniform Pressure BC.	151
6.17	Uniform Normal Velocity BC.	151
6.18	Uniform Velocity BC.	152
6.19	Non-Uniform Normal Velocity BC.	153
6.20	Structure Velocity BC.	154
6.21	Arbitrary Uniform BC	155
6.22	Arbitrary Non-Uniform BC	156
6.23	Description of element normal and its sides.	157
6.24	Transparent BC	158
6.25	Anechoic Termination BC	158
6.26	Perforated Plate	158
6.27	Uniform Perforated	159
6.28	Non-Uniform Perforated	161
6.29	Uniform Pressure (Continuous)	161
6.30	Non-Uniform Pressure (Continuous)	162
6.31	Uniform Normal Velocity (Continuous)	162
6.32	Uniform Velocity (Continuous)	163
6.33	Non-Uniform Normal Velocity (Continuous)	164
6.34	Structure Velocity (Continuous)	164
6.35	Discontinuous BC (Don't Care)	165
6.36	Discontinuous Uniform Pressure	166
6.37	Discontinuous Non-Uniform Pressure	167
6.38	Discontinuous Uniform Normal Velocity	167
6.39	Discontinuous Uniform Velocity	168
6.40	Discontinuous Non-Uniform Normal Velocity	169
6.41	Structure Velocity	170
6.42	Uniform Arbitrary BC	170
6.43	Sample script for the function GetAlphaBetaGamma().	171
6.44	Alpha Frequency Dependence Script	172
6.45	Beta Frequency Dependence Script	172
6.46	Gamma Frequency Dependence Script	172
6.47	Applying boundary conditions through elements.	174
6.48	Creating a new set, a group of elements and nodes.	175

6.49	Add elements to a set.	176
6.50	Applying boundary conditions through sets.	177
7.1	Creating new analysis sequence.	182
7.2	Analysis sequence edit dialog box.	183
7.3	Run new analysis sequence.	184
7.4	Abort the analysis run.	185
7.5	Abort message.	185
7.6	Generate command script.	186
7.7	Generate command script alert message.	186
7.8	Solver controls window.	187
7.9	Suggested number of FMM levels.	190
7.10	Computational cell hierarchy constructed at different FMM levels.	191
7.11	Solution Method FMM	192
7.12	GMRES	193
7.13	Direct Solution Method	194
7.14	Variable Order Integration Scheme	197
7.15	Frequency ranges window.	198
7.16	Frequency filter window.	199
7.17	Load frequencies from Coustyx Results file window.	199
7.18	Load frequencies from an ASCII file window.	200
7.19	Load frequencies from frequency response data file.	200
7.20	Load frequencies from the natural mode data file.	201
7.21	Load frequencies from the structure frequency response data.	202
7.22	Load frequencies from the structure natural mode data.	202
7.23	Load center frequencies of an octave band.	206
7.24	Just-In-Time Loading window.	208
7.25	Diffuse Sound Field: Input Power Spectral Density tab.	209
7.26	Load PSD from SPL window.	210
7.27	Diffuse sound field modeled as a superposition of uncorrelated planewaves.	212
7.28	Uncorrelated plane waves selection through User Options.	212
7.29	Uncorrelated plane waves selection through Table.	213
7.30	General Options window.	214
7.31	Relative response plots for different weighting filters.	216

7.32	Sensor output window.	219
7.33	Sensor Update Domain window.	219
7.34	Import Sensor Coordinates from a File.	220
7.35	ISO3744 sound power standard window.	224
7.36	ISO3744 - Microphone array on the hemisphere.	225
7.37	Coaxial circular paths in parallel planes for microphone traverses over a reflecting plane.	227
7.38	ISO3744 - Microphone array on the quadrant.	228
7.39	ISO3744 - Microphone array on the octant.	229
7.40	ISO3744 - Procedure for fixing microphone positions on a parallelepiped measurement surface.	230
7.41	ISO3744 - Parallelepiped measurement surface with microphone positions.	230
7.42	ISO3744 - Parallelepiped measurement surface with microphone positions for a source placed on the floor against a wall.	232
7.43	ISO3744 - Parallelepiped measurement surface with microphone positions for a source placed on the floor against two walls.	233
7.44	ISO3745 sound power standard window.	235
7.45	ISO3745 - Meridional paths for a moving microphone.	237
7.46	ISO3745 - Spiral path for a moving microphone.	238
7.47	ISO9614-1 sound power standard window.	241
7.48	ISO9614-1 - Parallelepiped measurement surface with probe positions.	243
7.49	Muffler Transmission Loss window.	245
7.50	Acoustic Field Response window.	246
7.51	Select field point grids window.	246
7.52	Field Point Grids tab.	248
7.53	IGlass field point grid types - Quadrilateral, Triangle, Annular Disc, Box, and Sphere.	249
7.54	IGlass viewer showing sound pressure distribution on the exterior surface of a housing.	252
7.55	Structure Response window.	255
7.56	Analysis script.	256
A.1	Detail of the NASA Lewis gearbox.	312
A.2	FE model of the NASA Lewis gearbox.	313
A.3	Surface normal velocity distribution for the first five free vibration modes.	315
A.4	New Model Selection Window.	316
A.5	Import a finite element structure mesh.	317
A.6	Load frequency response data into Coustyx.	318



A.7	Structure mesh opened in Coustyx GUI.	319
A.8	Coustyx GUI control panel tools.	319
A.9	Select all displayed elements in the GUI.	320
A.10	Element Display Style Window.	320
A.11	Select elements for creating a seam.	321
A.12	Display connected nodes for creating a seam.	321
A.13	Pick nodes to create a seam.	322
A.14	Edit material properties.	323
A.15	Fill hole using triangle elements.	324
A.16	Edit boundary conditions window.	324
A.17	Edit structure velocity boundary condition window.	325
A.18	Apply boundary conditions through selected elements.	326
A.19	Apply boundary conditions through sets.	327
A.20	Specifying boundedness of the domain.	328
A.21	Element orientations.	329
A.22	Set the side of the mesh on which the domain is.	330
A.23	Analysis solver controls.	331
A.24	Set analysis frequencies.	332
A.25	Sound power from the forced vibration response.	333
A.26	Radiation efficiency of the gearbox forced vibration.	333
A.27	IGlass viewer showing sound pressure distribution at 760 Hz.	334
A.28	New Model Selection Window.	336
A.29	Import a finite element structure mesh.	336
A.30	Load frequency response data into Coustyx.	337
A.31	Structure mesh opened in Coustyx GUI.	338
A.32	Coustyx GUI control panel tools.	339
A.33	Select all displayed elements in the GUI.	340
A.34	Element Display Style Window.	340
A.35	Select elements for creating a seam.	341
A.36	Display connected nodes for creating a seam.	341
A.37	Pick nodes to create a seam.	342
A.38	Edit material properties.	343
A.39	Fill hole using triangle elements.	344

A.40	Edit boundary conditions window.	344
A.41	Edit structure velocity boundary condition window.	345
A.42	Apply boundary conditions through selected elements.	346
A.43	Apply boundary conditions through sets.	347
A.44	Analysis solver controls.	348
A.45	Set analysis frequencies.	349
A.46	Sound power from the forced vibration response.	350
A.47	Radiation efficiency of the gearbox forced vibration.	350
A.48	IGlass viewer showing sound pressure distribution on the exterior surface of the housing at 760 Hz.	351

## LIST OF TABLES

---

3.1	Length scale factors.	19
3.2	Mass scale factors.	19
3.3	Variables and Units	20
3.4	Values for acoustic medium properties of air in different units	21
4.1	Description of the main menu bar items in <i>Coustyx</i> UI	27
4.2	Description of GUI control panel tools	35
4.3	Differences between MultiDomain and Indirect models	51
5.1	Some of the common types of Bad Elements and their Treatment.	122
7.1	List of valid integration orders and the corresponding quadrature points on triangle elements.	196
7.2	List of the integration orders and the corresponding quadrature points on quadrilateral elements. Note, the total number of quadrature points are $npts \times npts$	196
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.	198
7.4	Comparison of preferred center frequencies for fractional octave-bands between 1000–10000 Hz.	203
7.5	Relative response levels for various weightings ([1]).	217
7.6	ISO3744 - Coordinates of key microphone positions (1-10) and additional microphone positions (11-20) on a hemisphere ([2]).	226
7.7	ISO3744 - Coordinates of microphone positions for sources emitting discrete tones.	226
7.8	ISO3745 - Coordinates of microphone positions on a sphere ([3]).	236

7.9	ISO3745 - Coordinates of microphone positions on a hemisphere ([3]).	239
8.1	Coustyx Language Grammar	303
8.2	Coustyx Language Grammar (contd.)	304
A.1	Physical properties of materials used in FEA.	313
A.2	Gearbox natural frequencies from FEA.	314

## 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(s) and do not necessarily reflect the views of the NSF.

Advanced Numerical Solutions, LLC  
Hilliard, OH



# 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 formulations robust to deal with several problems associated with BE methods, such as Thin Shape Breakdown (TSB), Irregular frequencies (or non-uniqueness issue), ability to model single

sided surfaces such as Möbius strip, handle extremely complex junctions and inconsistent mesh normal orientations. Independent interpolation schemes for coordinates and acoustic variables provide greater modeling flexibility.

3. *Coustyx* allows a 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 results from external 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 A 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://ansol.us/Products/Coustyx/> and follow the downloading instructions provided on the web site. *Coustyx* is compatible with Windows 7, Vista, XP, and 2000. For installation on linux machines contact us at [sales@ansol.com](mailto: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

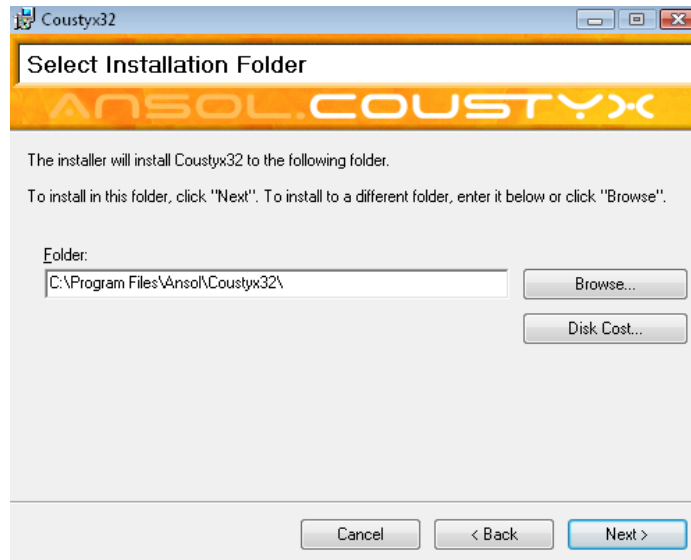
By default, the folder selected to install *Coustyx* is:

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

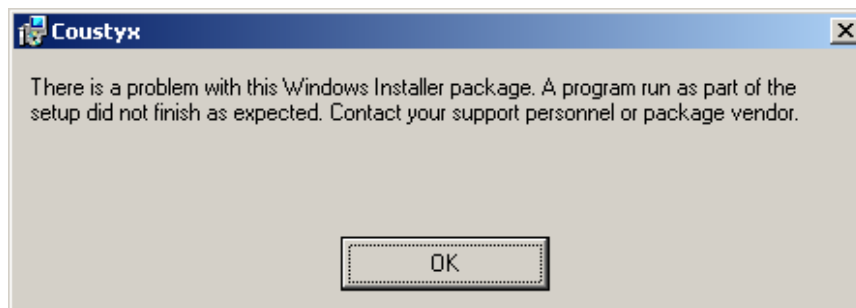
You can change the installation folder by browsing through the folders using “Browse” button. The “Disk Cost” button lists all the available drives on your computer to which *Coustyx* can be installed, along with each drive’s available and required disk space.(Figure 2.1)

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

## 4 INSTALLING COUSTYX



**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 licensing schemes listed in the drop down menu. Select the option **Local** when you have a valid local license key or local dongle provided by ANSOL. Select **Altair Hyperworks** when you want to use Altair Hyperworks License management system. If you are using a network license select either **Network Dongle** or **Ansol Server**.

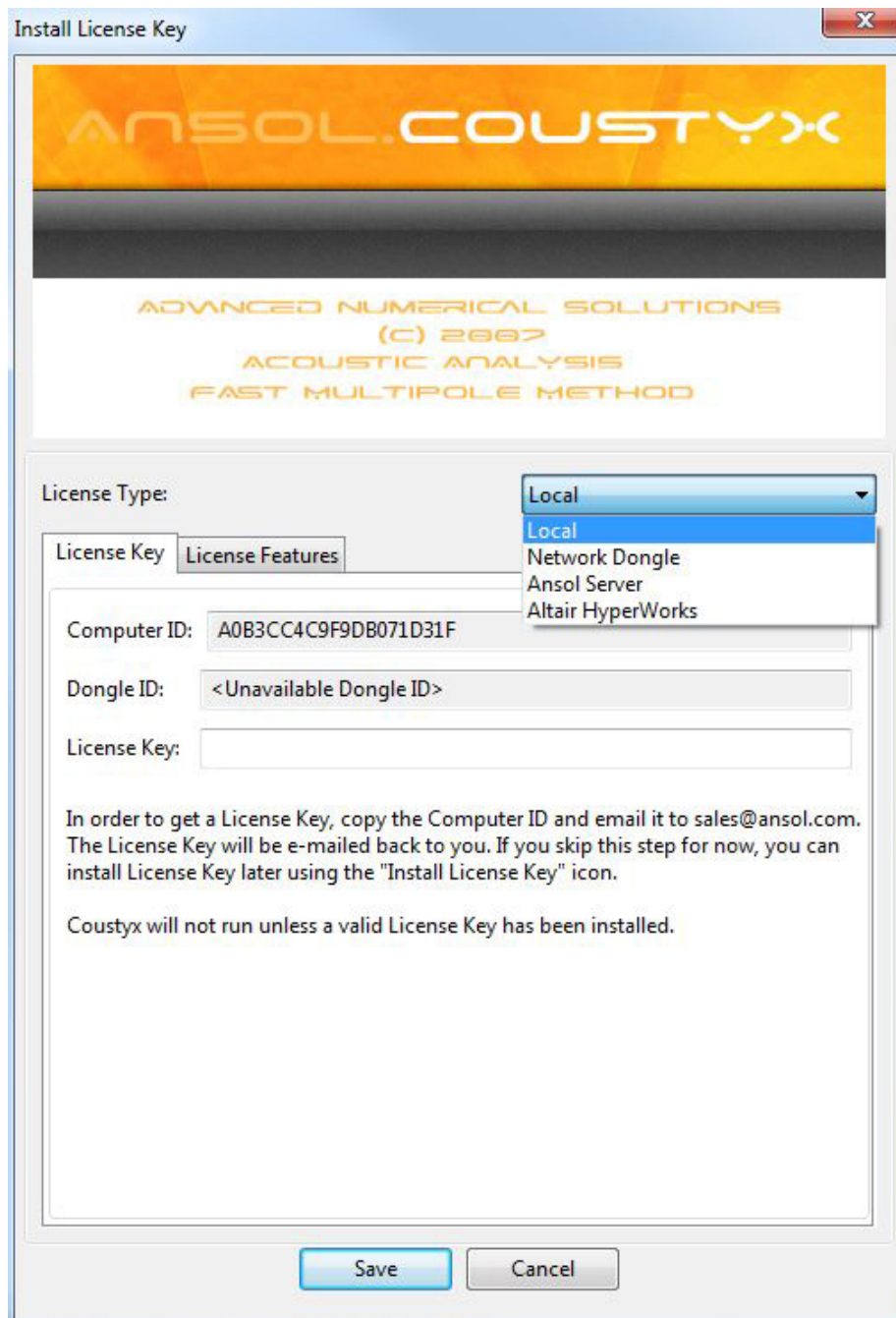
### 2.3.1. Local License Type

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 an 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 2.5). If the license key is invalid or expired, appropriate information is highlighted (Figure 2.6). 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 “Save” to install the license key and exit the window, or press “Cancel” to discard changes before exiting the window (Figure 2.4).

**2.3.1.2. If you are using Coustyx without a dongle** Copy the Computer ID (Figure 2.4) and email it to [sales@ansol.com](mailto: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. Select the License Type to be **Local**, then 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 2.5). If the license key is invalid or expired, appropriate information is highlighted (Figure 2.6). 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 “Save” to install the license key and exit the window, or press “Cancel” to discard changes before exiting the window (Figure 2.4).

**2.3.1.3. 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.

**Figure 2.3** Install License Key window.

Install License Key

ANSOL.COUSTYX

ADVANCED NUMERICAL SOLUTIONS  
(C) 2007  
ACOUSTIC ANALYSIS  
FAST MULTIPOLE METHOD

License Type: Local

License Key | License Features

Computer ID: A0B3CC4C9F9DB071D31F

Dongle ID: <Unavailable Dongle ID>

License Key:

In order to get a License Key, copy the Computer ID and email it to sales@ansol.com. The License Key will be e-mailed back to you. If you skip this step for now, you can install License Key later using the "Install License Key" icon.

Coustyx will not run unless a valid License Key has been installed.

Save Cancel

**Figure 2.4** Local License Key window.

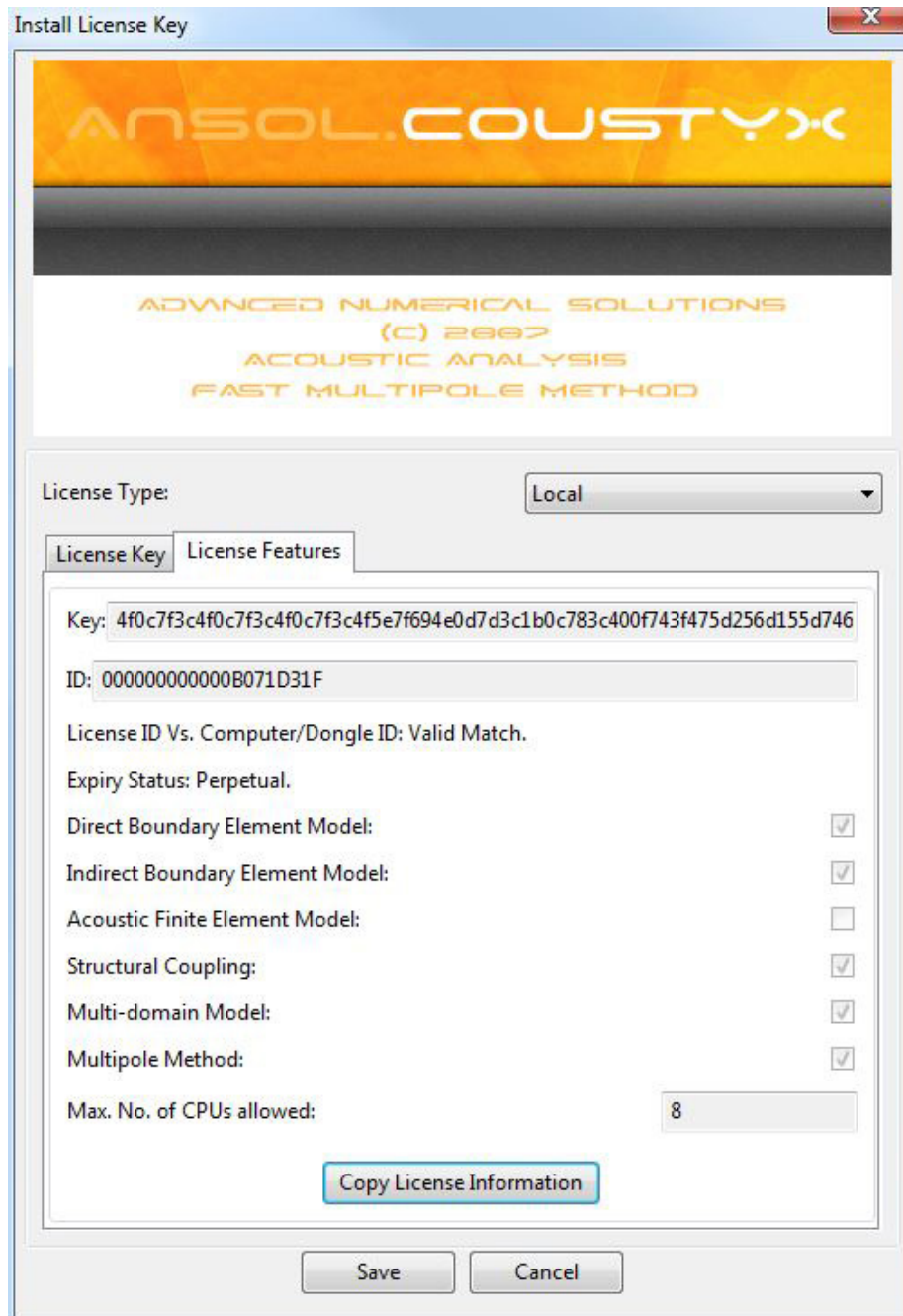
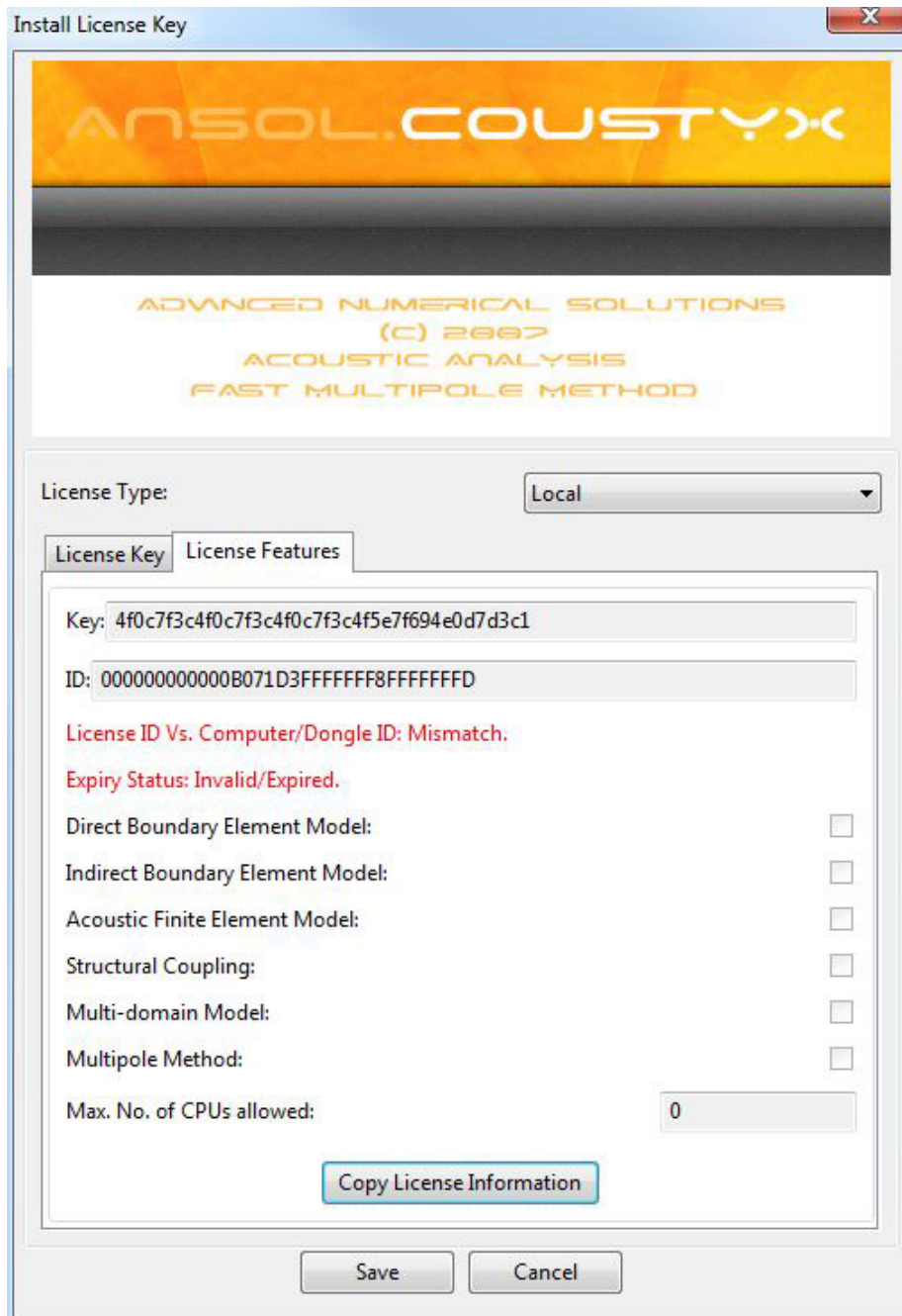


Figure 2.5 License Features window.



**Figure 2.6** Invalid or expired License status.

### 2.3.2. 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 type **Network Dongle**. From the tabbed window “License Key”, 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 (Figure 2.7). 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 2.5). If the license key is invalid or expired, appropriate information is highlighted (Figure 2.6). 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 “Save” to install the key on to the network dongle and exit the window, or press “Cancel” to discard changes before exiting the window (Figure 2.4).

**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 “Save” to register and exit the window, or press “Cancel” to discard changes before exiting the window (Figure 2.4).

### 2.3.3. Ansol Server

To install **Ansol (License) Server** on your local network contact [sales@ansol.com](mailto:sales@ansol.com) for detailed instructions and relevant installation files. Once Ansol License Server installation is complete, you can access the available floating licenses from any client connected to the server network.

To set the client use license through **Ansol Server**, provide the following information in the **Install License Key** window and press **Save**. Figure 2.8.

**Server Host Name** Provide the server name where Ansol license manager is installed.

**Token ID** Provide the token id provided for the current license. Contact your network administrator or Ansol for more information.

**Port** Enter the port number 17342. Ansol server communicates with the client through ports 17341 and 17342. Make sure your computer’s firewall does not block these two ports.

**Password** Enter the password provided for the current token id.

With the above information entered, switch to the “License Features” tab to verify that the connection is made as well as to view license details.



Install License Key

ANSOL.COUSTYX

ADVANCED NUMERICAL SOLUTIONS  
(C) 2002  
ACOUSTIC ANALYSIS  
FAST MULTIPOLE METHOD

License Type: Network Dongle

License Key License Features

Server Host Name: localhost Find Dongle

Dongle ID: Sentinel\_USB:0138CCB10E33C87468AB[1/Jul/2015]

License Key: 185b2968110b72694359266a15092d3e1908216943597069470d213d1

Coustyx will not run unless a valid License Key has been installed.

Save Cancel

Figure 2.7 Network Dongle window.

Install License Key

ANSOL.COUSTYX

ADVANCED NUMERICAL SOLUTIONS  
(C) 2002  
ACOUSTIC ANALYSIS  
FAST MULTIPOLE METHOD

License Type: Ansol Server

License Key License Features

Server Host Name:

Token ID:

Port:  Password:

Coustyx will not run unless a valid License Key has been installed.

Save Cancel

**Figure 2.8** Ansol Server window.

### 2.3.4. Altair HyperWorks

Coustyx has partnered with Altair to enable users run Coustyx under Altair's HyperWorks License system. For more information on HyperWorks 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 HyperWorks Units (HWUs), will be drawn from the Altair License server each time Coustyx is invoked. These HWUs are returned to the server once the user exists Coustyx. The number of HWUs 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 `LMX_LICENSE_PATH` or `ALTAIR_LICENSE_PATH` is set to `6200@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 type **Altair HyperWorks**. Press the **Test** button to check the connection to the license server and the number of available HyperWorks units (Figure 2.9). Press "Save" to install the license and exit the window, or press "Cancel" to discard changes before exiting the window.

### 2.3.5. 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. See Section 2.3.5 (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.

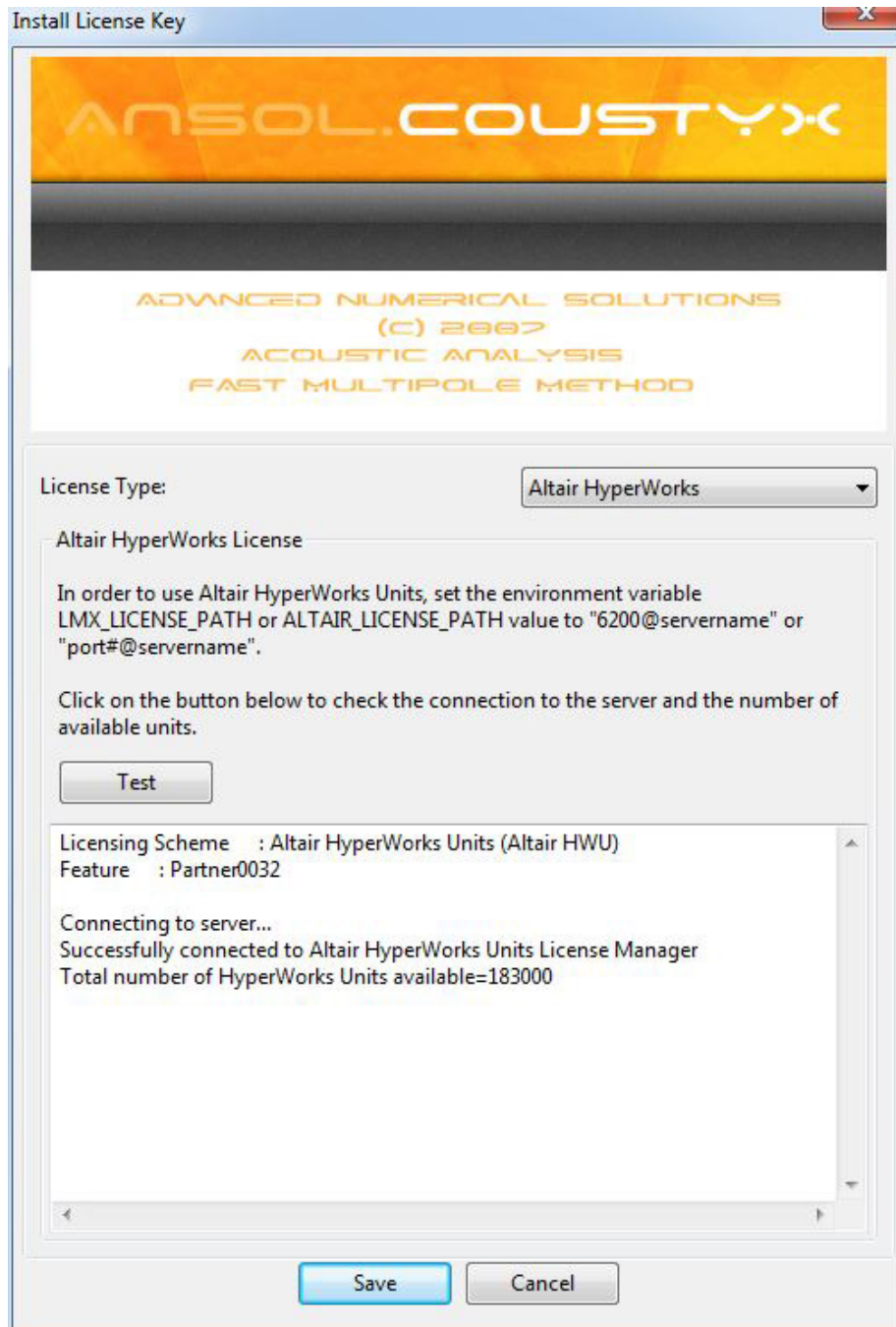


Figure 2.9 Altair HyperWorks License.

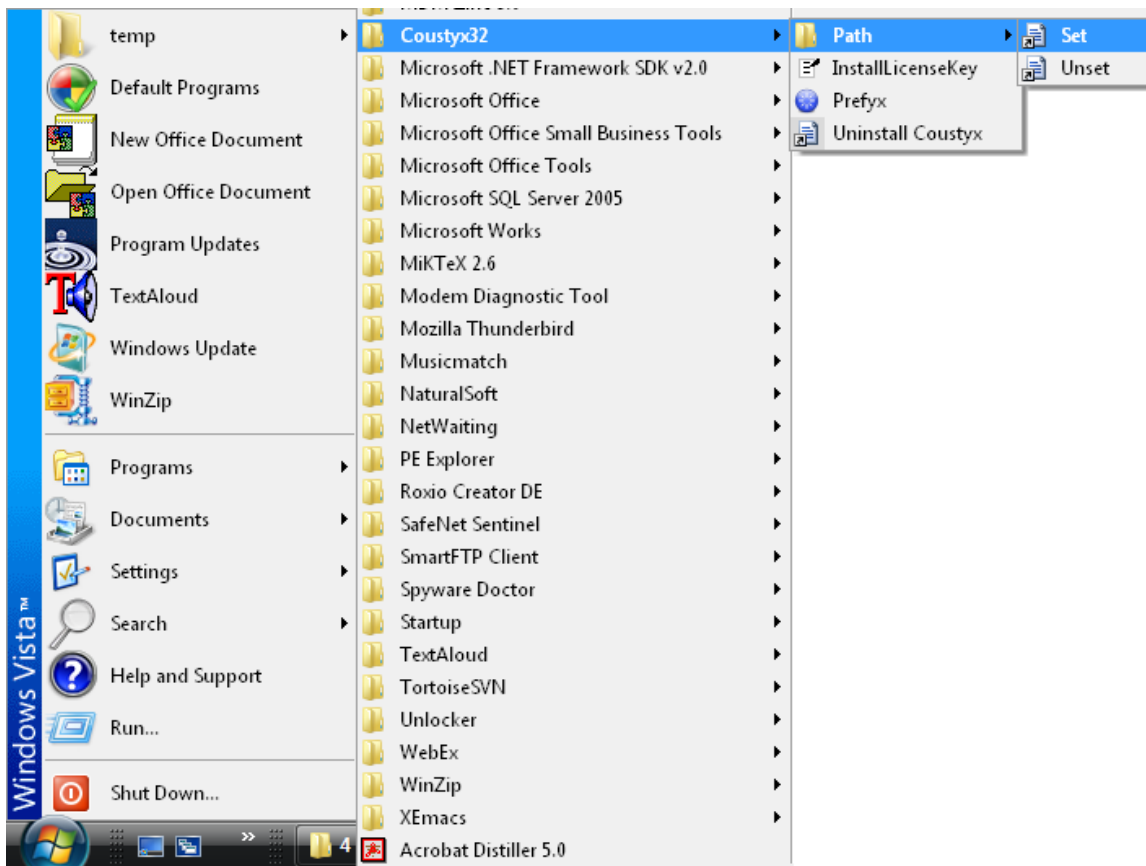


Figure 2.10 Set path variable.

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

$$P(t) = \mathbf{Re} \{ p e^{-j\omega t} \}$$

where  $\mathbf{Re}$  stands for “real part of”,  $p$  is the complex amplitude of the sound pressure, and  $\omega$  is the angular frequency.

### 3.2. Units

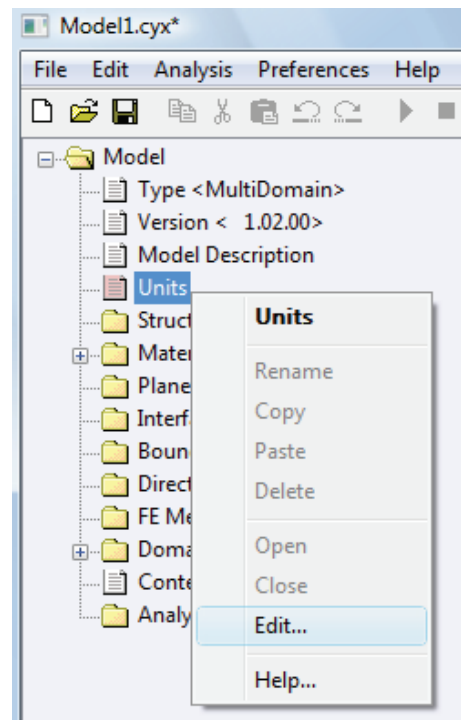
#### 3.2.1. Model Units

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.

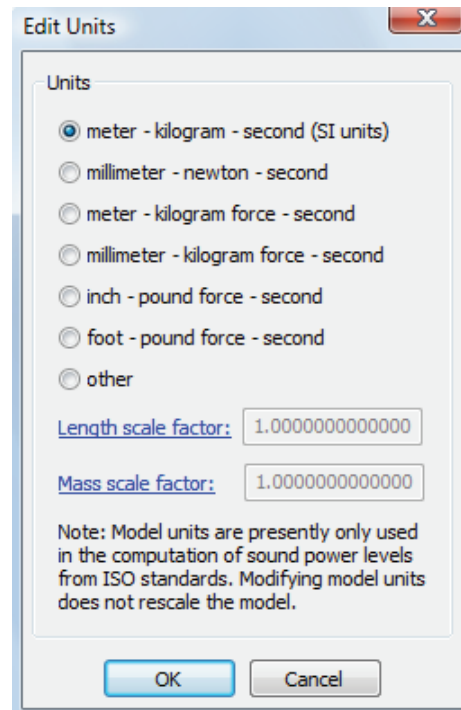
For convenience, six different standard unit systems are predefined: **meter - kilogram - second** ( $m - kg - s$ ), **millimeter - newton - second** ( $mm - N - s$ ), **meter - kilogram force - second** ( $m - kgf - s$ ), **millimeter - kilogram force - second** ( $mm - kgf - s$ ), **inch - pound force - 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  $1\text{ mm} = 0.001\text{ m}$ . Table 3.1 shows length scale factors for some commonly used length units.

**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?





**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

**Table 3.3** Variables and Units

Model units	Inputs			Outputs		
	Geometry units	Speed of sound	Ambient Density	Pressure	Particle Velocity	Sound Power
m – kg – s	m	m/s	kg/m <sup>3</sup>	kg/m/s <sup>2</sup> (Pa)	m/s	kg – m <sup>2</sup> /s <sup>3</sup> (W)
mm – N – s	mm	mm/s	N – s <sup>2</sup> /mm <sup>4</sup>	N/mm <sup>2</sup> (MPa)	mm/s	N – mm/s (mW)
m – kgf – s	m	m/s	kgf – s <sup>2</sup> /m <sup>4</sup>	kgf/m <sup>2</sup>	m/s	kgf – m/s
mm – kgf – s	mm	mm/s	kgf – s <sup>2</sup> /mm <sup>4</sup>	kgf/mm <sup>2</sup>	mm/s	kgf – mm/s
inch – lbf – s	inch	inch/s	lbf – s <sup>2</sup> /inch <sup>4</sup>	lbf/inch <sup>2</sup>	inch/s	lbf – inch/s
ft – lbf – s	ft	ft/s	lbf – s <sup>2</sup> /ft <sup>4</sup>	lbf/ft <sup>2</sup>	ft/s	lbf – ft/s

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** (m – kg – s). As shown in Table 3.3, the speed of sound should be in m/s and the ambient density should in kg/m<sup>3</sup>. The derived acoustic quantities, such as acoustic pressure should be in Pascal (1 Pa = 1 kg/m/s<sup>2</sup>), the particle velocity in m/s, the sound power in Watts (1 W = 1 kg.m<sup>2</sup>/s<sup>3</sup>), and the intensity in Watt/m<sup>2</sup>.

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

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

Model units	Material Properties	
	Speed of Sound	Ambient Density
m – kg – s	343 m/s	1.21 kg/m <sup>3</sup>
mm – N – s	343000 m/s	$1.21 \times 10^{-12}$ N – s <sup>2</sup> /mm <sup>4</sup>
m – kgf – s	343 m/s	0.1233857 kgf – s <sup>2</sup> /m <sup>4</sup>
mm – kgf – s	343000 mm/s	$0.1233857 \times 10^{-12}$ kgf – s <sup>2</sup> /mm <sup>4</sup>
inch – lbf – s	13503.94 inch/s	$1.132228 \times 10^{-7}$ lbf – s <sup>2</sup> /inch <sup>4</sup>
ft – lbf – s	1125.33 ft/s	0.0023478 lbf – s <sup>2</sup> /ft <sup>4</sup>

### 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).

**New Boundary Condition**

Name: New BC 2

Type: Uniform Normal Velocity (Continuous) [Help](#)

Normal Velocity

Frequency Dependence Type: Table

	AngFrq.(Rd/s)	Freq. (Hz)	Real(Value)	Imag(Value)
New:				

Import from file...

OK Cancel

**Figure 3.3** Boundary Condition Table.

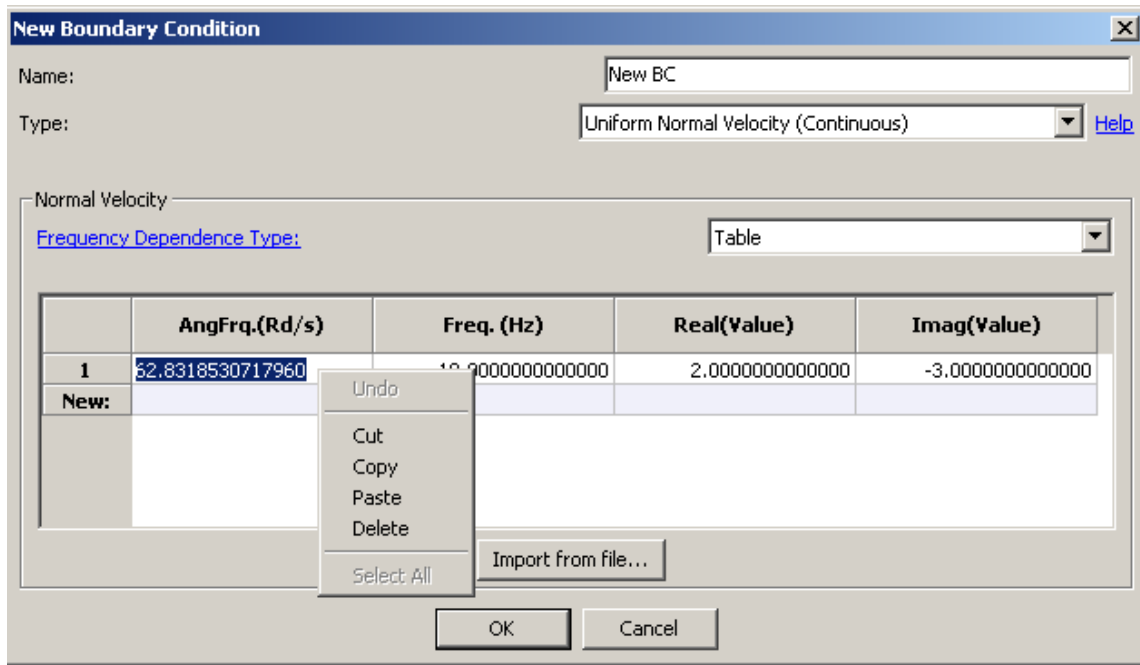


Figure 3.4 Edit Boundary Condition Table.

**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 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.

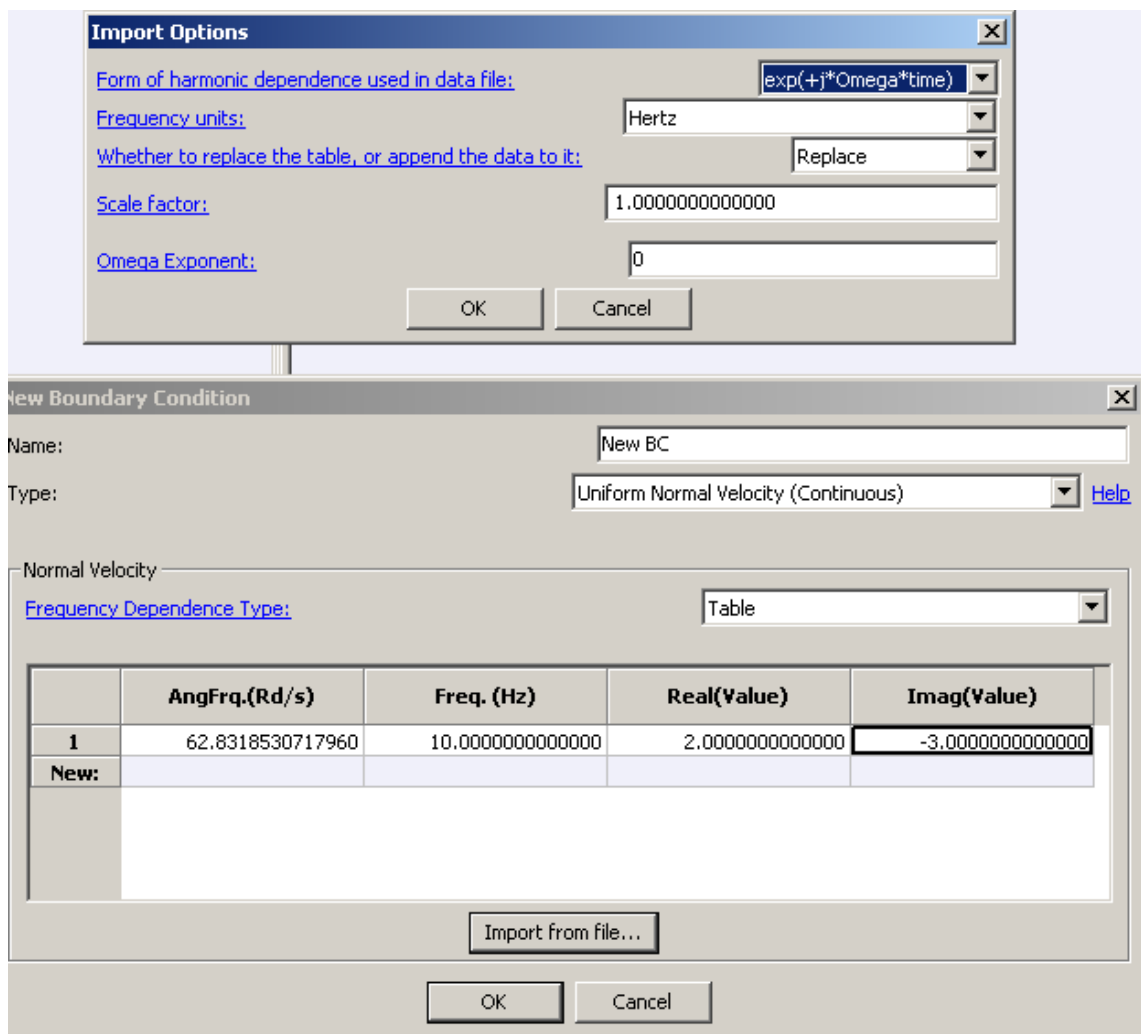
**3.3.2.2. Import Options** The import options window, Figure 3.6, appears after the selection of the *ASCII* file to be used to import the table. This window controls the interpretation of the data from the file.

**Form of harmonic dependence used in data file**  $\exp(+j * \text{Omega} * \text{time})$  (or  $e^{+j\omega t}$ ) and  $\exp(-j * \text{Omega} * \text{time})$  (or  $e^{-j\omega t}$ ) are the two available options, where **Omega** ( $\omega$ ) 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.



**Figure 3.5** Table Boundary Condition example.

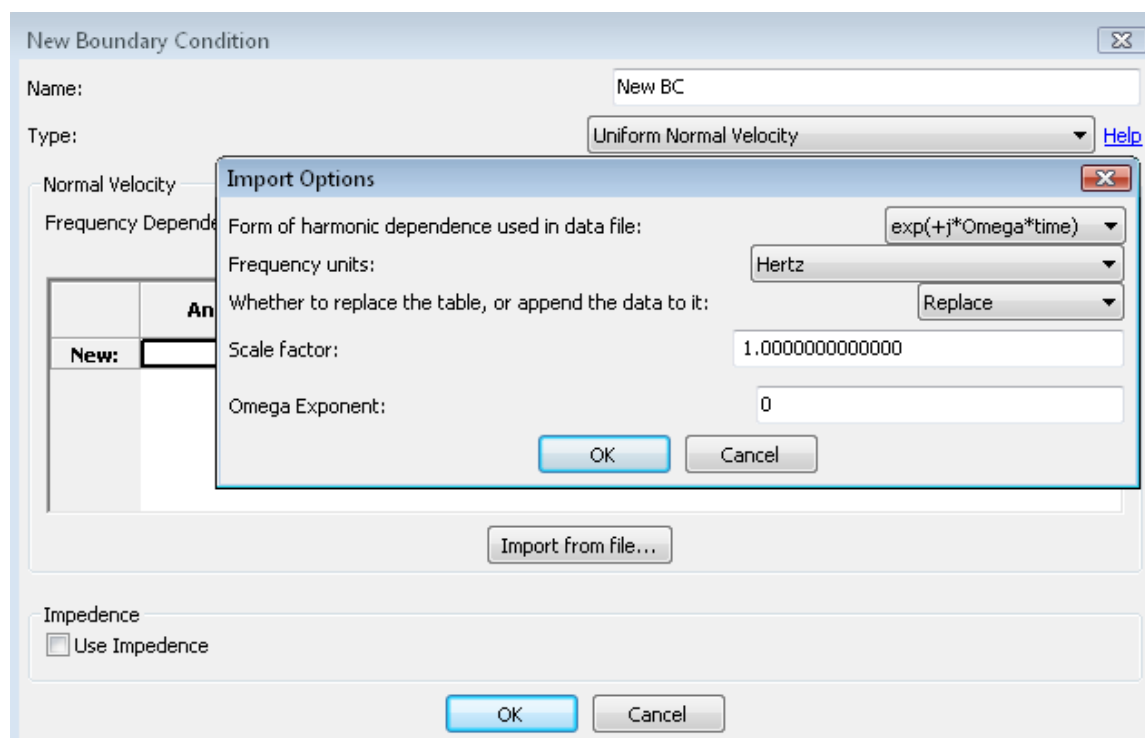


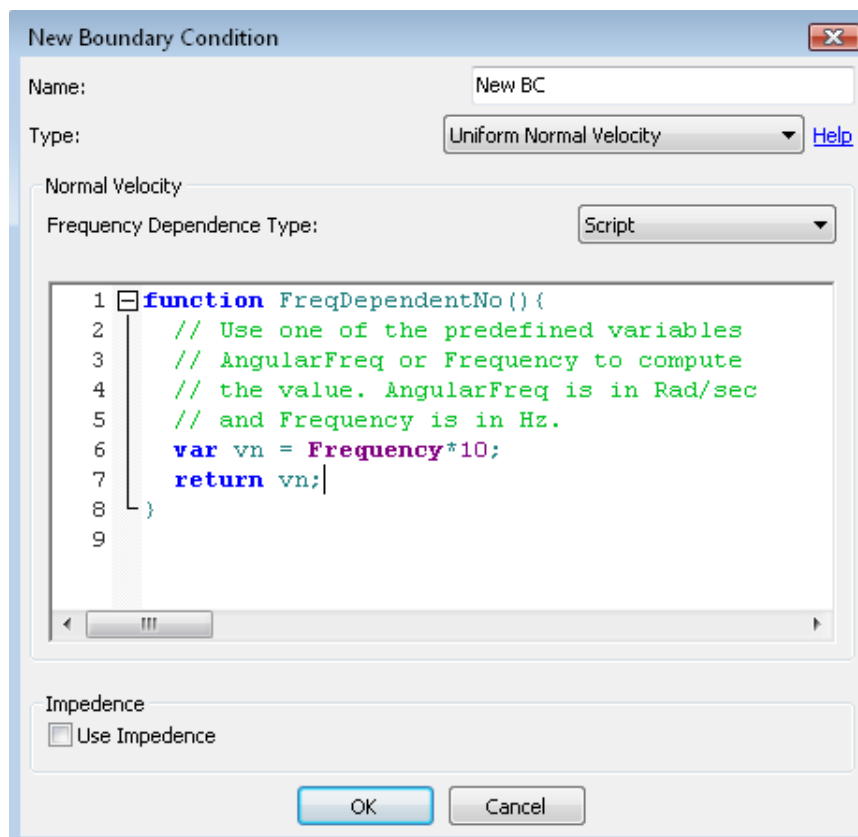
Figure 3.6 Table import options window.

**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  $\omega$ ) 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  $s = s_0 e^{j\omega t}$  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  $a = a_0 e^{j\omega t}$  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
<b>File</b>	
<i>Open Model File</i>	Open an existing <i>Coustyx</i> model (*.cyx files only)
<i>New Model</i>	Create a new <i>MultiDomain</i> or <i>Indirect Coustyx</i> model
<i>Save Model File</i>	Save all the changes made to a model

Table 4.1: (continued)

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

Table 4.1: (continued)

Menu Items	Description
<i>Search Previous Selected</i>	Do backward search for the word selected in the script. Search criteria include Case sensitivity search ( <i>Case</i> ), Whole word search ( <i>Word</i> ), and Regular expression search ( <i>RegExp</i> )
<i>Search Next</i>	Do forward search for the word typed in the search text box. Search criteria include Case sensitivity search ( <i>Case</i> ), Whole word search ( <i>Word</i> ), and Regular expression search ( <i>RegExp</i> )
<i>Search Next Selected</i>	Do forward search for the word selected in the script. Search criteria include Case sensitivity search ( <i>Case</i> ), Whole word search ( <i>Word</i> ), and Regular expression search ( <i>RegExp</i> )
<i>Search</i>	Search
<i>Page</i>	Browse through multi-page window to view <i>First</i> , <i>Previous</i> , <i>Next</i> , and <i>Last</i> pages
<b>Analysis</b>	
<i>Run</i>	Run the selected Analysis Sequence
<i>Abort</i>	Abort the current analysis run
<b>Preferences</b>	
<i>Common</i>	(Figure 4.2)
<i>Show splash screen on start up</i>	Enable or disable splash window at the startup of <i>Coustyx</i>
<i>Show log messages</i>	Show log messages in the log window
<i>Verbosity Level</i>	This determines how much data that is output while running <i>Coustyx</i> . 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)

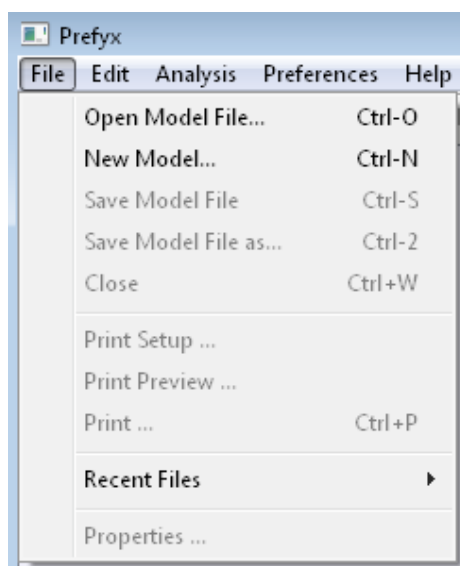
Menu Items	Description
<i>Save file in binary format</i>	Enable saving <i>Coustyx</i> model file (*.cyx) in compressed binary file format. When unchecked the file is saved in <i>ASCII</i> (XML) format. It is recommended to save a large <i>Coustyx</i> model file in the compressed binary file format
<i>3D Viewer</i>	(Figure 4.3)
<i>Feature angle (deg)</i>	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 <a href="#">Select Connected Elements Through Feature Angle</a> is used to select surface features in a mesh. The default value for the feature angle is set at 15°. But it can be any value between 0° and 180°. Note that the definition is applicable only for 2D elements
<i>Incremental rotation angle (deg)</i>	Set the increments of rotation angle in the GUI
<i>Time for rotation (msec)</i>	Set the speed of the rotation in the GUI
<i>Zoom Increment</i>	Specify the increments at which the zoom takes place in the GUI
<i>Show element edges by default</i>	Show element edges when a structure mesh is opened in the GUI
<i>Element resolution level by default</i>	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
<i>Relative size of nodes</i>	Set the relative size of the nodes displayed in the GUI
<i>Show shadows</i>	Show shadows while displaying a structure mesh in the GUI
<i>Use fast draw mode</i>	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

Table 4.1: (continued)

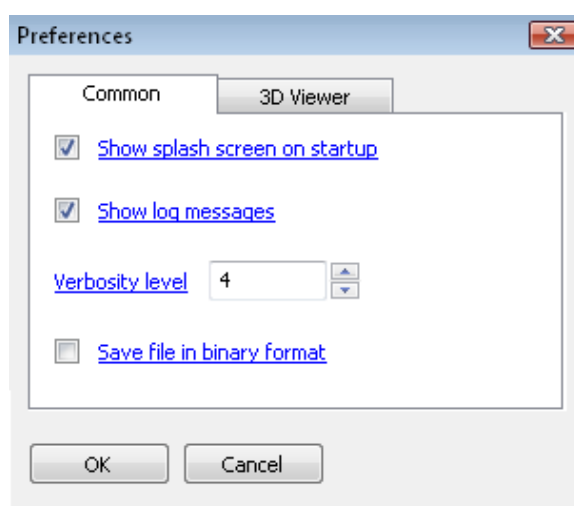
Menu Items	Description
<i>Define Cut Plane 1</i>	Define a cutting plane to view a cross section of the model. Only the section (x,y,z) of the model that satisfies: $a_1x + b_1y + c_1z \geq d_1$ is displayed. By default the Cut mode is turned off. To view the model in Cut mode, press the Cut tool in the GUI control panel
<i>Define Cut Plane 2</i>	Define a cutting plane to view a cross section of the model. Only the section (x,y,z) of the model that satisfies: $a_2x + b_2y + c_2z \geq d_2$ is displayed. By default the Cut mode is turned off. To view the model in Cut mode, press the Cut tool in the GUI control panel
<i>Define Cut Plane 3</i>	Define a cutting plane to view a cross section of the model. Only the section (x,y,z) of the model that satisfies: $a_3x + b_3y + c_3z \geq d_3$ is displayed. By default the Cut mode is turned off. To view the model in Cut mode, press the Cut tool in the GUI control panel
<i>Background Color</i>	Set the background color of the GUI
<b>Help</b>	
<i>Help Content</i>	Show the user manual for help
<i>About</i>	Show the license information

## 4.2. Operations on Mesh Viewer Window

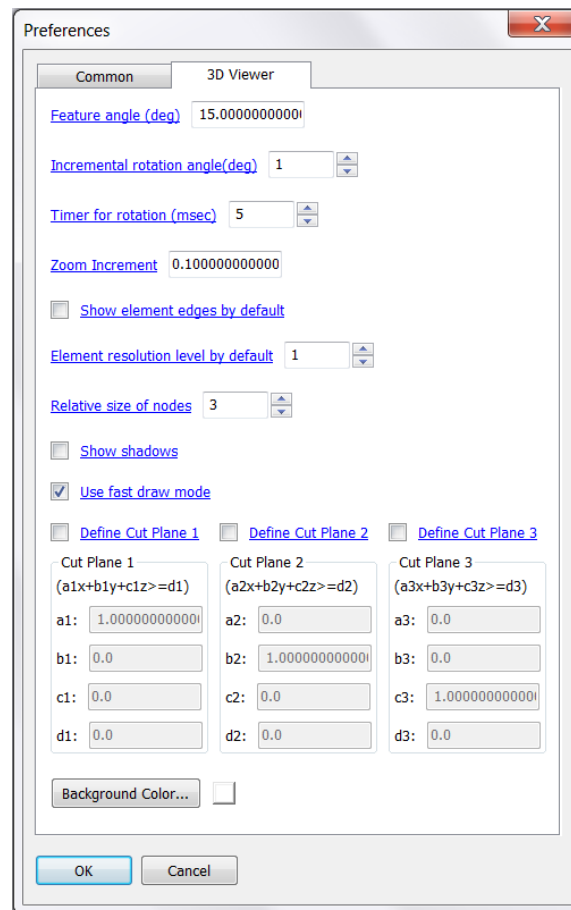
We can view the Finite Element (FE) structure mesh or the Boundary Element (BE) mesh by opening them in a *Mesh Viewer* window. This section describes the user controls and the functions available to manipulate a mesh when it is open in the *Mesh Viewer* window. Note that the *Mesh Viewer* window appears 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



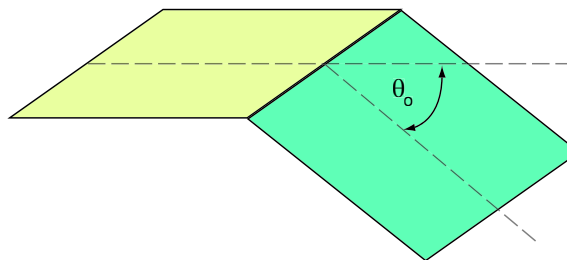
**Figure 4.1** Main menu items



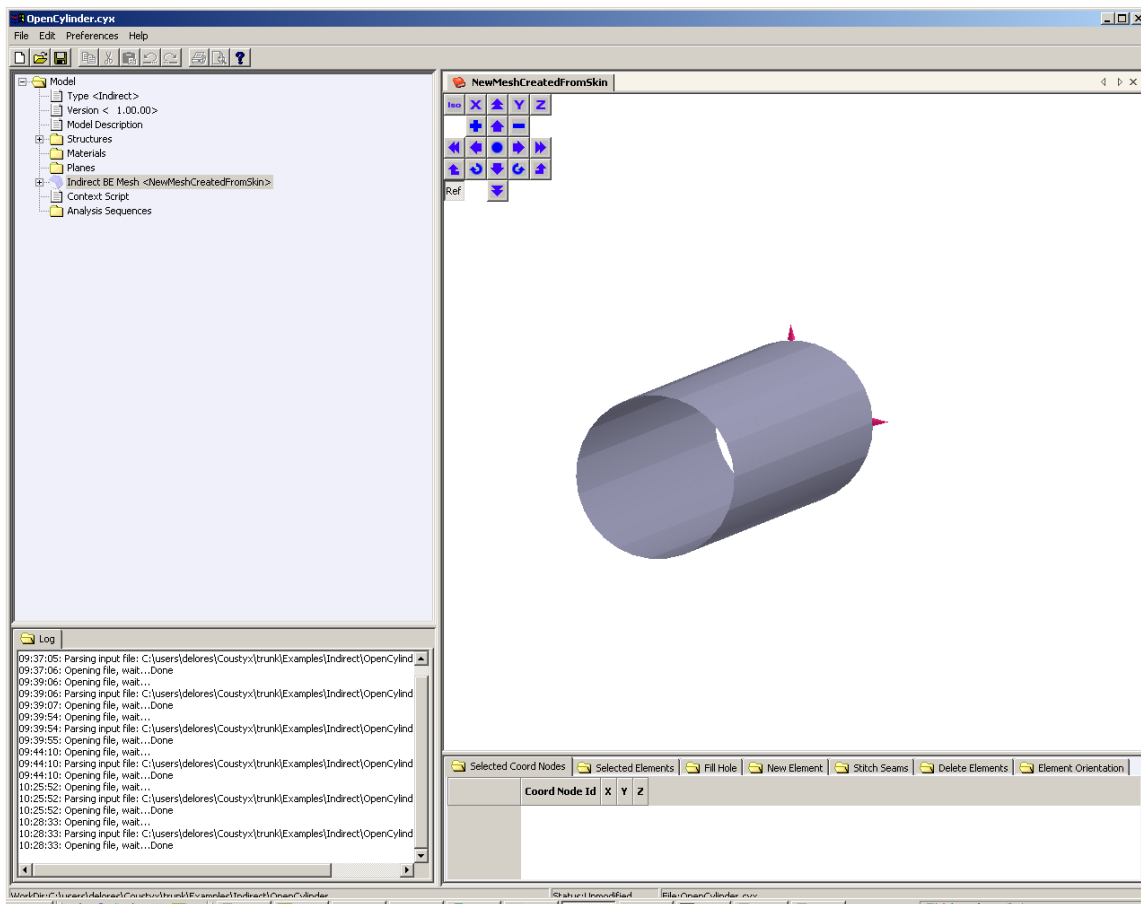
**Figure 4.2** Preferences dialog box - Common parameters



**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.



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.

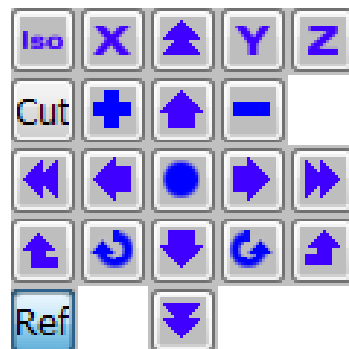


Figure 4.6 GUI Control Panel Tools

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°
	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

Table 4.2: (continued)

Tool	Usage	Tool	Usage
	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 <i>shift-key</i> while the cursor is in the GUI
	turn on or off the display in Cut mode. Define cutting planes in the Preferences Dialog Box		

## 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

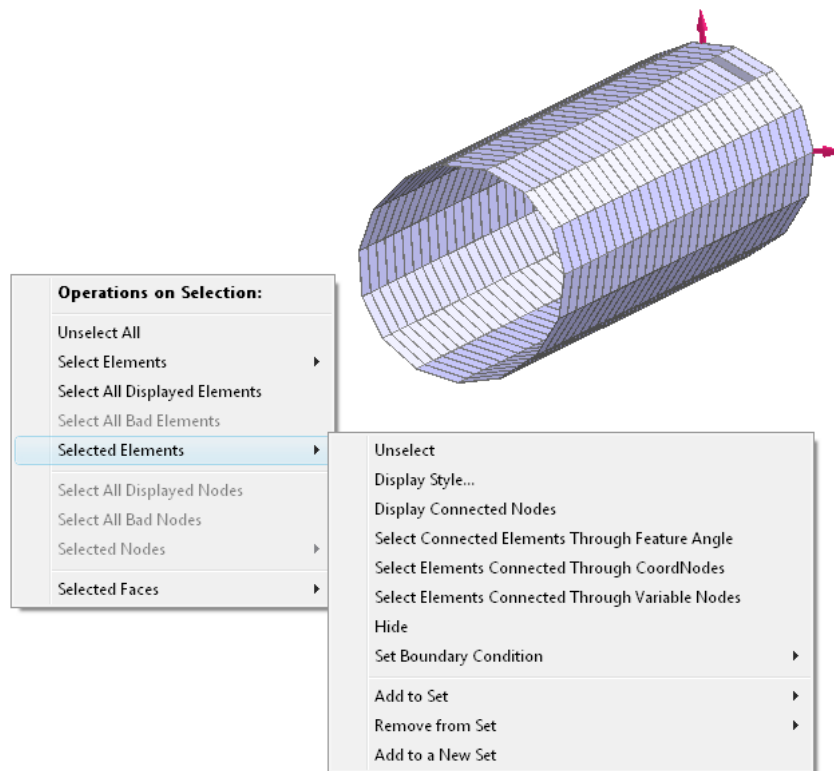
Below is a brief description of the operations that may be performed on displayed elements, nodes or faces in the *Mesh Viewer* window. Figure 4.7 shows the context menu that pops up when you right-click anywhere in the *Mesh Viewer* window while holding down the *shift-key*.

### 4.2.3.1. Operations on Displayed Elements

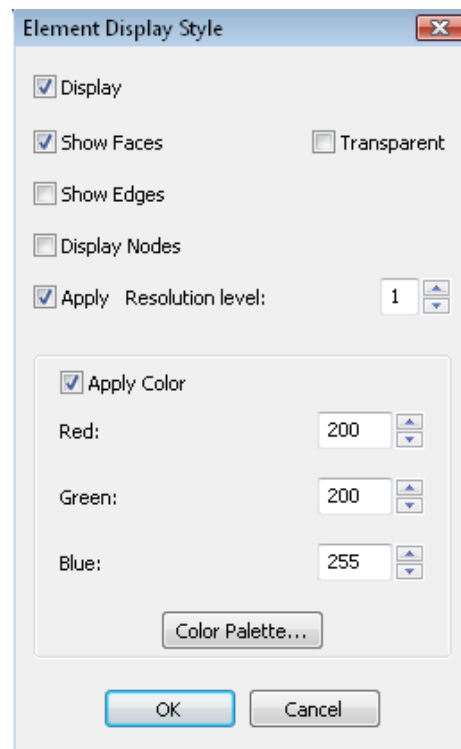
**Unselect All** This un-selects all the elements in the GUI.

**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 applied with a particular boundary condition from the displayed list. Note that this option is available only for boundary element meshes. For a model with the same boundary



**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.

element mesh in multiple domains, or with multiple instances of the mesh in the same domain, choose the domain name and the instance number through **Choose Domain** and **Choose Instance** submenus.

**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.

**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  $\mu$  (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. For a model with the same boundary element mesh specified in multiple domains or multiple instances of the mesh in the same domain, choose the domain name and the instance number through **Choose Domain** and **Choose Instance** submenus before applying the boundary condition. 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 4.3.9.5 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 4.3.9.5 for more information on Sets.

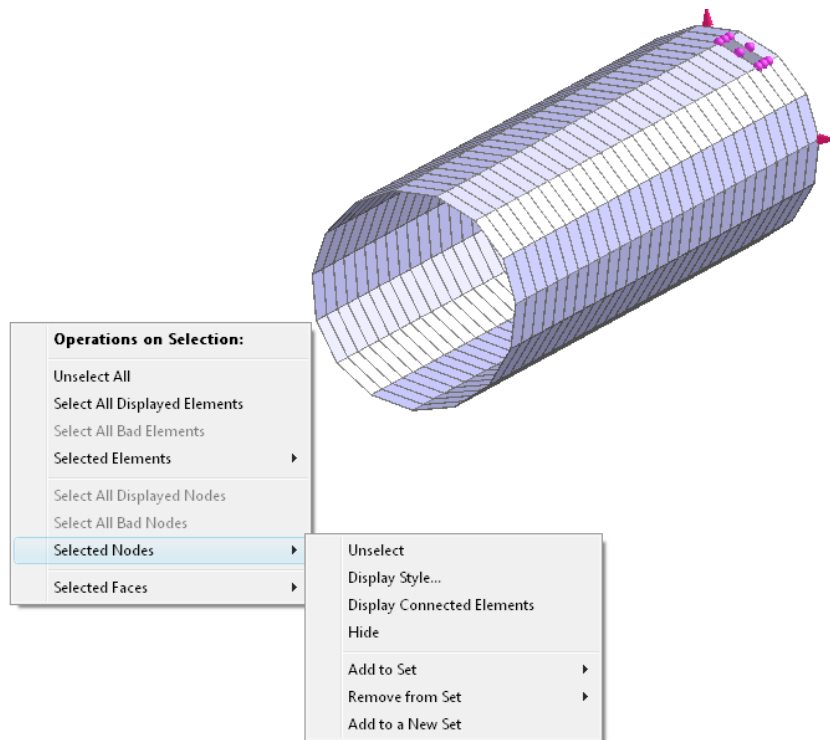
**Add to a New Set** This option allows user to add selected elements to a new Set. A new set with a default name **New Set** or **New Set i** (i is a number) is created. Refer to Section 4.3.9.5 on how to rename the Set.

**4.2.3.2. Operations on Displayed Nodes** To use operations on displayed nodes, first make nodes visible in the GUI.

**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.



**Figure 4.9** GUI operations available for selected nodes.

**Unselect** This un-selects the selected nodes.

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

**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.

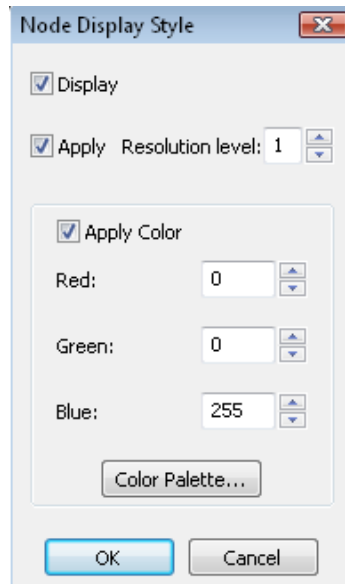
**Add to Set** This option allows user to add selected nodes to a pre-defined Set. Refer to Section 4.3.9.5 on how to define a Set.

**Remove from Set** This option is used to remove selected nodes from their current Sets. Refer to Section 4.3.9.5 for more information on Sets.

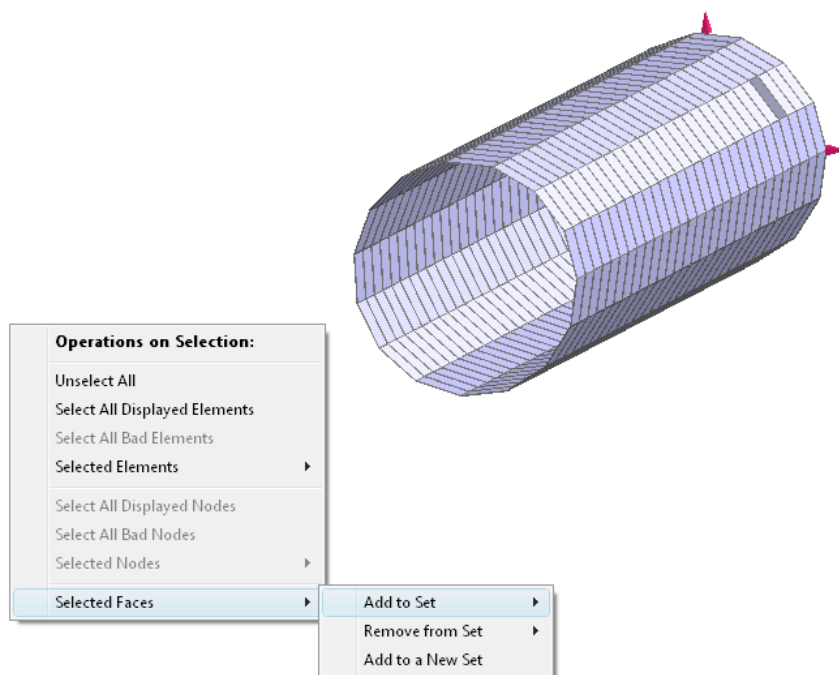
**Add to a New Set** This option allows user to add selected nodes to a new Set. A new set with a default name **New Set** or **New Set i** (i is a number) is created. Refer to Section 4.3.9.5 on how to rename the Set.

**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.

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



**Figure 4.10** Nodes Display Style Window.



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

**Add to Set** This option allows user to add selected faces to a pre-defined Set. Refer to Section 4.3.9.5 on how to define a Set.

**Remove from Set** This option is used to remove selected faces from their current Sets. Refer to Section 4.3.9.5 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 4.3.9.5 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.

**Display B.C for Mesh in:** This option is displayed only for BE meshes in *MultiDomain* models. In a *MultiDomain* model, each BE mesh could be part of multiple domains or could be instantiated more than once in the same domain. Each such instance of the BE mesh can be applied with different boundary condition (B.C). Specify the **Domain** and the **Instance** of the BE mesh to display appropriate B.C of selected elements. Click on the buttons to change the domain name or the instance number of the mesh.

**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 ( $\sigma$ ) and double layer ( $\mu$ ) 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 ( $\sigma$ ) 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 ( $\mu$ ) 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  $\mu$  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 5.4.1.

#### 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 5.4.1. (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 5.6.2.1 for more details.

#### 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 5.6.2.1.

#### 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

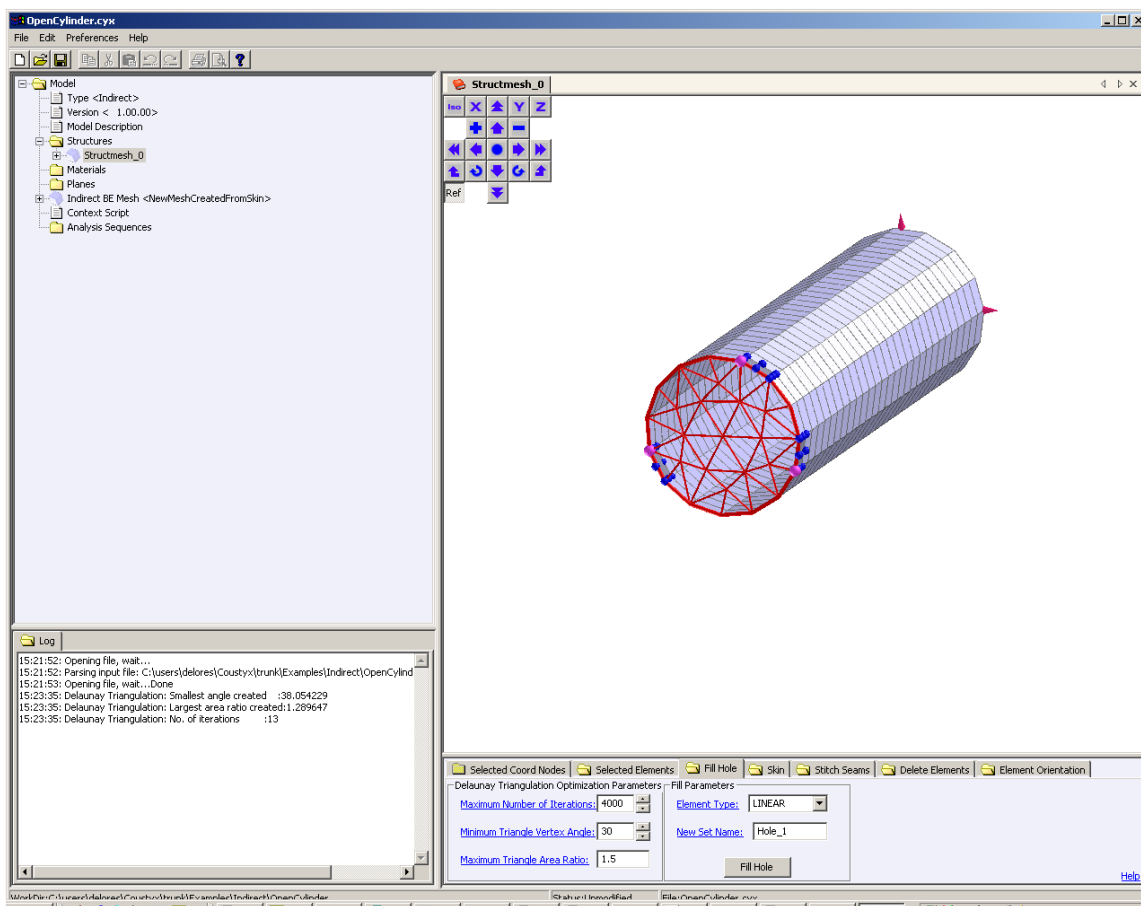


Figure 4.12 Fill Hole

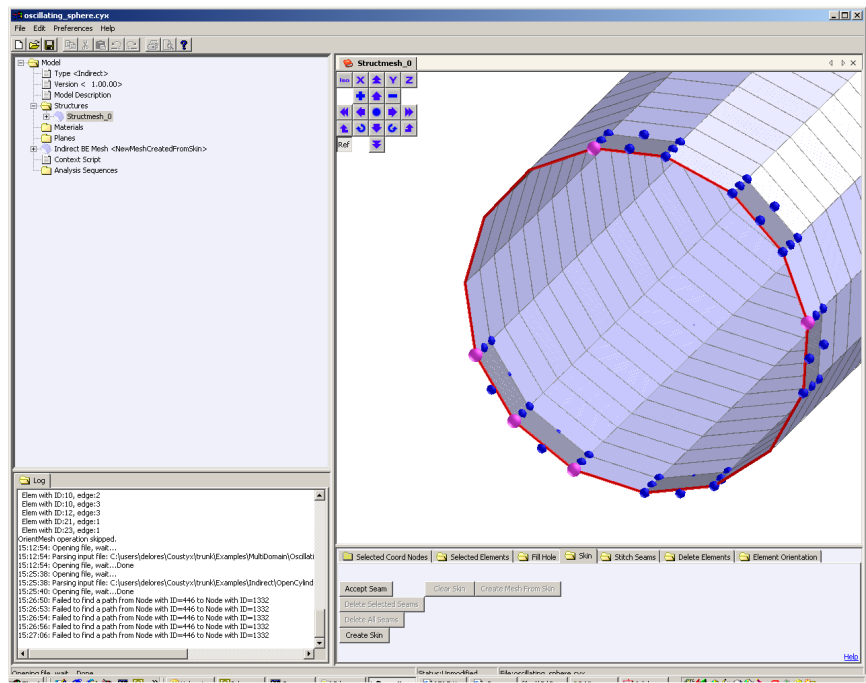


Figure 4.13 Create Skin

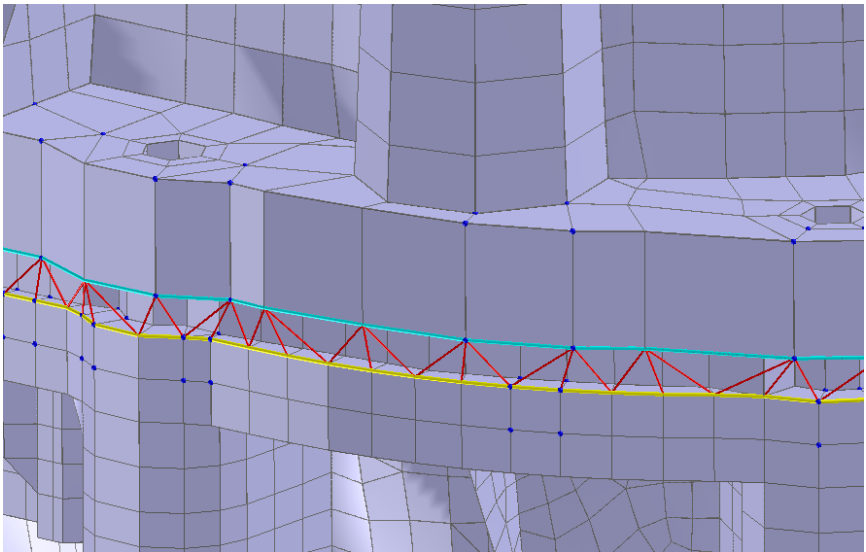


Figure 4.14 Stitch Seams

coordinate nodes and unshared variable nodes of the deleted element. Refer to Section 5.6.2.1 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 5.6.2.1 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 5.6.2.1 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 5.6.2.1 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* ( $p_n$ ).

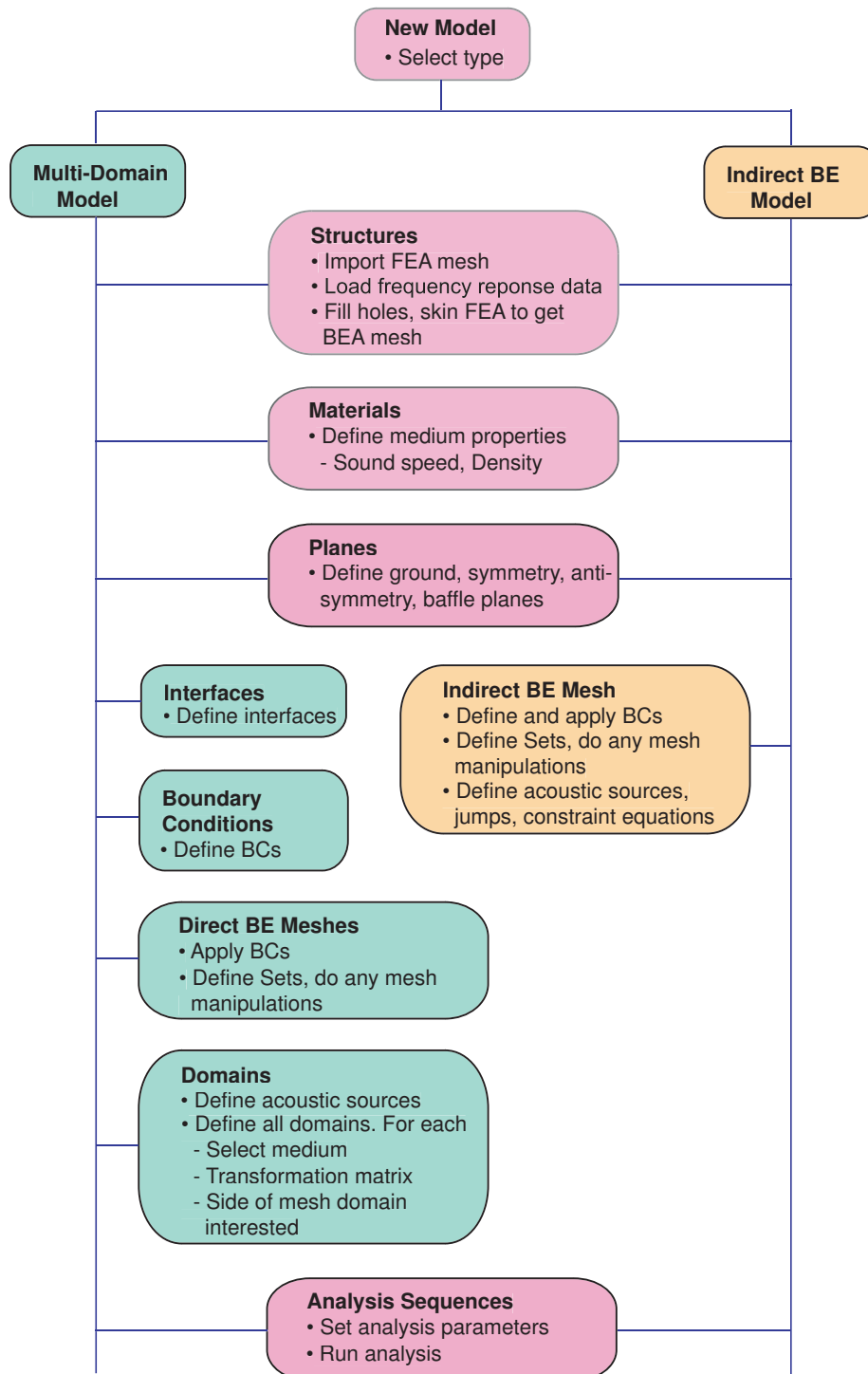
The *normal derivative of pressure* ( $p_n$ ) and *normal velocity* ( $v_n$ ) at a point are related as follows:

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

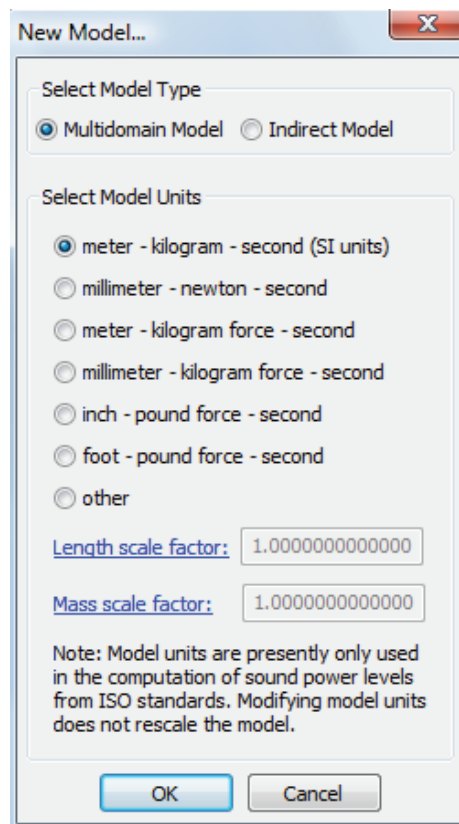
where  $\rho_o$  is the ambient density,  $\omega$  is the frequency.

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.

**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 *pres-*



**Figure 4.15** Coustyx model tree structure.



**Figure 4.16** New model dialog box.

sure discontinuities and velocity jumps or velocity discontinuities across the boundary. These are also known as *double layer potentials* ( $\mu$ ) and *single layer potentials* ( $\sigma$ ) respectively.

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

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

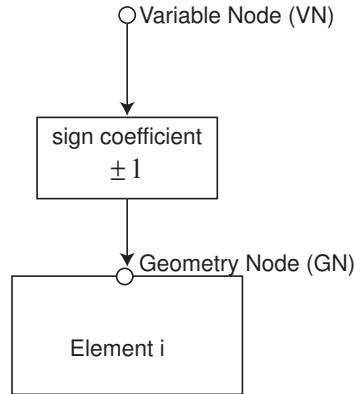
The double layer potential ( $\mu$ ) is the difference between the pressures on the positive ( $p^+$ ) and negative ( $p^-$ ) sides of an element, that is,

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

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 ( $\mu$ ) at a geometry node in an Element  $i$  to its variable node through a positive or a negative sign (Figure 4.17). Each  $\mu$  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.

$$\mu_i(\text{Within Element}) = \text{sign coefficient} \bullet \mu(\text{Variable Node}) \quad (4.1)$$



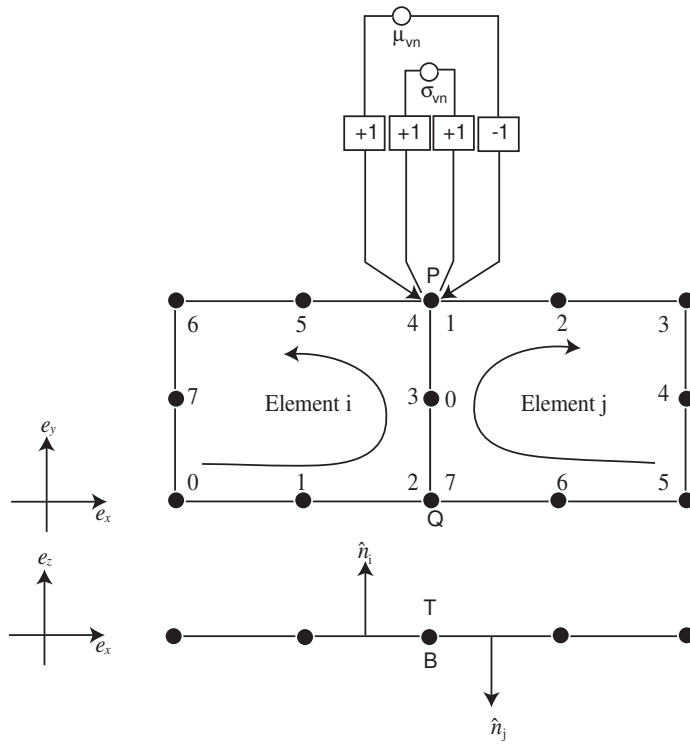
**Figure 4.17** Variable Nodes in Indirect BEM

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  $\mu_{vn}$  and  $\sigma_{vn}$ .

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

$$\sigma_i(4) = \frac{\partial p_T}{\partial n_i} - \frac{\partial p_B}{\partial n_i} = \frac{\partial p_T}{\partial z} - \frac{\partial p_B}{\partial z} = \sigma_{vn}$$

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 4.18** Variable Nodes in Indirect BEM

$$\mu_j(1) = p_B - p_T = -\mu_{\text{vn}}$$

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

The sign coefficient for  $\mu_j(1)$  is  $-1$ , negative of the sign coefficient for  $\mu_i(4)$ .

$$\begin{aligned}\mu_i(4) &= \mu_{\text{vn}} \\ \mu_j(1) &= -\mu_{\text{vn}}\end{aligned}$$

However, the sign coefficient for  $\sigma_j(1)$  is equal to the sign coefficient for  $\sigma_i(4)$ . 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  $i$ .

$$\begin{aligned}\sigma_i(4) &= \sigma_{vn} \\ \sigma_j(1) &= \sigma_{vn}\end{aligned}$$



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 ( $\sigma$ ) and double layer ( $\mu$ ) 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 ( $\sigma$ ) and double layer ( $\mu$ ) potentials on the surface.
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 5.4.1).

### 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).

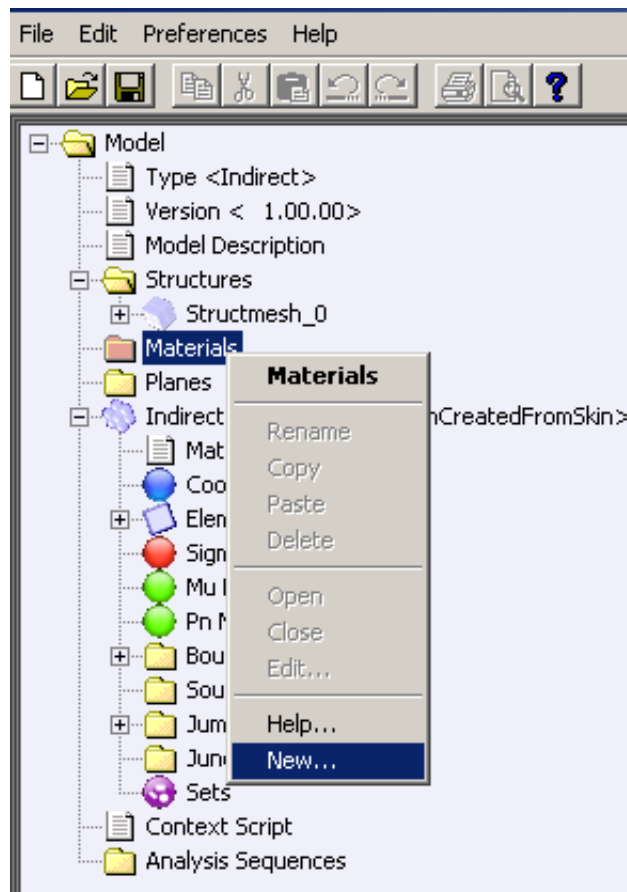


Figure 4.19 Create New Material

**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*,

$$P = \text{Re}(pe^{-j\omega t})$$

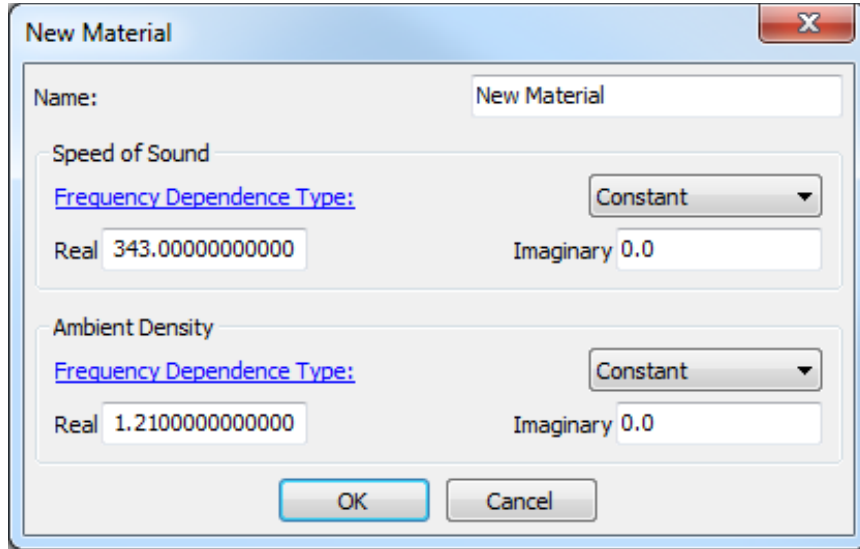


Figure 4.20 Edit Material Properties

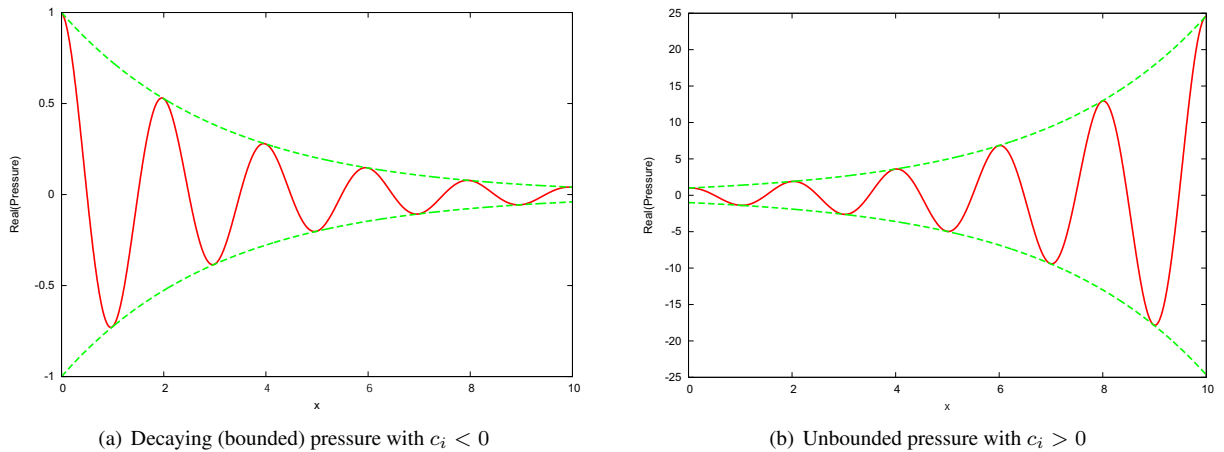
where **Re** stands for “real part of”,  $P$  is the time-harmonic pressure wave,  $p$  is the complex amplitude of the sound pressure,  $\omega$  is the frequency of fluctuation, and  $j = \sqrt{-1}$ .

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 to Chapter 3.



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

**4.3.3.2. Ambient Density** The value for the ambient density ( $\rho_o$ ) of the fluid medium is defined here. Density by definition is always real, but to accomodate specification of porous medium using equivalent

fluid concept Coustyx allows users specify equivalent density as a frequency dependent complex value. The variation with frequency can be introduced by defining the ambient density to be a frequency dependent type: *Table* or *Script*.

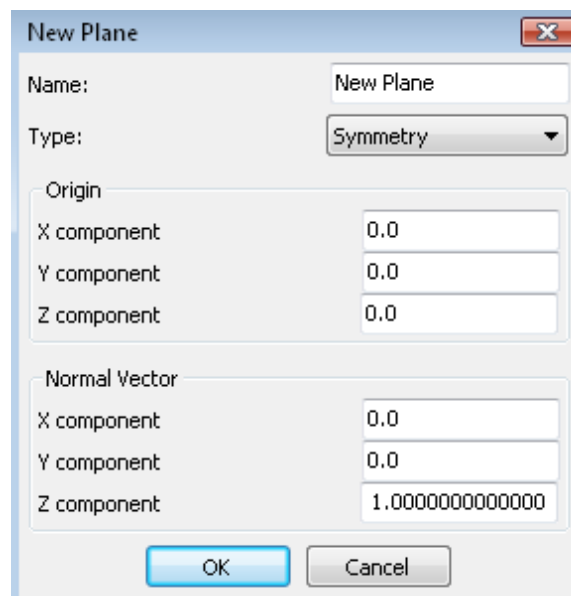
Note that the units used for ambient density along with the units for speed of sound determine *Coustyx* model units. For more information on the unit conventions followed in *Coustyx* refer to Chapter 3.

#### 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 are defined, they must be orthogonal to each other. Only one baffle plane can be defined in a model.

- **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.

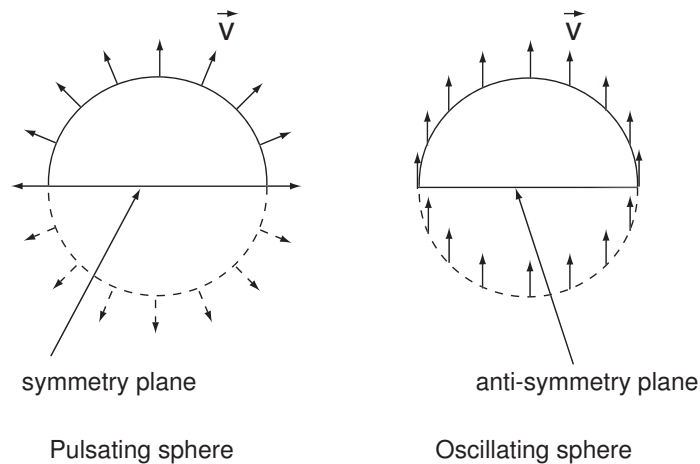


**Figure 4.22** Creating a new plane.

**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



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

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 currently only implemented for *MultiDomain* models with HIE Collocation and Burton Millier Collocation formulations.

**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. For more information see Section 6.

### 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 6.5 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 6.6.12).

- **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 5.4.1 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 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 a table listing the properties of all the elements (such as ID, B.C., connectivity, etc.,) in the mesh. The table rows can be copied to a clip board by right-clicking on the rows and selecting *Copy*.

For a BE mesh in *MultiDomain* model, there is an additional option at the top of the table to specify the **Domain** and the **Instance** of the mesh so as to display appropriate boundary conditions (B.C.). Click on the buttons to change the domain name or the instance number of the mesh.

**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 4.3.9.5 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  $\rho_o$  is the ambient density,  $\omega$  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 4.3.9.5 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 5.4.1 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 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 4.3.1.2 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.



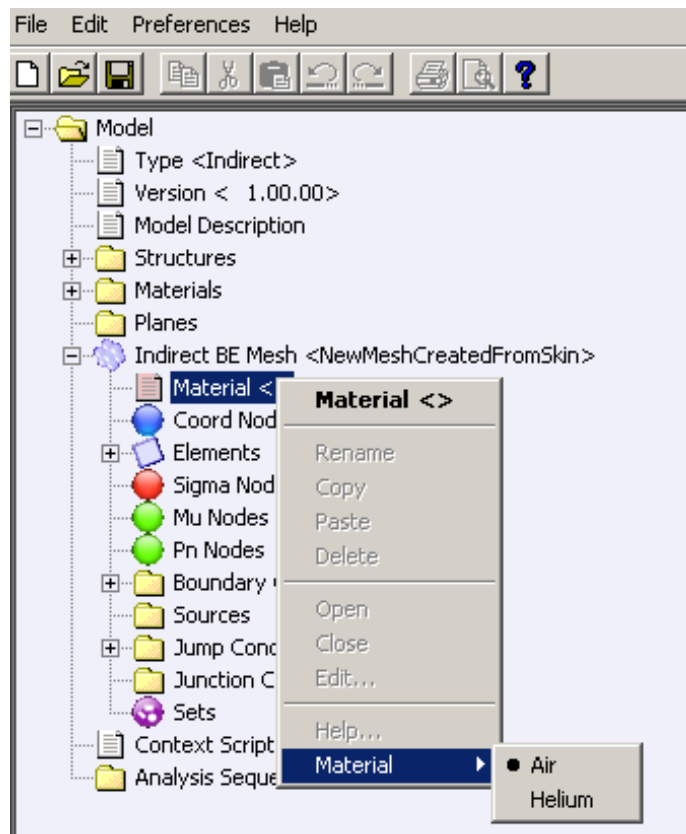


Figure 4.24 Selecting Material

**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.

**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 ( $\sigma$ ) 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 ( $\sigma$ ) and double layer potential ( $\mu$ ) 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  $4\pi$  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 ( $\mu$ ) 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 ( $\sigma$ ) and double layer potential ( $\mu$ ) 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  $4\pi$  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 6.5.8.4 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 6.6.12).

- **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.

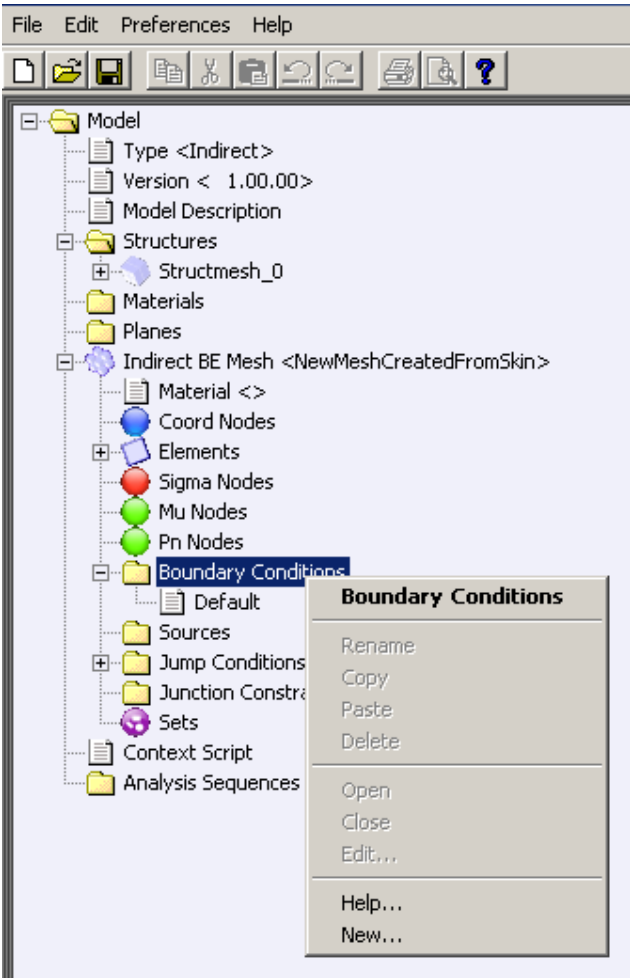


Figure 4.25 Create Boundary Conditions

#### 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 4.3.9.5 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.

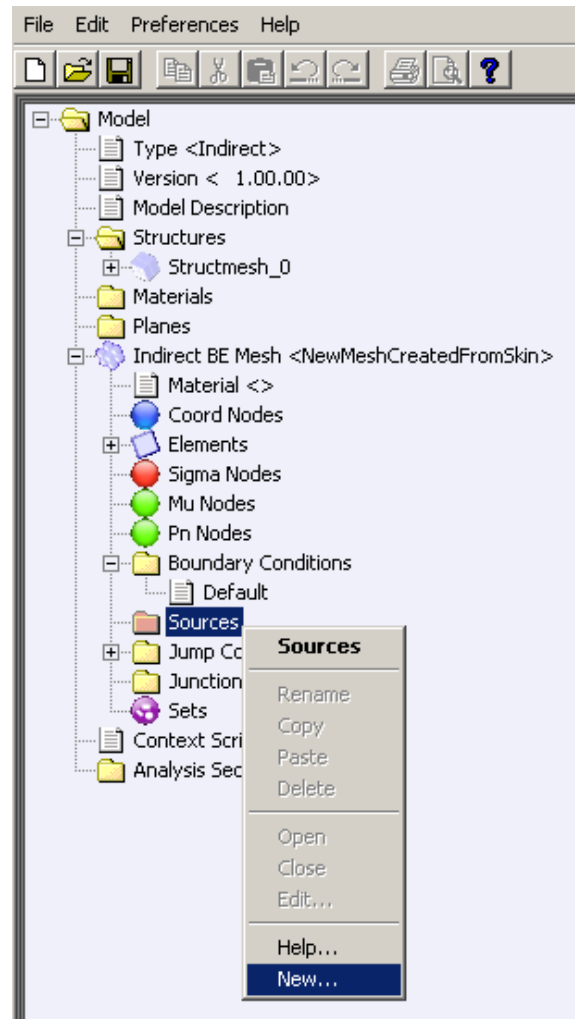


Figure 4.26 Create New Source

#### 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 (*Mus*) 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 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  $\mu_1$ ,  $\mu_2$ ,  $\mu_3$  represent the value of the double layer density at the same spatial location. By definition,  $\mu_1$ ,  $\mu_2$ ,  $\mu_3$  represent the following pressure differences.

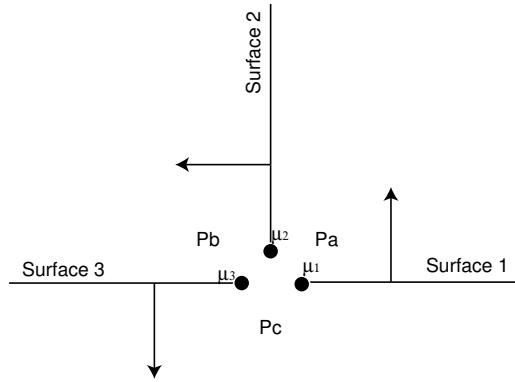
$$\mu_1 = p_a - p_c$$

$$\mu_2 = p_b - p_a$$

$$\mu_3 = p_c - p_b$$

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

$$\mu_1 + \mu_2 + \mu_3 = 0 \quad (4.2)$$



**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 4.3.9.5 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. A default domain is automatically created when a BE mesh is generated from the skin of a structure FE mesh.

To add a new domain right-click on **Domains** and select **New...**. A new dialog box pops up to let you choose the following basic properties of a domain.

**Name** Specify a unique name to identify the new domain.

**Type** Specify the type of the domain. Currently only domains of the type *Direct BE* are allowed.

**Material Name** Select the fluid medium present in the domain.

**Unbounded Domain** Set the boundedness of the domain by appropriately checking this box. If the domain of interest is *exterior* then check the box to set the boundedness to **Unbounded**. If the domain of interest is *interior* then uncheck the box to set the boundedness to **Bounded**. Note that in a *MultiDomain* model you can specify multiple *Bounded* domains but only one *Unbounded* domain except for the case of a model with *baffle* plane where two *Unbounded* domains are allowed.

Other components of a domain that are specified later on are listed below.

#### 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 4.3.9.5 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 except for the case of a model with *baffle* plane where two *Unbounded* domains are allowed..

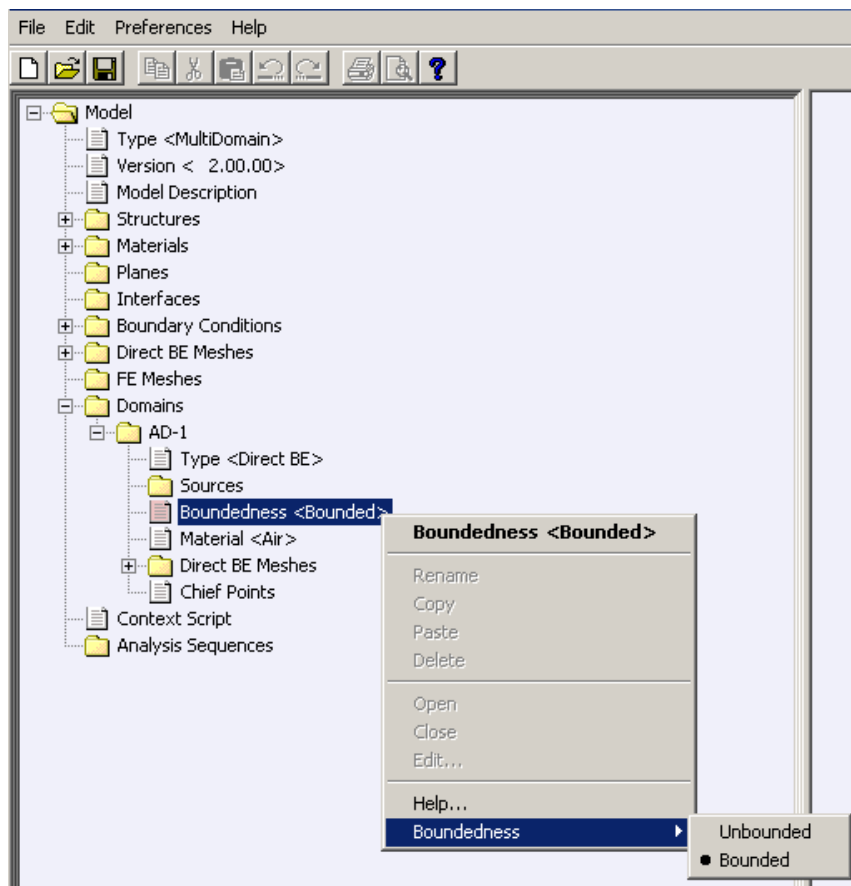


Figure 4.28 Choosing Boundedness

#### 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 4.3.1.2 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. To add a BE mesh to a domain, right-click on **Direct BE Meshes** → **Add BE Mesh** and choose the name of the mesh. One can add multiple instances of the same BE mesh to the domain. The instance number of each BE mesh is shown along with its name. To delete any particular instance of a BE mesh right-click on *<Direct BE Mesh Name[Inst:#]>* and select **Delete**.

For each instance of a 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[Inst:#]>*.

**Xfm Matrix** This defines a  $4 \times 4$  coordinate transformation matrix. The current location of the instance of the BE mesh in the domain is obtained by applying this transformation to all the coordinates of the original mesh. The default transformation matrix is an identity matrix which places the BE mesh at its original position. The transformation matrix in Coustyx is represented in the following form:

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

where the first  $3 \times 3$  entries are for rotation transformation;  $\Delta x$ ,  $\Delta y$ ,  $\Delta z$  are the translations in  $x$ ,  $y$  and  $z$  directions.

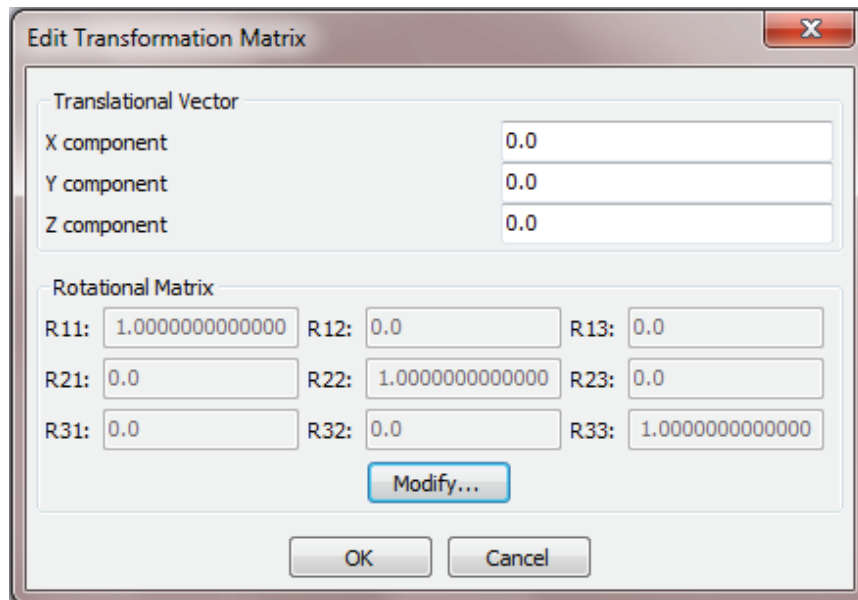
Note that the above transformation matrix is a general matrix that can be applied to an absolute vector (**s**), such as position vector, or to a relative vector (**n**), such as normal and velocity vector. These vectors are represented in Coustyx as follows:

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

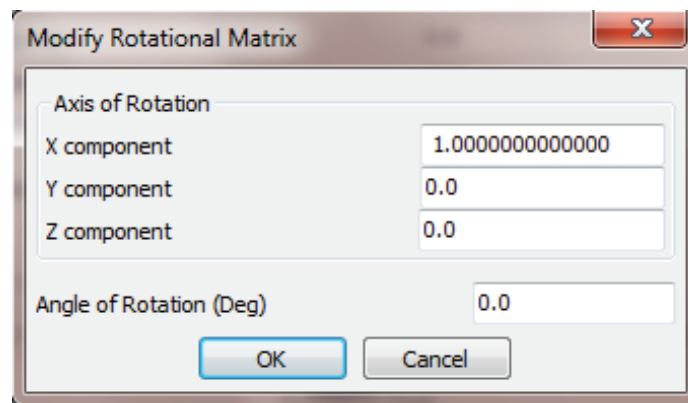
where  $x$ ,  $y$ ,  $z$  are components of absolute vector;  $n_x$ ,  $n_y$ ,  $n_z$  are components of relative vector.

To modify the transformation matrix right-click on **Xfm Matrix** and select **Edit...** A dialog box pops up as shown in Figure 4.29. Modify the translations in the  $x$ ,  $y$ ,  $z$  directions by changing the values of the Translational vector. The Rotational Matrix can only be modified through clicking on the **Modify...** button. A new dialog box as shown in Figure 4.30 pops up. A new rotational transformation can then be defined by specifying the axis and the angle (in degrees) of rotation.

**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 4.31). 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 5.6.2.1).



**Figure 4.29** Transformation Matrix Edit Dialog Box.



**Figure 4.30** Rotational Matrix Edit Dialog Box.



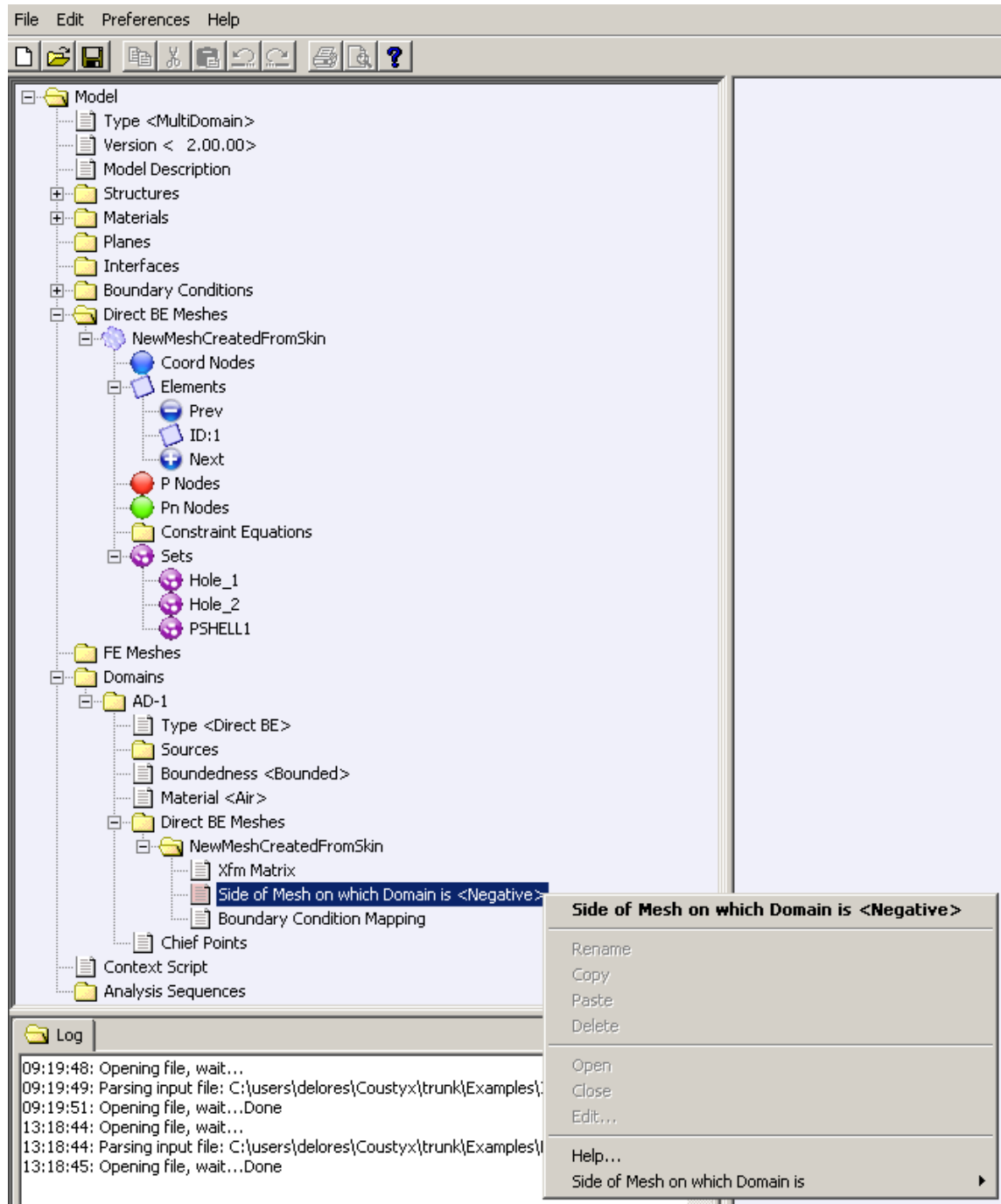


Figure 4.31 Side of Mesh on Domain Function

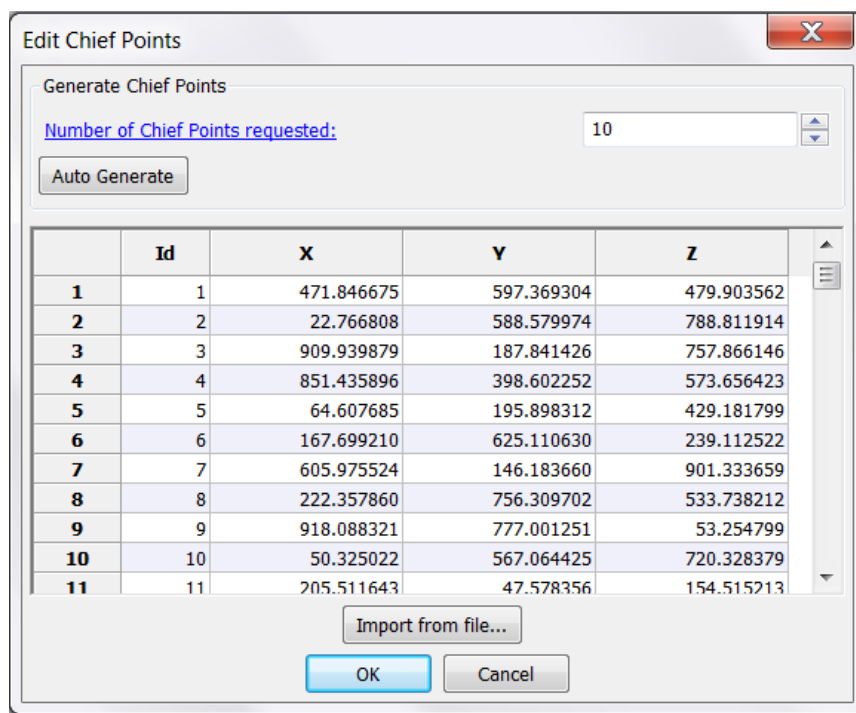
Note that in order to specify this flag we need to have all the elements of the mesh to be oriented consistently. For a *MultiDomain* model *Coustyx* automatically generates consistent BE mesh normals when skinned from a FE mesh.

**Boundary Condition Mapping** This lists all the boundary conditions applied on the elements of the mesh in this instance. Right-click and select **Open** to view the table showing elements id and their boundary conditions.

#### 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 unique ID, and coordinates X, Y, Z. Right-click and select **Add/Modify** to review or modify the list of Chief Points from a table. (Figure 4.32)



**Figure 4.32** Chief Points Edit Dialog Box.

**Auto Generate** Chief Points can be auto-generated randomly within the domain based on the number of points requested. Import Options window opens up after the generation of the chief points

**Number of Chief Points requested** Set the number of points to be generated within the interior of the domain

**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 4.33). 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 4.33)

**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 4.33). 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.

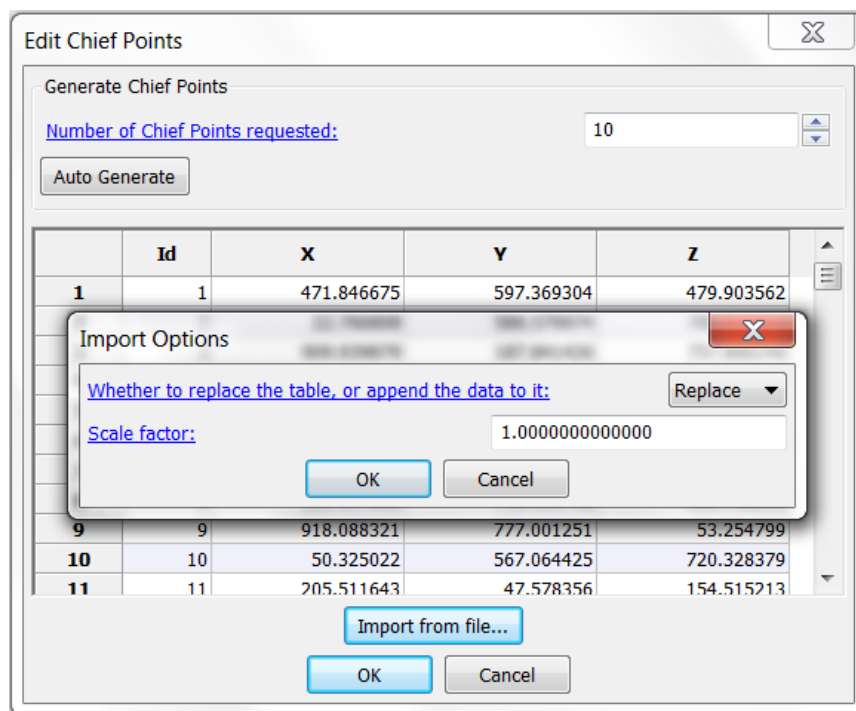


Figure 4.33 Import Chief Points from a File.

#### 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 variables/functions that can be used in the Context script are: *AngularFreq* ( $\omega$ ) – frequency in radians/sec, *GetSoundSpeed*("MaterialName") ( $c$ ) – speed of sound in the fluid medium with the same units as those defined in materials, *GetWaveNumber*("MaterialName") ( $k = \frac{\omega}{c}$ ), *GetAmbientDensity*("MaterialName") ( $\rho$ ) – 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 4.34).

#### 4.3.11. Analysis Sequences

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

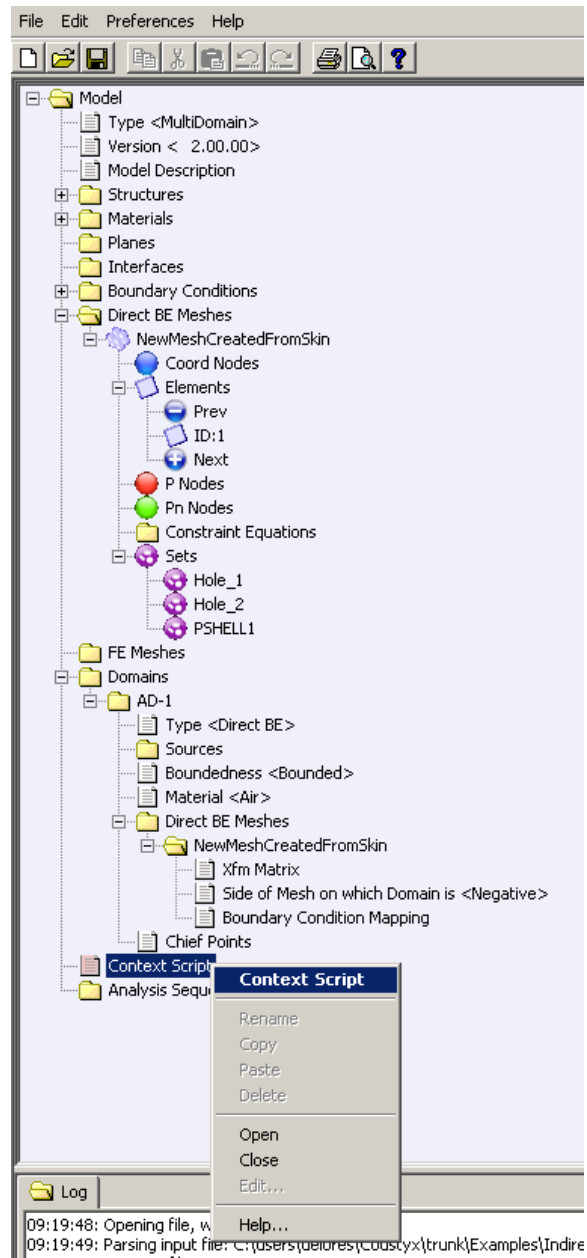


Figure 4.34 Context Script

**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

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 4.35).

**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 4.36.

**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.

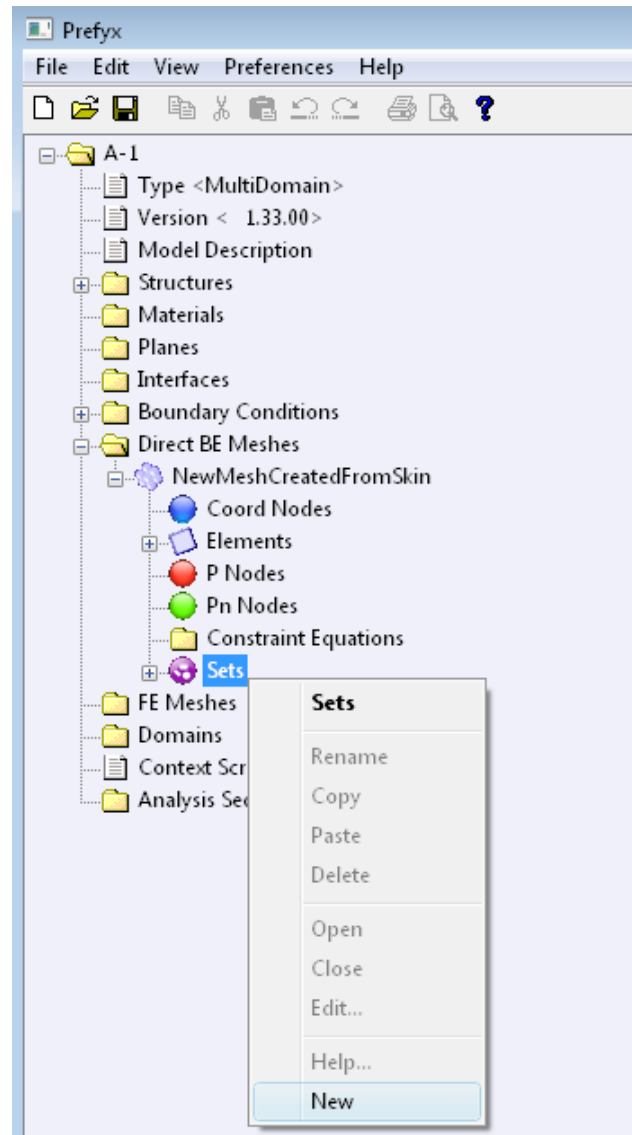
**Unselect** Unselects elements of the Set in the GUI.

**Add Selection to Set** Adds selected elements to the current Set.

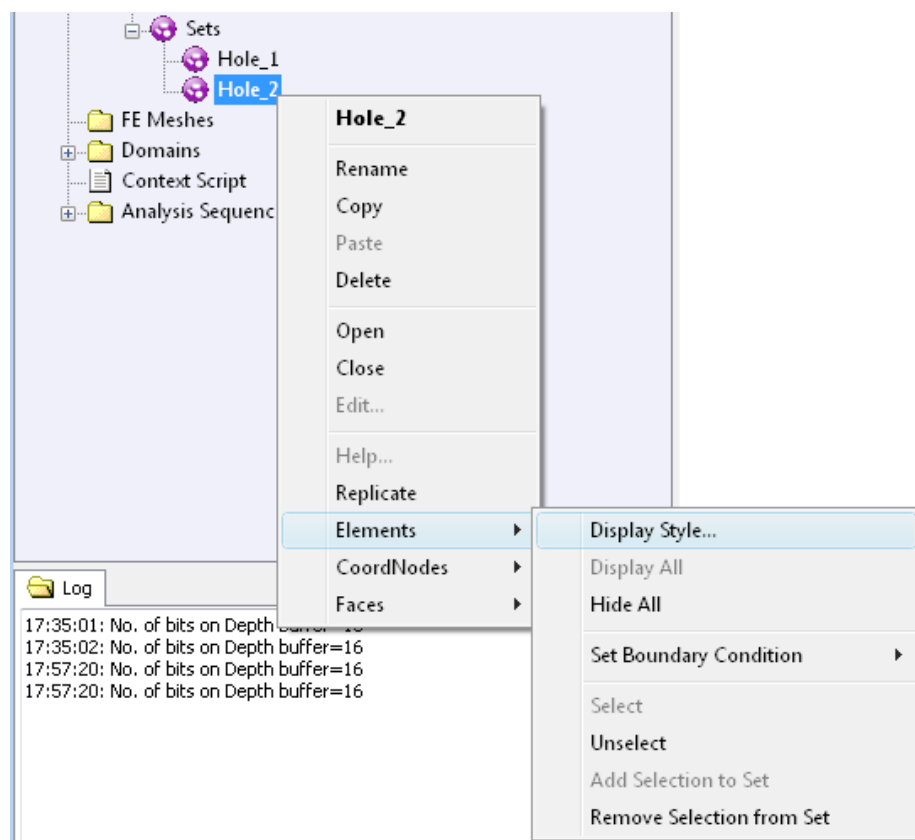
**Remove Selection from Set** Removes the selected elements from the current Set.

**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 4.37.

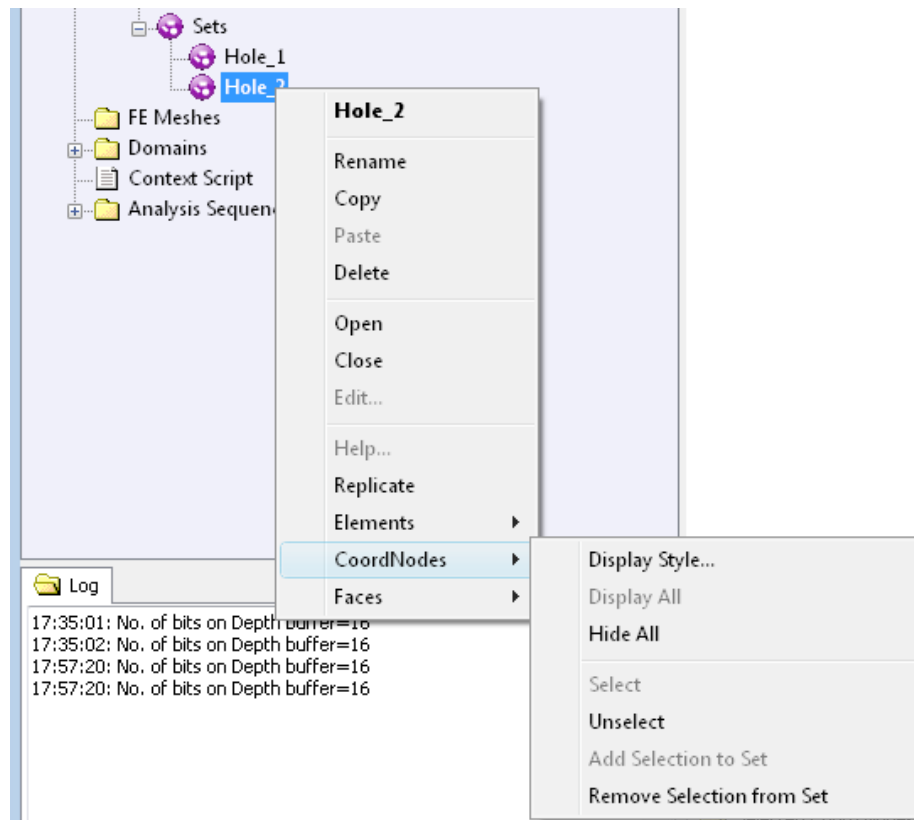
**Display Style** Changes the display style of the coordinate nodes in the Set. Refer to Section 4.2.3.2 for more details on each of the options in the [Display Style](#) dialog window.



**Figure 4.35** Creating a new set, a group of elements and nodes.



**Figure 4.36** Menu options for elements in a Set.



**Figure 4.37** Menu options for coordinate nodes in a Set.



**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.

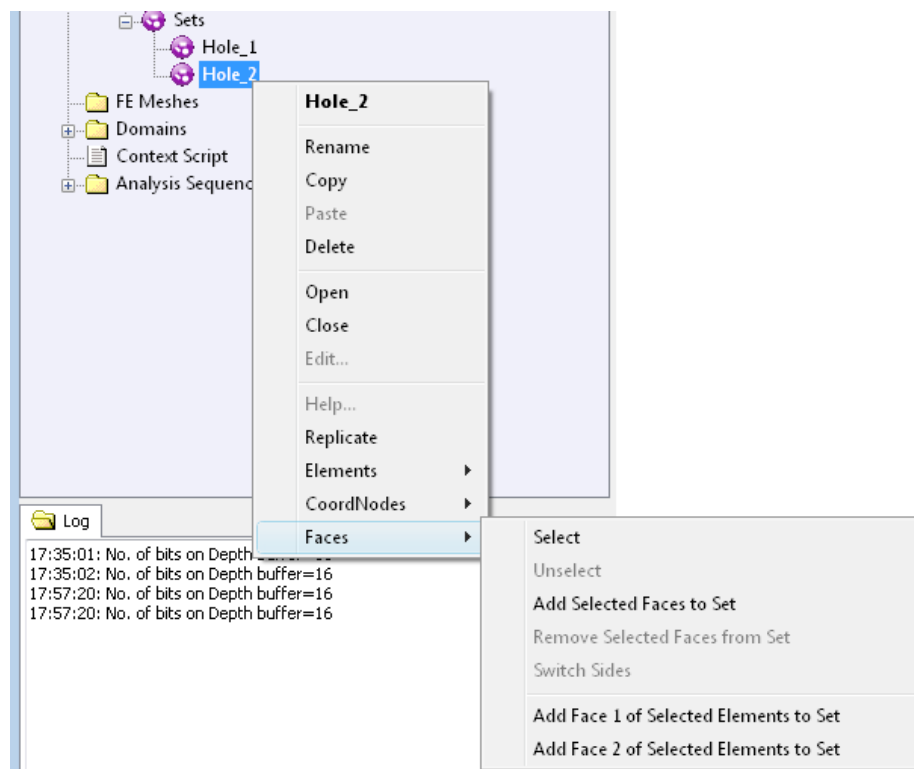
**Select** Selects all the coordinate nodes of the Set in the GUI.

**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 4.38.



**Figure 4.38** 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.

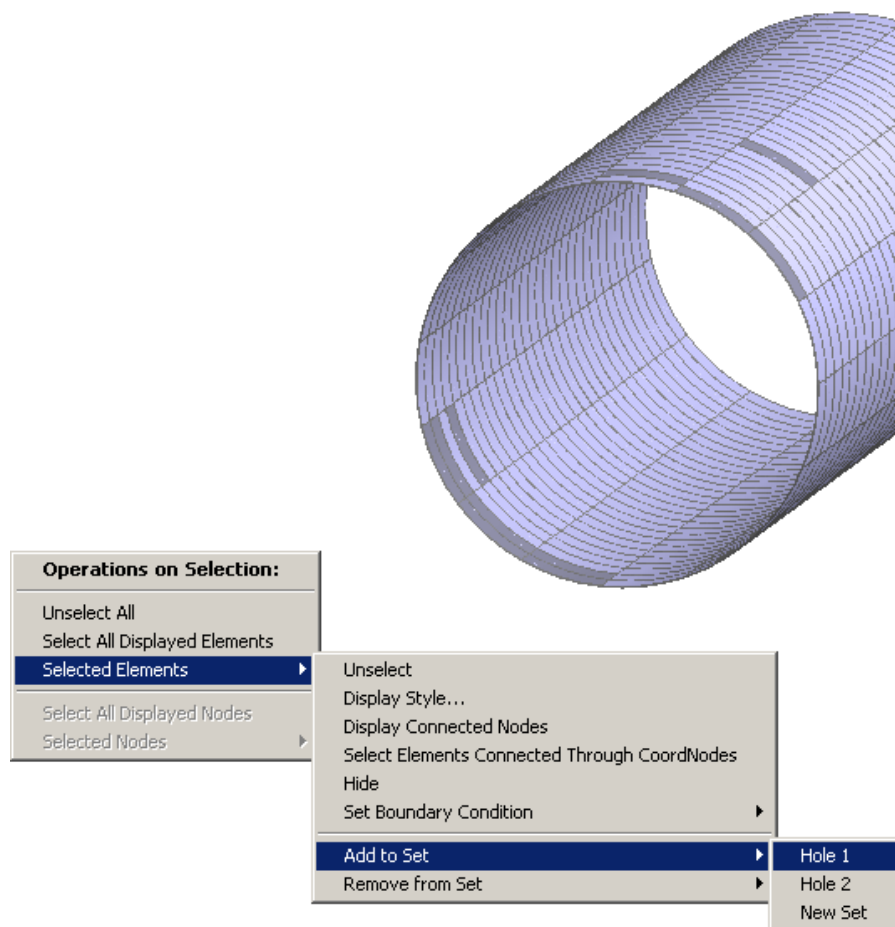
**Switch Sides** This option is available for BE meshes (surface elements) only. See Figure 6.23 for the description of Side 1 and Side 2 of a surface element. When selected, *Coustyx* switches the selection side of the element from the current face to its opposite. It modifies the component list of 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 6.23 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 6.23 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 4.39.



**Figure 4.39** Add elements to a set.

## 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 4.40).

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

$$A = jkZ_oV$$

where  $Z_o = \rho_o c$  is the characteristic impedance of the fluid medium,  $c$  is sound speed,  $\rho_o$  is ambient density,  $k = \omega/c$  is wave number,  $\omega$  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

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

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

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

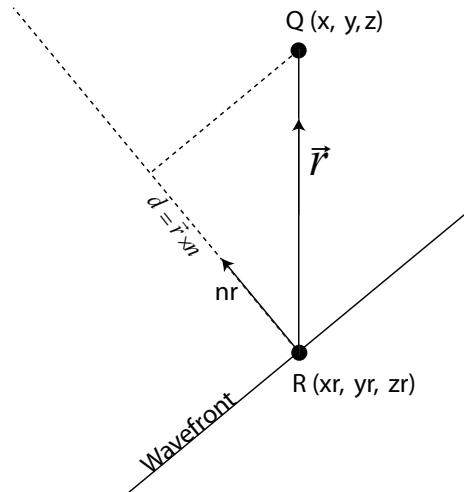
Figure 4.40 Monopole

**Position Vector** The location  $\mathbf{R}$  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 4.41** 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 4.41, Figure 4.42). The pressure  $p$  due to a plane wave at any point  $\mathbf{Q}(x, y, z)$  is

$$\tilde{p}(Q) = A e^{(jk\mathbf{r} \cdot \mathbf{n}_r)}$$

$$\mathbf{r} = \begin{Bmatrix} x - x_r \\ y - y_r \\ z - z_r \end{Bmatrix} \quad (4.5)$$

**Position Vector** The location  $\mathbf{R}$  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 4.43, Figure 4.44). The wave equation for a cylindrical wave, propagating in  $z$  direction is

**New Source**

ID: 1

Type: Plane Wave

Position Vector

X component: 0.0

Y component: 0.0

Z component: 0.0

Source Strength

Frequency Dependence Type: Constant

Real: 1.0000000000000000 Imaginary: 0.0

Direction Vector

X component: 0.0

Y component: 0.0

Z component: 1.0000000000000000

OK Cancel

Figure 4.42 Plane Wave

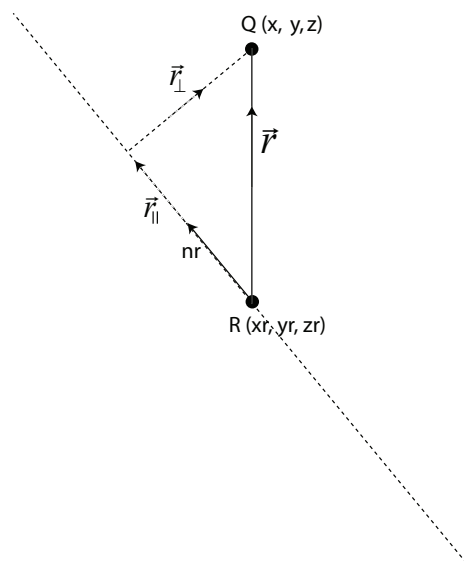


Figure 4.43 Description of a cylindrical wave acoustic source.

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

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

**Position Vector** The location  $\mathbf{R}$  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**.

The image shows a software dialog box titled "New Source". It contains several input fields and dropdown menus. The "ID" field is set to "1". The "Type" dropdown menu is set to "Cylindrical". Under the "Position Vector" section, the "X component", "Y component", and "Z component" fields are all set to "0.0". Under the "Source Strength" section, the "Frequency Dependence Type" dropdown is set to "Constant". The "Real" field is set to "1.0000000000000000" and the "Imaginary" field is set to "0.0". Under the "Direction Vector" section, the "X component" and "Y component" fields are set to "0.0", and the "Z component" field is set to "1.0000000000000000". At the bottom of the dialog are "OK" and "Cancel" buttons.

Figure 4.44 Cylindrical

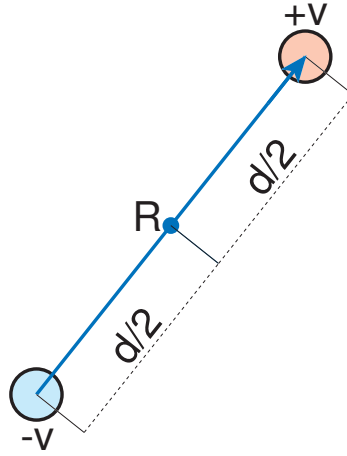
#### 4.5.4. Dipole

A dipole source is defined by a location, an amplitude,  $A$  (or dipole moment,  $D$ ) and the direction of the dipole. Figure 4.45 shows an acoustic dipole source at a location  $\mathbf{R}(x_r, y_r, z_r)$  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:

$$A = jkZ_o D$$

where  $Z_o = \rho_o c$  is the characteristic impedance of the fluid medium,  $c$  is sound speed,  $\rho_o$  is ambient density,  $k = \omega/c$  is wave number,  $\omega$  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  $L^4/T$ .



**Figure 4.45** 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  $+V$ .

The wave equation for a dipole source is given by

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

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**.

#### 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 4.47).

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

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:

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

where  $Z_o = \rho_o c$  is the characteristic impedance of the fluid medium,  $c$  is sound speed,  $\rho_o$  is ambient density,  $k = \omega/c$  is wave number,  $\omega$  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

Figure 4.46 Dipole

$V$  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.

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

where  $G(Q, R)$  is the Green's function evaluated between  $\mathbf{Q}$  and  $\mathbf{R}$ .

**Position Vector** The location  $\mathbf{R}$  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 4.48). Note that a valid user defined acoustic source should have a pressure field that satisfies the Helmholtz equation exactly.



New Source

ID: 1

Type: Quadrupole

Position Vector

X component 0.0

Y component 0.0

Z component 0.0

Quadrupole Strength Coefficients

Strength Type: Amplitude

Coefficient	Real	Imaginary
T[1][1]	1.00000000000000	0.0
T[1][2]	0.0	0.0
T[1][3]	0.0	0.0
T[2][1]	0.0	0.0
T[2][2]	0.0	0.0
T[2][3]	0.0	0.0
T[3][1]	0.0	0.0
T[3][2]	0.0	0.0
T[3][3]	0.0	0.0

OK Cancel

Figure 4.47 Quadrupole Dialog Box

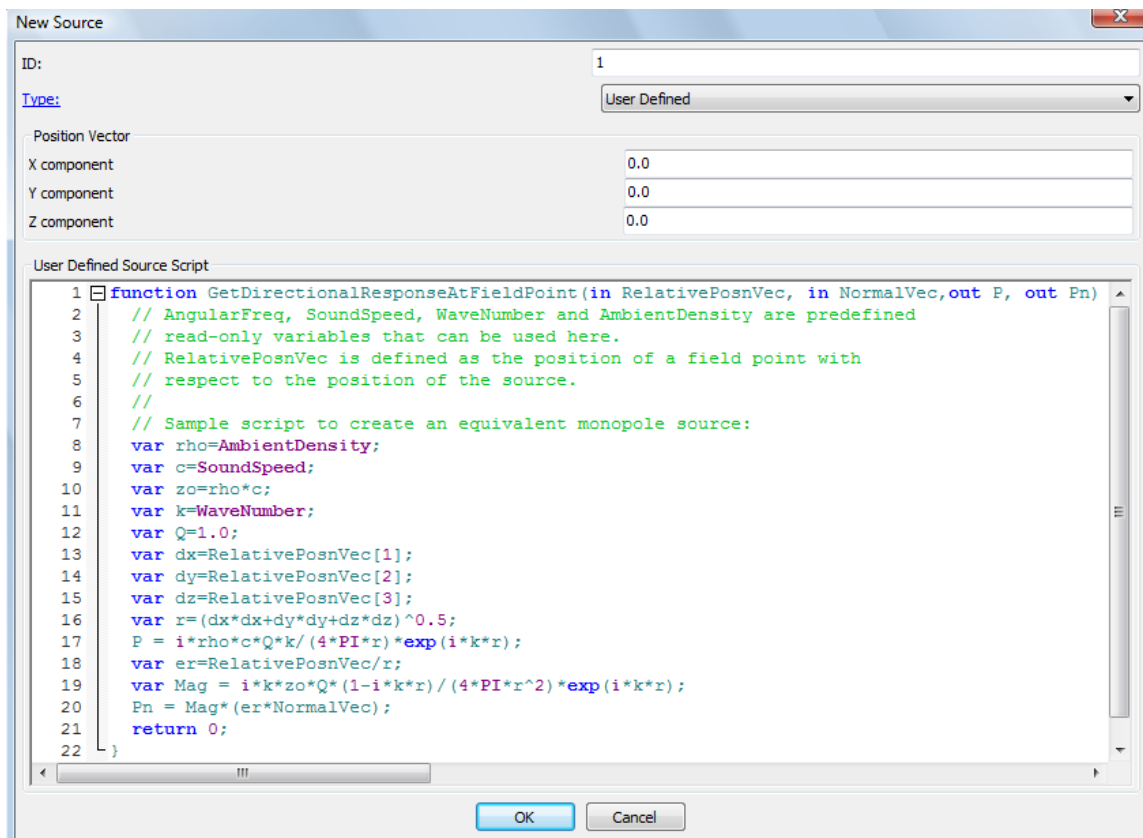


Figure 4.48 User Defined Acoustic Source

## 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, I-DEAS (Universal Files), RADIOSS (HyperMesh), Polygon File Format.

Select: **Model** → **Structures**.

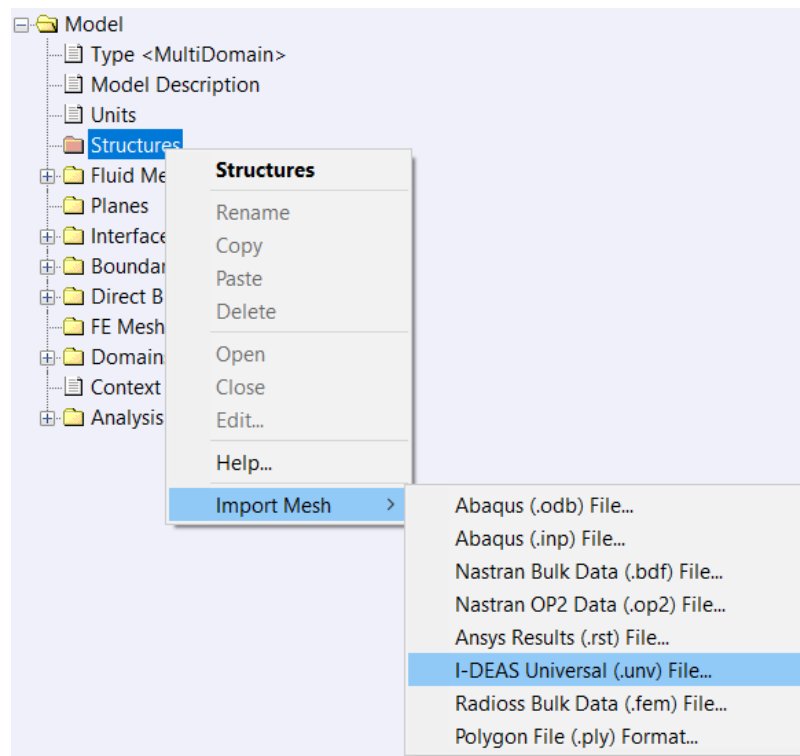
- To import FE mesh: Right-click on **Structures** and select **Import** → **Nastran Bulk Data (.bdf) File** (see Figure 5.1) to import mesh from Nastran bulk data format. The FE meshes from other data formats can be imported by selecting **Nastran OP2 Data (.op2) File**, **Abaqus (.odb) File**, **Abaqus (.inp) File**, **Ansys Results (.rst) File**, **I-DEAS Universal (.unv) File**, **Radioss Bulk Data (.fem) File**, or **Polygon File Format (.ply)**.
- 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 5.5). The other supported data formats are **Abaqus odb File**, **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.

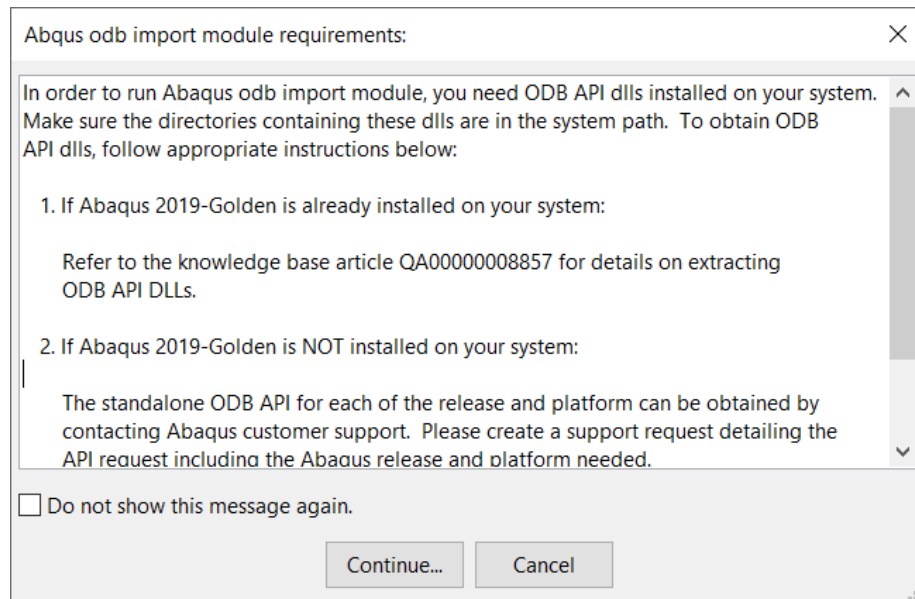


**Figure 5.1** Importing a finite element structure mesh.

**5.1.1.1. Abaqus \*.odb file** Abaqus Output Database file (\*.odb) is a results file from the Abaqus FE analysis. It includes information about both the geometry and the response. On import, Coustyx displays the dialog box (Figure 5.2) about the dll requirements for the odb reader. Coustyx requires the version Abaqus 2019-Golden odb API dlls for the odb reader. If Abaqus 2019-Golden is installed on the same machine as Coustyx, then make sure the directories containing the odb API dlls in the Abaqus installation folder are exposed to Coustyx through the system path. If Abaqus 2019-Golden is not installed on your machine, then download appropriate dlls from the Abaqus ftp site and copy them to the installation directory as instructed in the dialog box. After successfully reading the file, odb import options dialog box (Figure 5.3) appears. The user can select the part and the response at various frequencies to be imported into *Coustyx*.

The odb file created from older versions of Abaqus are upgraded before read into Coustyx. An upgrade dialog box (Figure 5.4) pops up requiring the filename of the new odb file. Coustyx performs the upgrade and then reads the data stored in the file.

**Load Response** Set this option true to load response data along with the structure mesh.



**Figure 5.2** Odb Reader Import Requirements Dialog Box.

**5.1.1.2. 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.

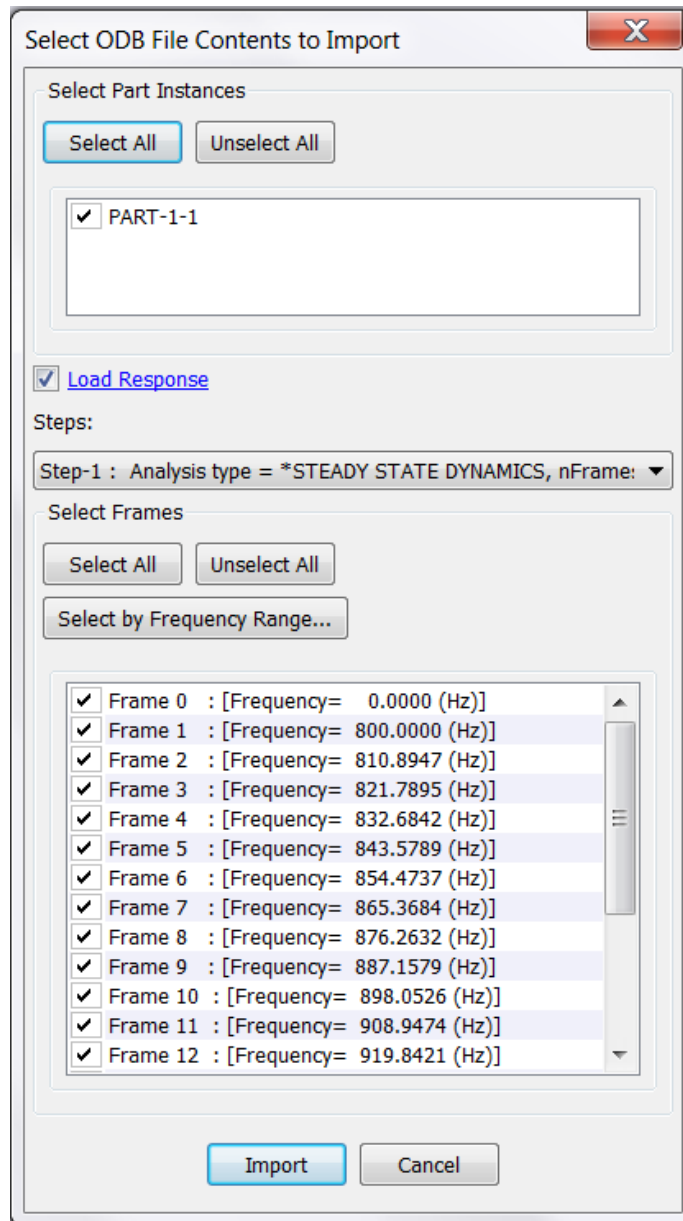
**5.1.1.3. Nastran Bulk Data \*.bdf files** The NASTRAN bulk data format (\*.bdf) is widely used in the industry and can be generated from a large number of commercially available FE mesh generators. The \*.bdf file contains information about the geometry of the mesh which is extracted by *Coustyx*.

**5.1.1.4. Nastran OP2 \*.op2 files** The NASTRAN Output2 format (\*.op2) is a results file that is produced as an output of Nastran FEA analysis. The \*.op2 file contains information about the mesh which is extracted by *Coustyx*.

**5.1.1.5. 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.

**5.1.1.6. 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.

**5.1.1.7. Radioss Bulk Data \*.fem files** The RADIOSS bulk data format (\*.fem) is widely used in Hyper-Mesh user community. The \*.fem file contains information about the geometry of the mesh which is extracted by *Coustyx*.



**Figure 5.3** Odb Reader Import Options Dialog Box.

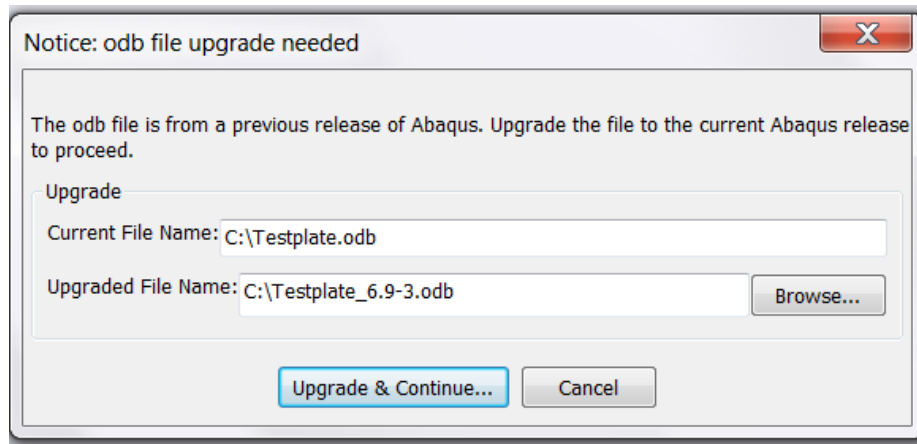


Figure 5.4 Odb File Upgrade Dialog Box.

**5.1.1.8. Polygon File Format \*.ply** The Polygon File Format (\*.ply) is widely used to store graphical objects data from 3-D scanners. The \*.ply file contains information about the vertices and faces of the object that are extracted by *Coustyx*.

## 5.1.2. Loading Frequency Response Data

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

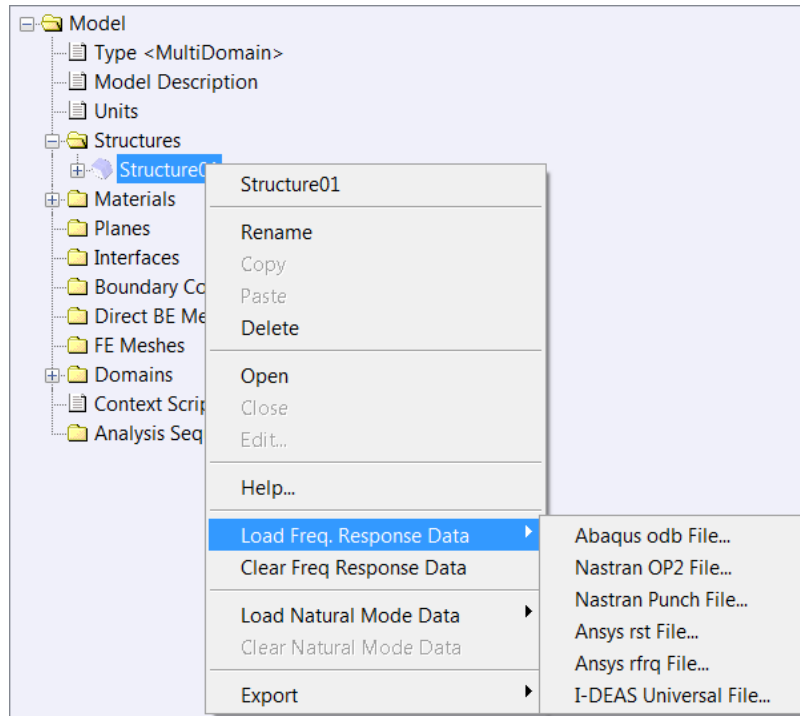
Right-click on <Structure Mesh Name> and select the frequency response data from any of the file format options through **Load Freq. Response Data** (refer to Figure 5.5). If the data is read successfully, a new dialog box with a list of frequencies available in the data appears. See Figure 5.6. The 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. Abaqus odb File** Abaqus \*.odb file is the output database file from the Abaqus FE analysis. *Coustyx* extracts the frequencies and the velocity response at these frequencies from this file. Refer to Section 5.1.1.1 for more details.

**5.1.2.2. Nastran OP2 File** The Output2 (OP2) file is a binary output file from the NASTRAN FE analysis. 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.

**5.1.2.3. 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.

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



**Figure 5.5** Loading frequency response data into *Coustyx* model.

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

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

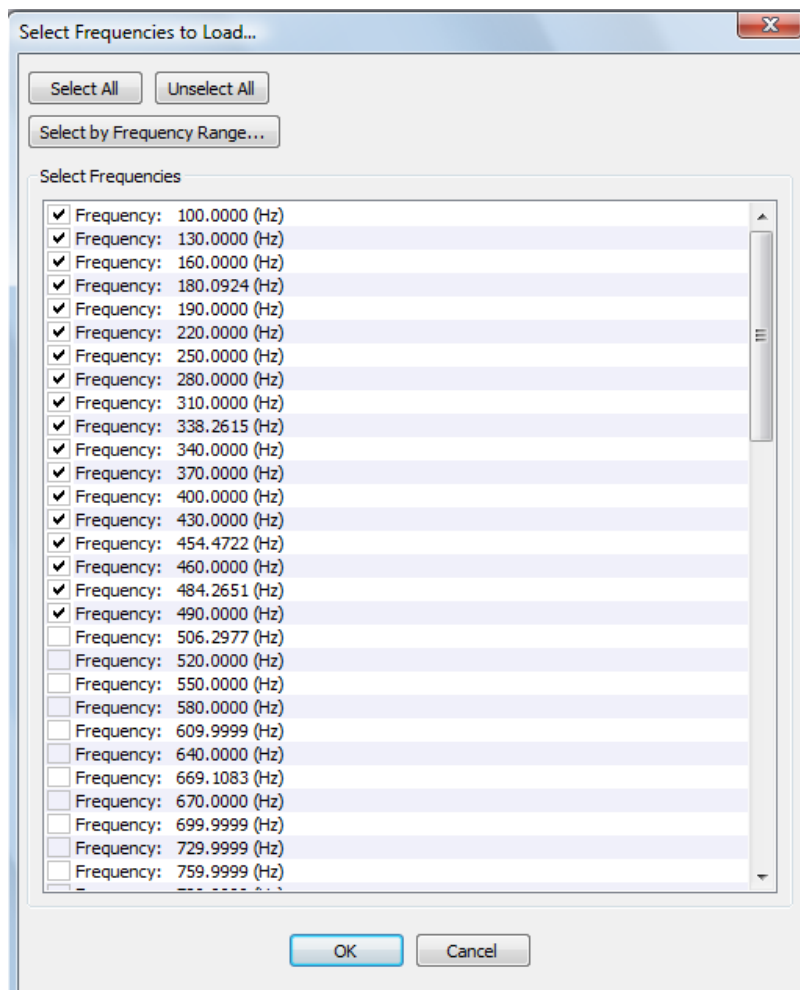
### 5.1.3. Loading Natural Mode Data

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

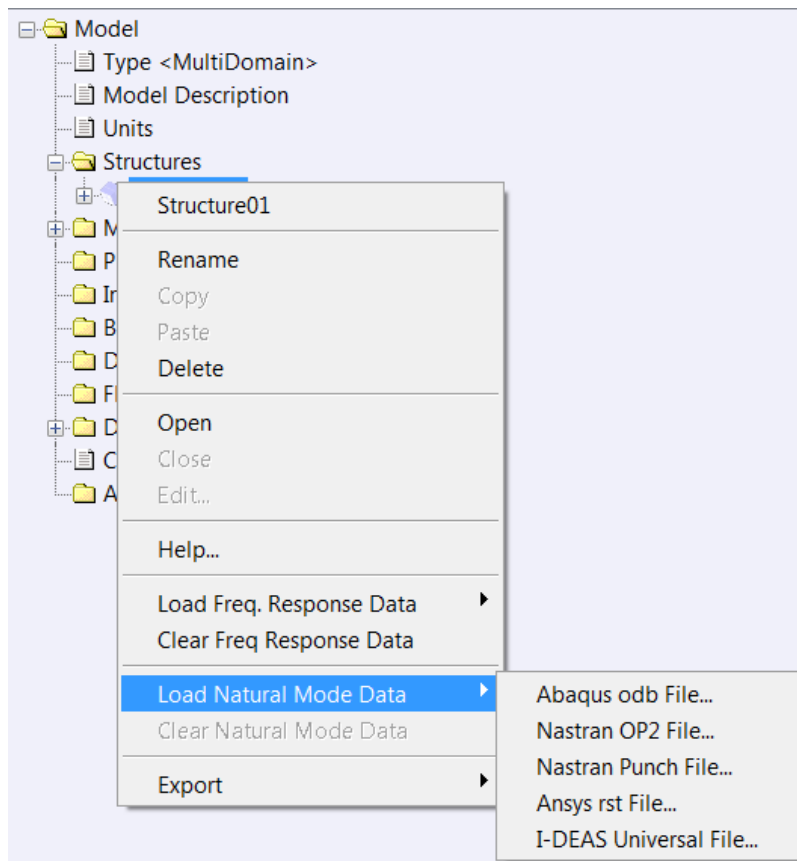
Right-click on the desired structure mesh <Structure Mesh Name> and select **Load Natural Mode Data** to load the natural modes data from a finite element analysis (refer to Figure 5.7). If the data is read successfully, a new dialog box with a list of natural frequencies available in the data appears. See Figure 5.8. 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: *Abaqus odb file*, *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.

**5.1.3.1. Copy to Frequency Response Data** Select: **Model** → **Structures** → <Structure Mesh Name> → **Natural Mode Data** → <Mode>.

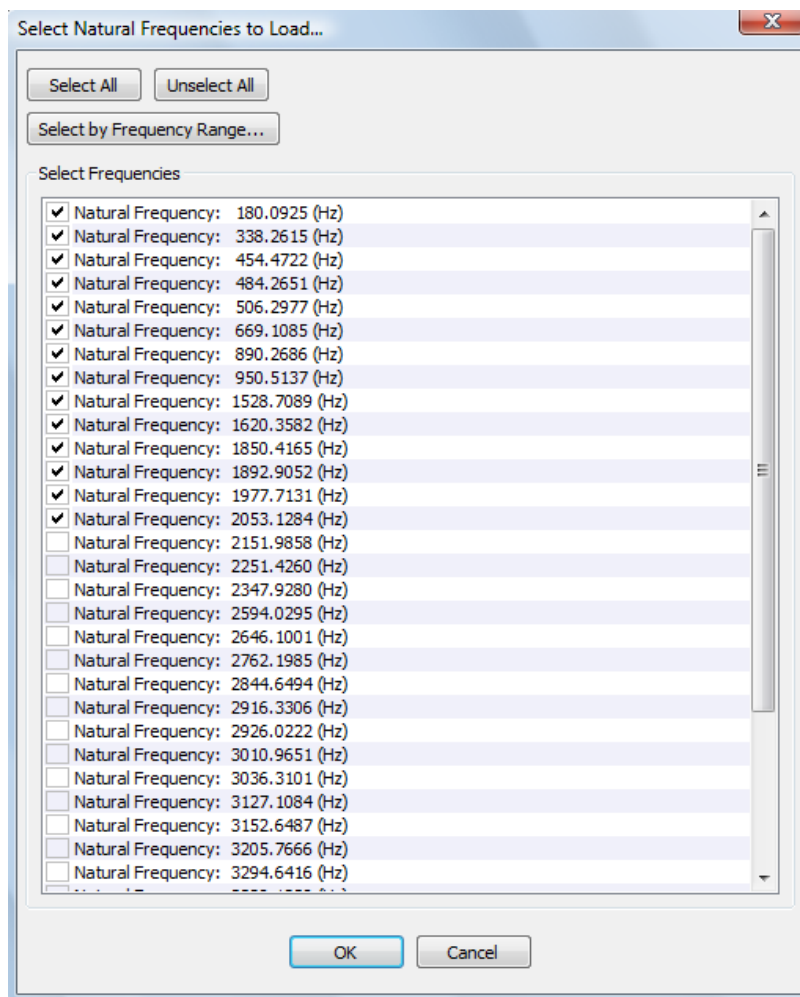




**Figure 5.6** Select frequencies to load dialog box.

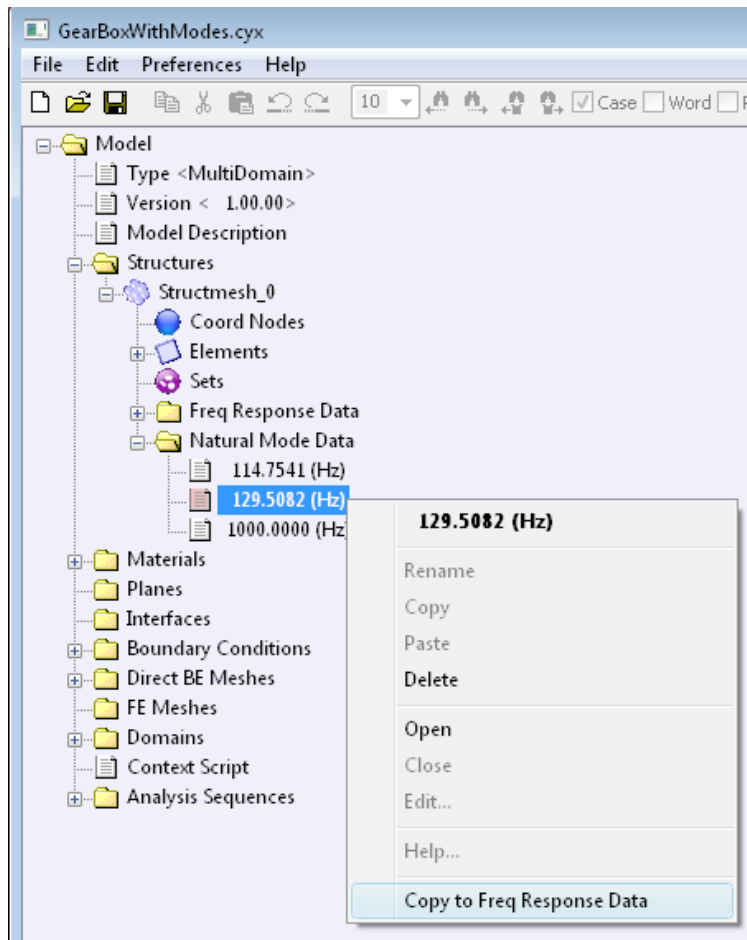


**Figure 5.7** Loading natural mode data into *Coustyx* model.

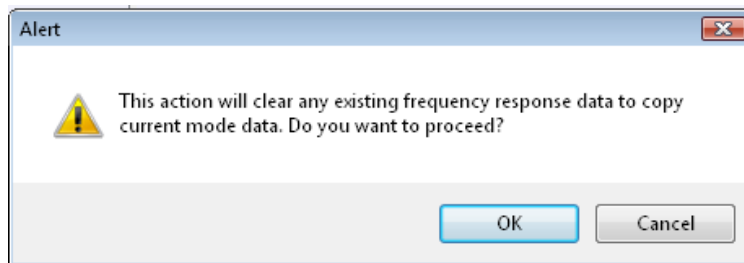


**Figure 5.8** Select natural frequencies to load dialog box.

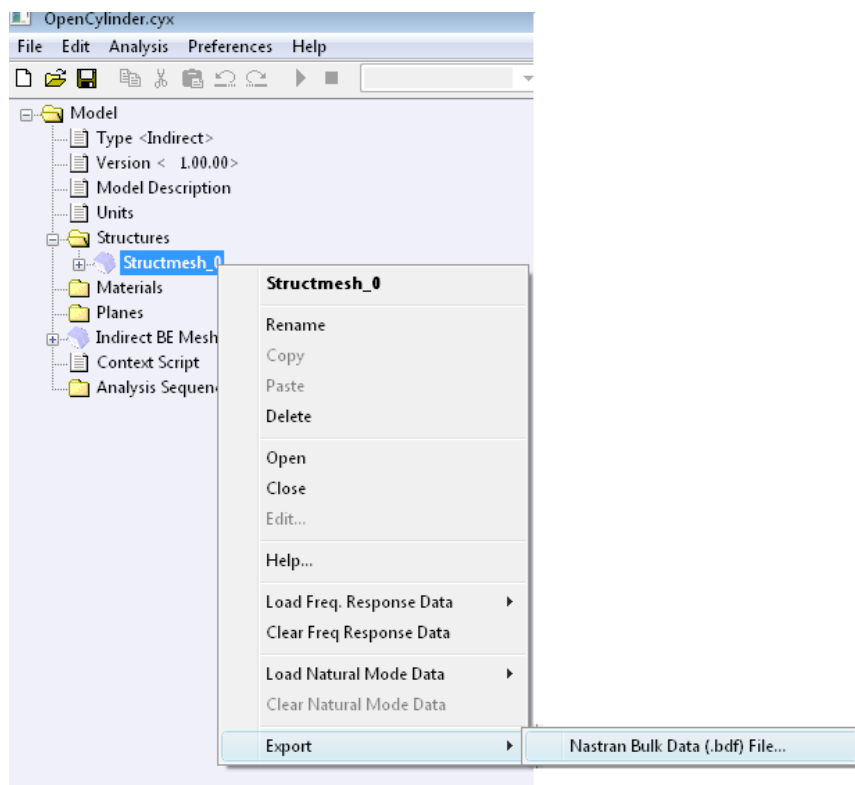
Right-click on the desired mode  $\langle Mode \rangle$  and select **Copy to Freq Response Data** (refer to Figure 5.9). The mode information from the **Natural Mode Data** is copied to the **Freq Response Data**. This function is useful when the user wants to perform acoustic analysis over a frequency 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 to Section 6.5.7.1 and Section 6.6.3.5). *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 5.10). 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 5.9** Copy the mode data to the frequency response data.



**Figure 5.10** Alert message displayed before copying mode data to frequency response data.



**Figure 5.11** Export mesh.

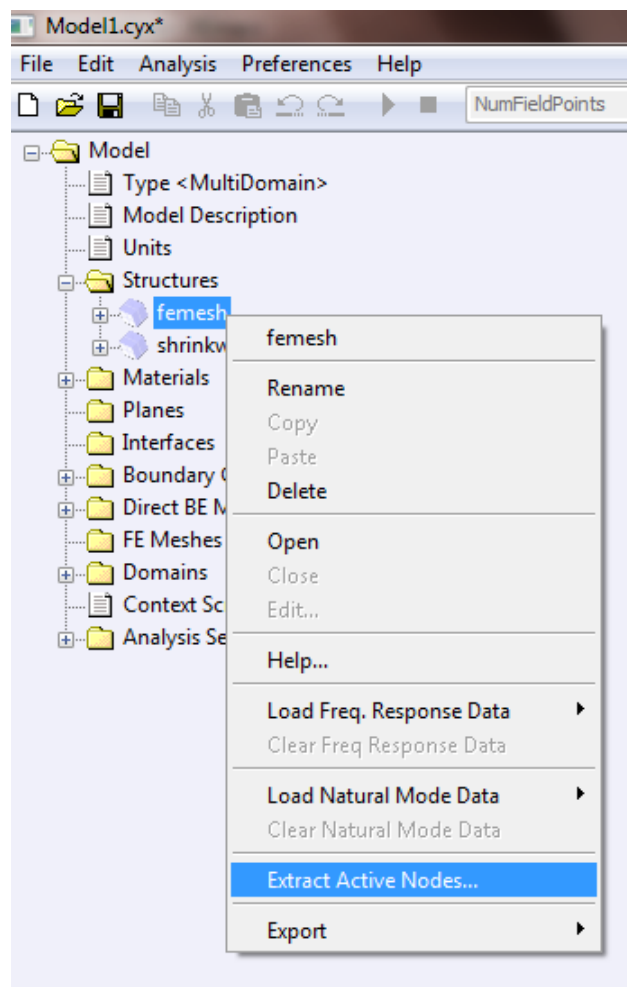
## 5.2. Exporting Mesh

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 5.11).

## 5.3. Extract Active Nodes

Use this feature to extract nodes from the current structure mesh that actively interact with nodes in another structure mesh (mismatched mesh), or with nodes in a frequency response data file or an invacuo mode data file. A new structure mesh is then created using these active nodes. To extract active nodes of a structure mesh: Right-click on the *<Structure Mesh Name>* and select **Extract Active Nodes...** (see Figure 5.12). A new dialog box as shown in Figure 5.13 will appear.

- **Source** lists the three different source options available: [Frequency Response Data File](#), [Natural Mode Data File](#), and [Mismatched Mesh](#).
- **New Structure Name** Create a new structure mesh by this name with the extracted active nodes.

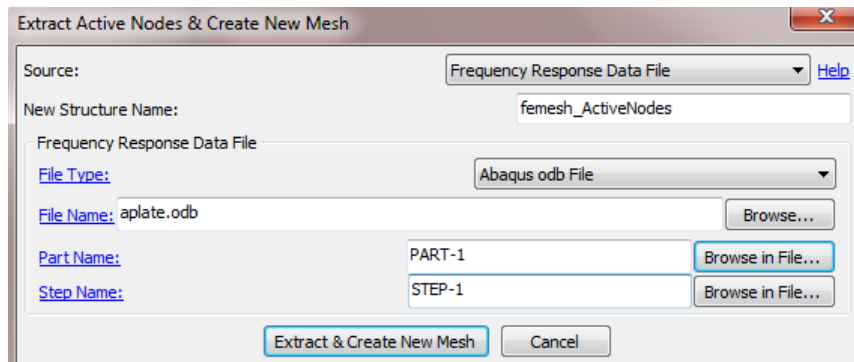


**Figure 5.12** Extract active nodes menu.

### 5.3.1. Frequency Response Data File

Choose this option to extract active nodes of the current structure mesh by matching with the nodes in the frequency response data file. The nodes ids common to both the structure mesh and the frequency response data are considered active. A new structure mesh is then created with extracted nodes (Note that this new mesh will not contain any elements).

- **File Type** Select appropriate FEA output data file type from the drop down menu. Supported FEA output data file formats are: **Abaqus odb File**, **Nastran OP2 File**, **Nastran Punch File**, **Ansys rst File**, **Ansys rfrq File**, and **I-DEAS Universal File**.
- **File Name** Provide the full path of the FEA data file name. Use the **Browse** button to browse folders for the file name.
- **Part Name** This option is applicable only for **Abaqus odb File** format. Provide the part name from which to load the data. Use the **Browse in File** button to browse for all the available parts in the file provided in **File Name**.
- **Step Name** This option is applicable only for **Abaqus odb File** format. Provide the step name from which to load the data. Use the **Browse in File** button to browse for all the available parts in the file provided in **File Name**.



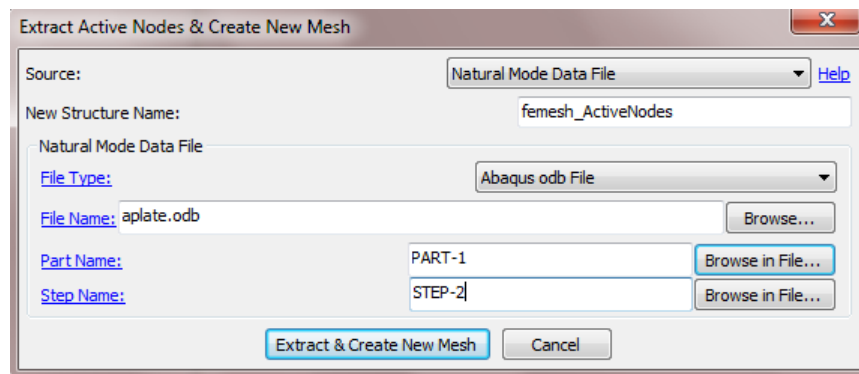
**Figure 5.13** Extract active nodes & Create new mesh dialog with frequency response data file as the source.

### 5.3.2. Natural Mode Data File

Choose this option to extract active nodes of the current structure mesh by matching with the nodes in the natural mode data file. The nodes ids common to both the structure mesh and the natural mode data are considered active. A new structure mesh is then created with extracted nodes (Note that the mesh will not contain any elements).

- **File Type** Select appropriate FEA output data file type from the drop down menu. Supported FEA output data file formats are: **Abaqus odb File**, **Nastran OP2 File**, **Nastran Punch File**, **Ansys rst File**, **Ansys rfrq File**, and **I-DEAS Universal File**.
- **File Name** Provide the full path of the FEA data file name. Use the **Browse** button to browse folders for the file name.
- **Part Name** This option is applicable only for **Abaqus odb File** format. Provide the part name from which to load the data. Use the **Browse in File** button to browse for all the available parts in the file provided in **File Name**.

- **Step Name** This option is applicable only for **Abaqus odb File** format. Provide the step name from which to load the data. Use the **Browse in File** button to browse for all the available parts in the file provided in **File Name**.



**Figure 5.14** Extract active nodes & Create new mesh dialog with natural mode data file as the source.

### 5.3.3. Mismatched Mesh

Choose this option to extract surface nodes of the current structure mesh that participate in the structure velocity interpolation when the mismatched mesh (specified by **Mismatched Structure Name**) is used as the boundary element mesh. The mismatched mesh is a surface mesh that is usually created by shrink wrapping the structure mesh for use in acoustic analysis. A new structure mesh is created with the extracted nodes. The user can use the new structure mesh (with list of active nodes) for performing data expansion to create FEA results file. The results file could later be imported into Coustyx and be used to apply the structure velocity boundary condition.

- **Mismatched Structure Name** Select the name of the mismatched mesh from the list of available structures. Note that this mesh is a surface mesh that is usually generated by shrink wrapping the original structure mesh.
- **Interpolation Options for Mismatched Mesh** Refer to Section 6 for details.

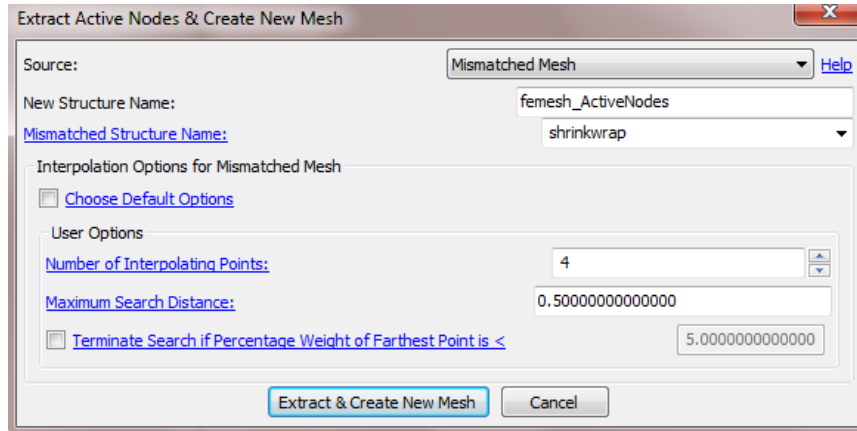
## 5.4. Forced Response Analysis using Modal Superposition

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 the modal analysis performed using any external FE packages. 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.





**Figure 5.15** Extract active nodes & Create new mesh dialog with mismatched mesh as the source.

- **Perform modal superposition.** Select analysis frequencies and the participating modes to compute forced response.

#### 5.4.1. 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 orthonormalized 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  $\mathbf{A} = [\mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_n]$  is the modal matrix containing natural mode vectors  $\mathbf{a}_i$ ,  $\mathbf{M}$  and  $\mathbf{K}$  are finite element mass and stiffness matrices,  $\Omega^2 = \text{Diag}(\omega_1^2, \dots, \omega_n^2)$ ,  $\omega_i$  is the natural frequency, and  $\mathbf{I}$  is a unit matrix.

#### 5.4.2. Modify Modal Damping

Modal damping is the only damping model allowed in *Coustyx*. Modal damping is estimated from the damping ratios  $\zeta$ . 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,  $\mathbf{A}^T \mathbf{C} \mathbf{A} = \text{Diag}(\dots, 2\zeta_i \omega_i, \dots)$ , where  $\mathbf{A}$  is the modal matrix,  $\zeta_i$  and  $\omega_i$  are damping ratio and natural frequency at the  $i^{\text{th}}$  mode.

Follow the instructions below to edit modal damping ratios.

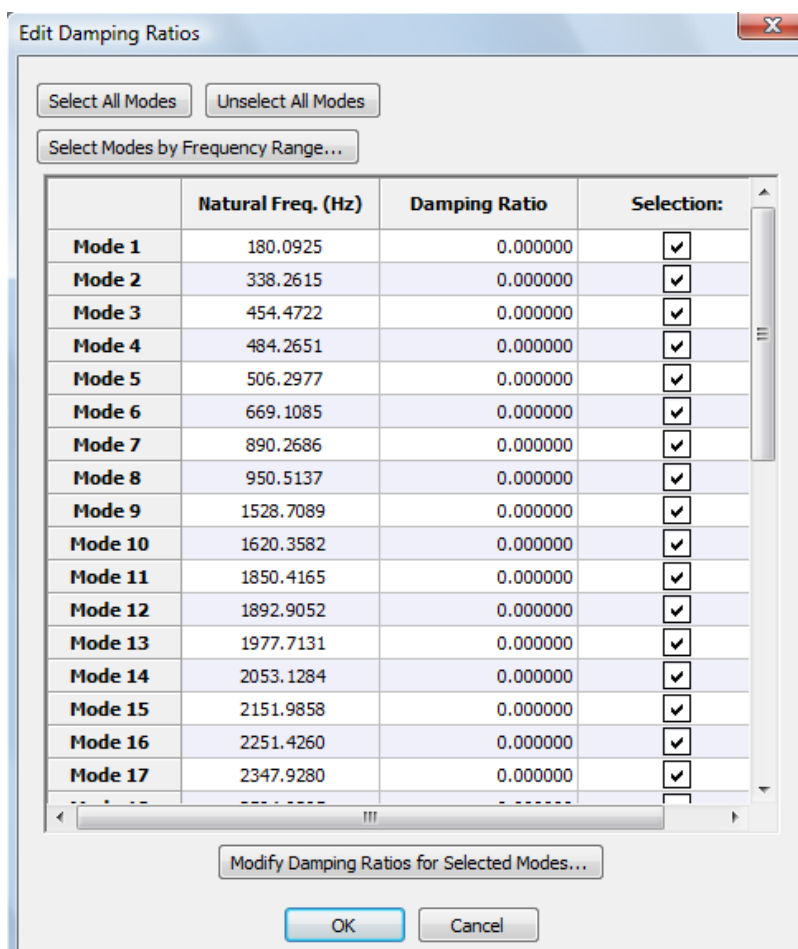
Select: **Model** → **Structures** → *<Structure Mesh Name>* → **Natural Mode Data**.

Right-click on **Natural Mode Data** and select **Edit Damping Ratios...**. A dialog window shown in Figure 5.16 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...**

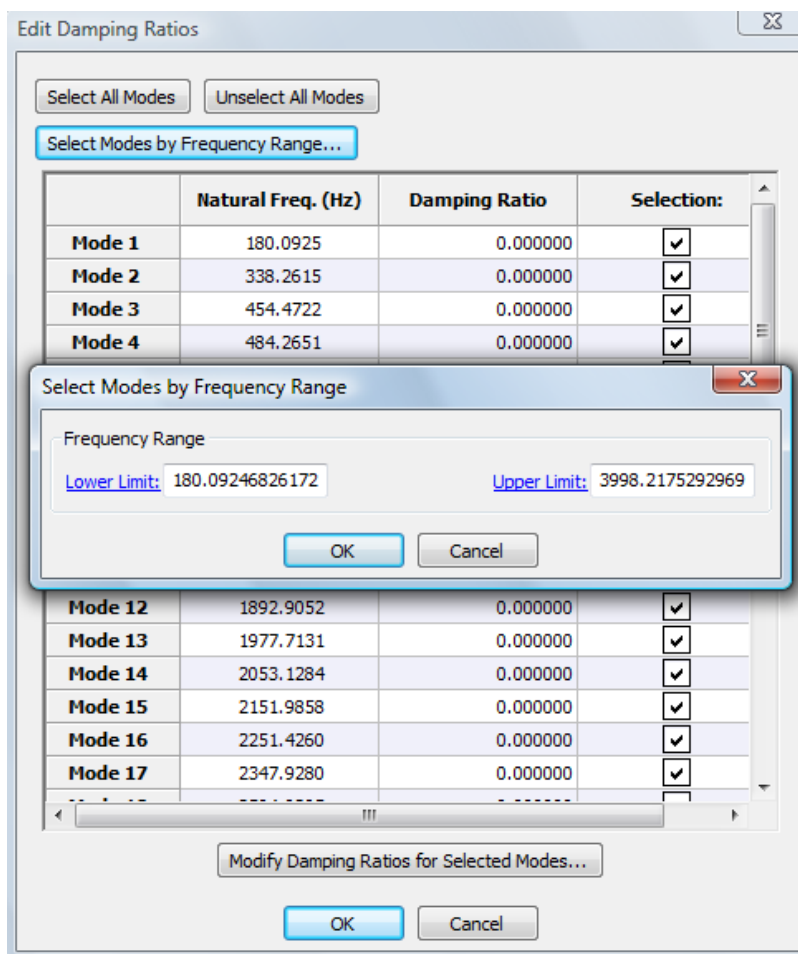
There are multiple ways to select modes in the table for editing: (a) check or uncheck the selection box against a mode in the table, (b) use buttons **Select All Modes** or **Unselect All Modes** to select or unselect all modes in the table, (c) use button **Select Modes by Frequency Range...** to select modes lying within a frequency range (Figure 5.17). Set the upper and lower frequency limits and press OK to select modes within the frequency limits.

The button **Modify Damping Ratios for Selected Modes...**, when pressed, opens up a dialog box as shown in Figure 5.18. 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 5.18. For example, to define



**Figure 5.16** Damping ratios edit dialog box.



**Figure 5.17** Select modes by frequency range.

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.

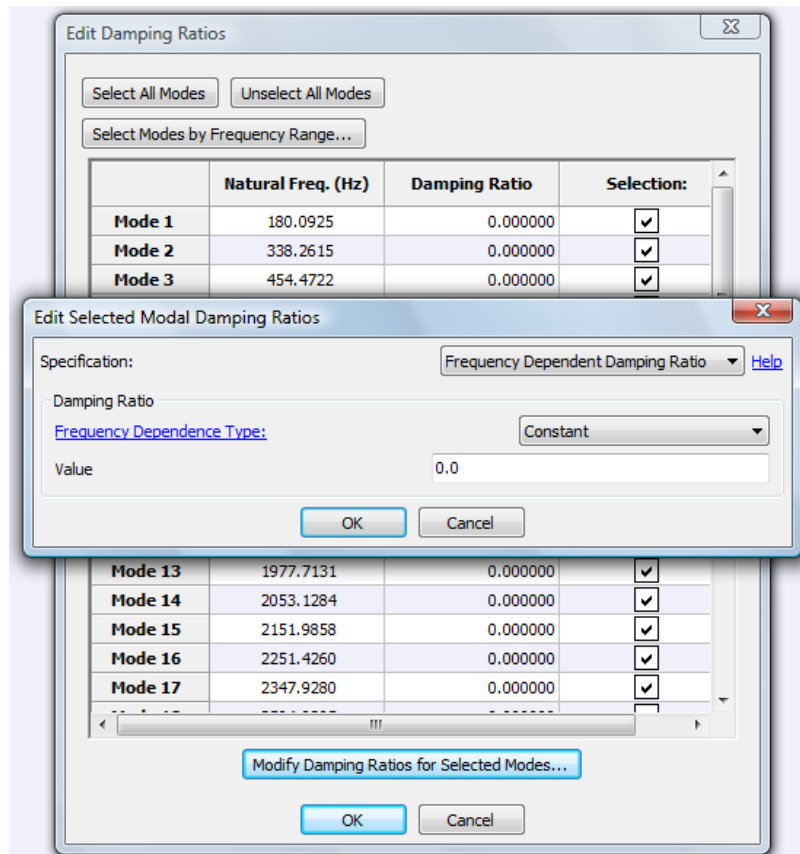


Figure 5.18 Edit selected modal damping ratios using Frequency dependent values.

**Rayleigh Damping** Rayleigh damping coefficients, **Alpha**  $\alpha$  and **Beta**  $\beta$  are defined through this option. See Figure 5.19. Damping ratio for each selected mode is computed from  $\alpha$  and  $\beta$  as follows:

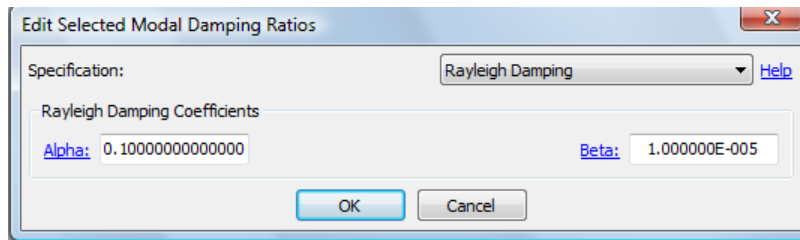
$$\zeta_i = \frac{\alpha}{2\omega_i} + \frac{\beta\omega_i}{2}$$

where  $\zeta_i$  is damping ratio and  $\omega_i$  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.

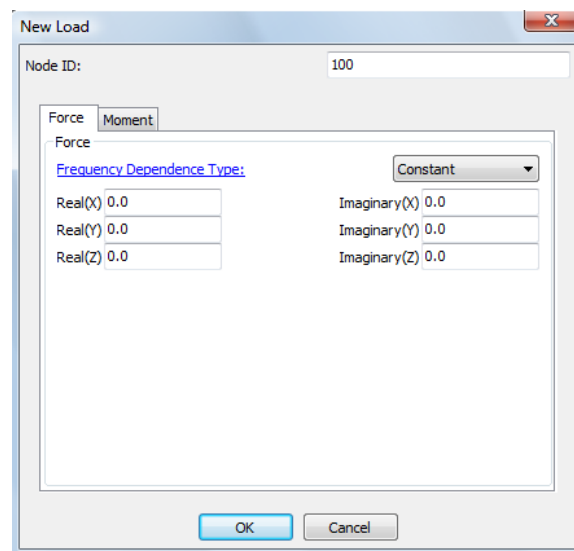
### 5.4.3. Apply Loads

Select: **Model**  $\rightarrow$  **Structures**  $\rightarrow$   $\langle$ Structure Mesh Name $\rangle \rightarrow$  **Loads**.

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



**Figure 5.19** Edit selected modal damping ratios using Rayleigh damping coefficients.



**Figure 5.20** Apply nodal force and moment.

moment applied at that node. To specify a new load, right-click on **Loads** and select **New**. A dialog box as shown in Figure 5.20 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**.

#### 5.4.4. 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,

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

where  $\omega$  is the analysis frequency,  $\mathbf{A} = [\mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_n]$  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,  $\Omega^2 = \text{Diag}(\omega_1^2, \omega_2^2, \dots, \omega_n^2)$ ,  $\omega_i$  is the natural frequency,  $\mathbf{u}$  is the generalized displacement vector,  $\mathbf{f}$  is the force vector,  $\zeta_i$  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*.

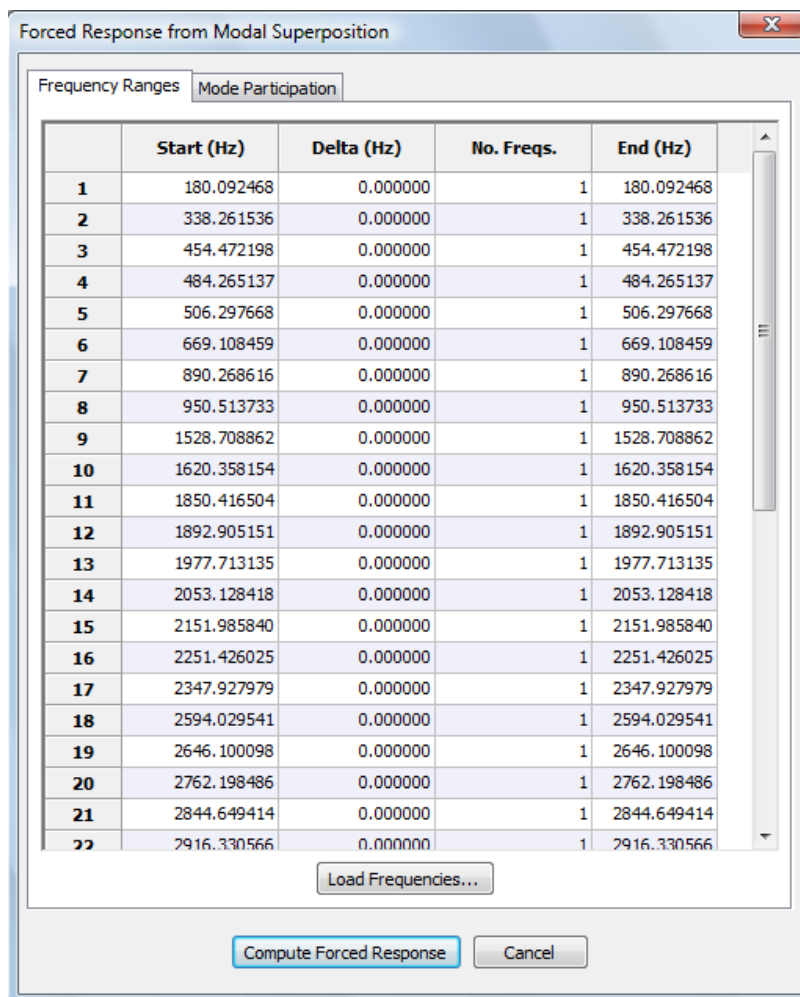
Select: **Model** → **Structures** → *<Structure Mesh Name>* → **Freq Response Data**.

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 5.21 appears. Enter analysis frequencies using the table in the **Frequency Range** tabbed window. Refer Section 7.2.1.6 for more details about the frequency range table. Select participating modes from the list in the **Mode Participation** tabbed window. See Figure 5.22. 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.

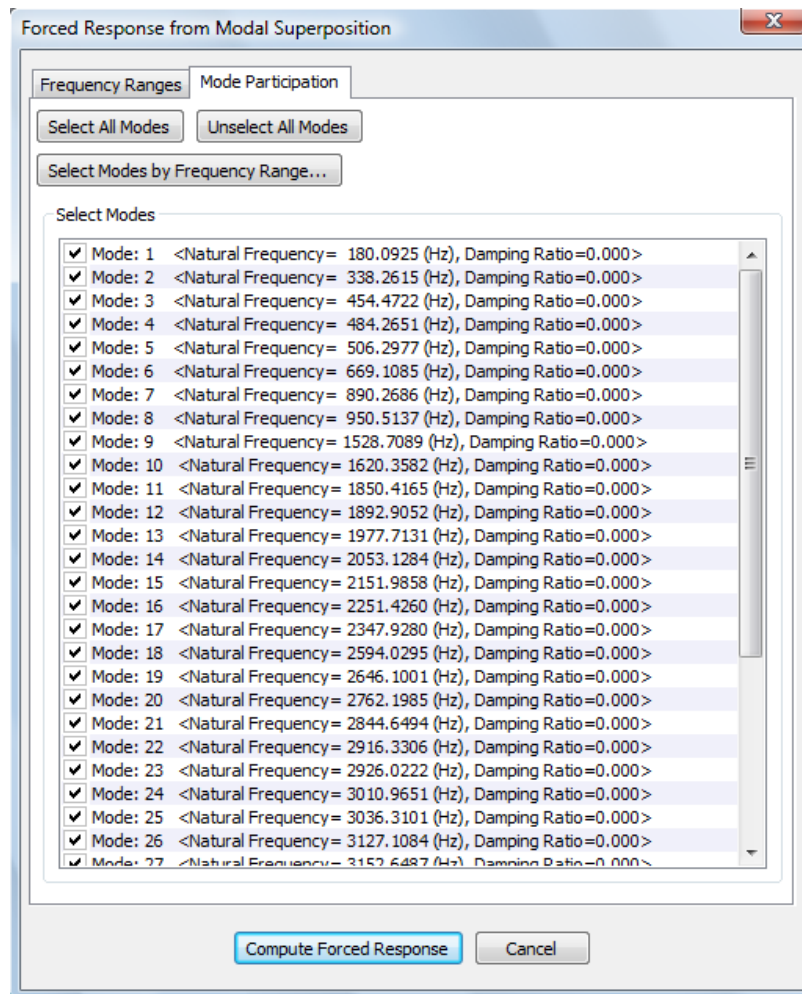
Once the analysis frequencies and the participating modes are selected, press the button **Compute Forced Response** to start the forced response harmonic analysis using modal superposition. The analysis results are stored into the model tree member **Freq Response Data**. This data could later be applied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

#### 5.5. Fill Hole

The **Fill Hole** tabbed window is located at the bottom pane of the *Mesh Viewer* window (Figure 5.23). *Coustyx* uses an *Optimized Delaunay Triangulation Method* to fill holes. The parameters used are shown in

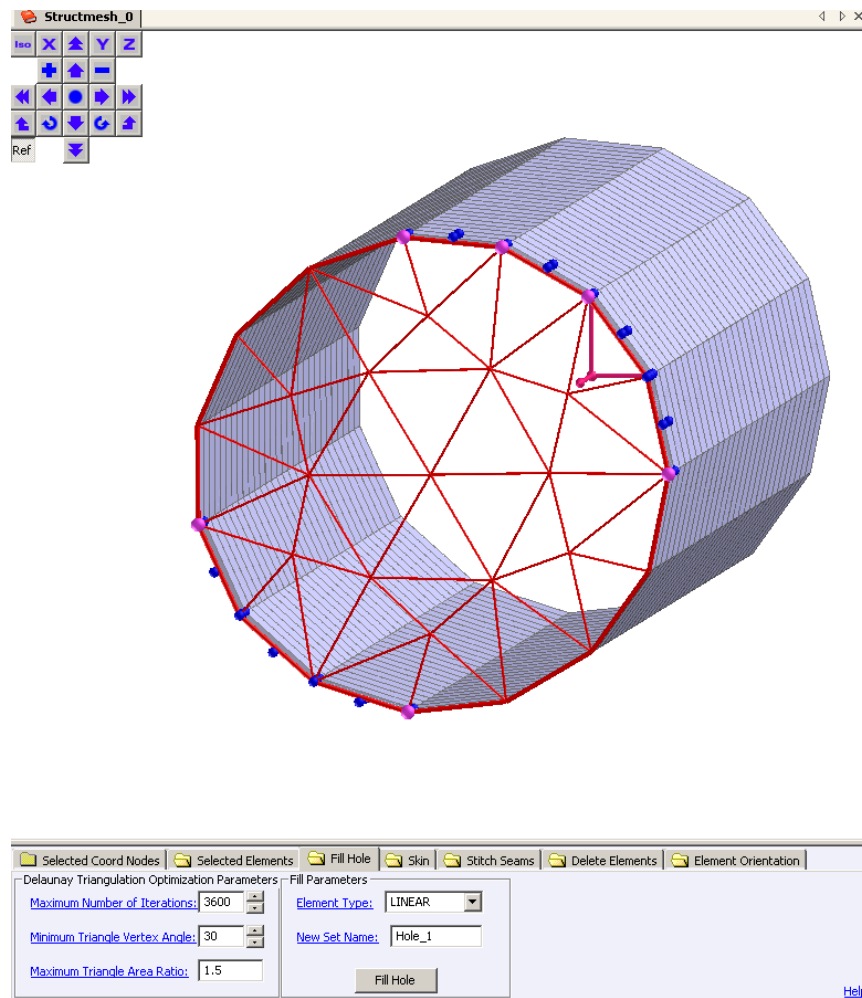


**Figure 5.21** Perform modal superposition.

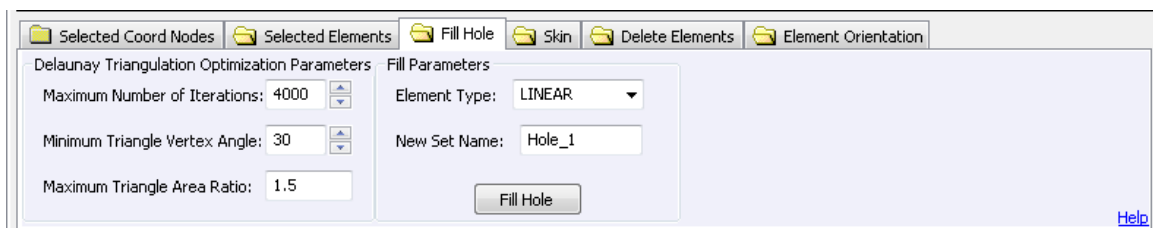


**Figure 5.22** Participating modes selection.





**Figure 5.23** Coustyx GUI - Fill Hole.



**Figure 5.24** Fill Hole Parameters.

Figure 5.24. First, we will explain each of the parameters in the Figure 5.24, and then discuss the procedure to fill a hole. (Figure 5.23)

### 5.5.1. Delaunay Triangulation Optimization 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 5.24)

**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.

### 5.5.2. Fill Parameters

**CoordConn Type** This option specifies the type of coordinate connectivity to be used for the triangle elements created to fill up the hole. The coordinate connectivity describes the interpolation scheme used to define the geometry of an element. Select **LINEAR** or **QUADRATIC** connectivity from the drop-down menu.

**VarConn Type** This option is available only for BE meshes. It specifies the type of variable connectivity to be used for the triangle elements created to fill up the hole. The variable connectivity describes the interpolation scheme used for acoustic variables on an element. Select **Same as CoordConn Type**, **CONSTANT**, **LINEAR**, or **QUADRATIC** from the drop-down menu. The option **Same as CoordConn Type** sets the variable connectivity the same as the coordinate connectivity. Note that the option **CONSTANT** is available only for *MultiDomain* models.

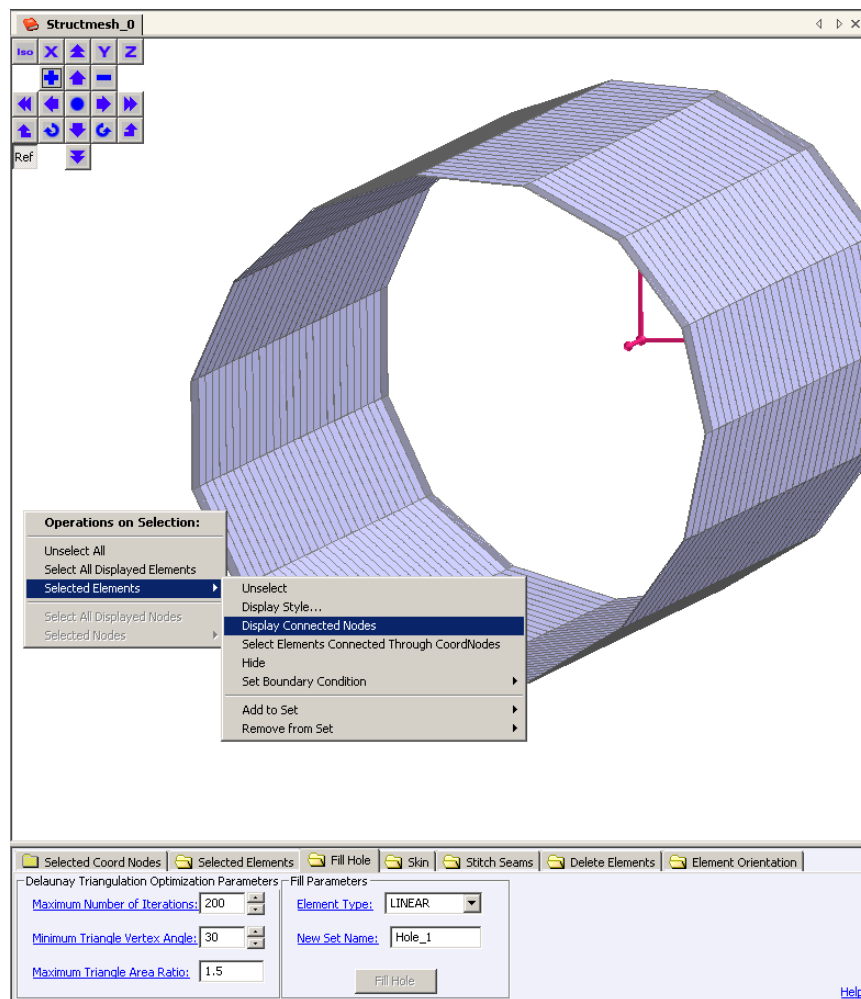
**New Set Name** All the triangle elements created to fill the hole can be conveniently added to a Set during the process. This option specifies the name of the new Set.

**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.

### 5.5.3. Procedure to Fill Hole

Follow the steps below 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 5.4.1 to set *Delaunay Triangulation Optimization Parameters* and *Fill Parameters* (Refer to Figure 5.24).
- 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.

**Figure 5.25** Display Connected Nodes

- 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 5.25)
- 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 5.23)
- 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 coordinate interpolation to be used for the triangle elements created to fill the hole: **Fill Hole** → **CoordConn Type** → **LINEAR** or **QUADRATIC**.
- Also, select the type of variable interpolation to be used: **Fill Hole** → **VarConn Type** → **Same as CoordConn Type**, **CONSTANT**, **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 5.24)

## 5.6. 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 5.27).

*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 5.6.1. 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.

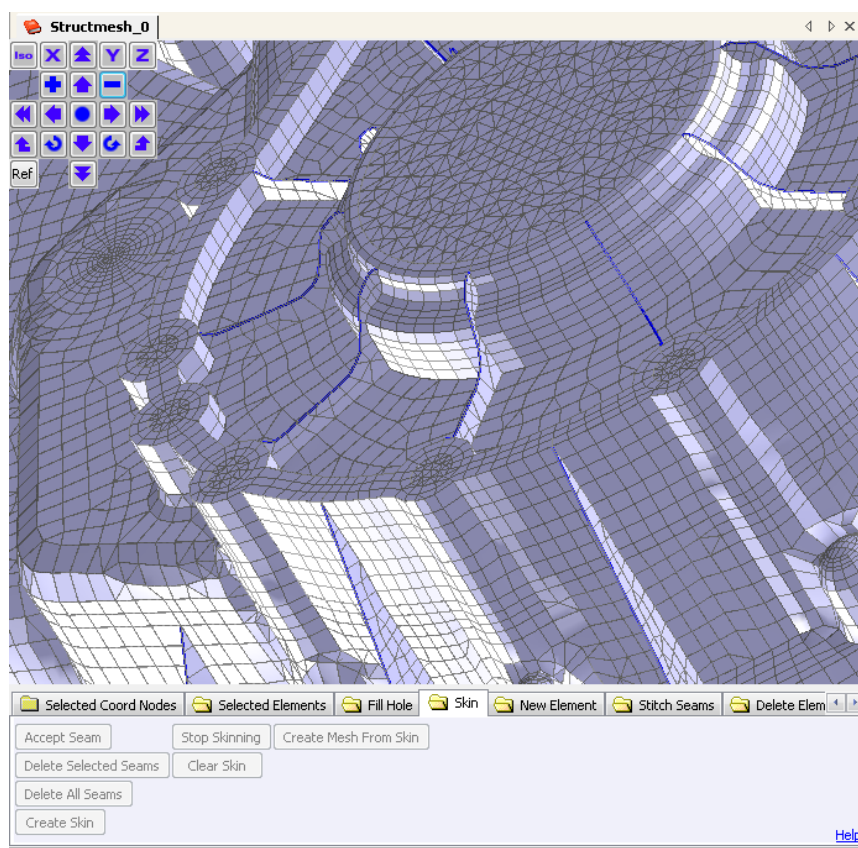
### 5.6.1. Free Edges in a FE Mesh

While in the **Skin** *tabbed* window, *Coustyx* displays all the *Free Edges* in the FE mesh in color *Blue*. (Figure 5.26)

### 5.6.2. 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 case where seams are useful is while skinning a FE mesh with symmetry planes. 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** Mesh with free edges shown in *blue* (color).

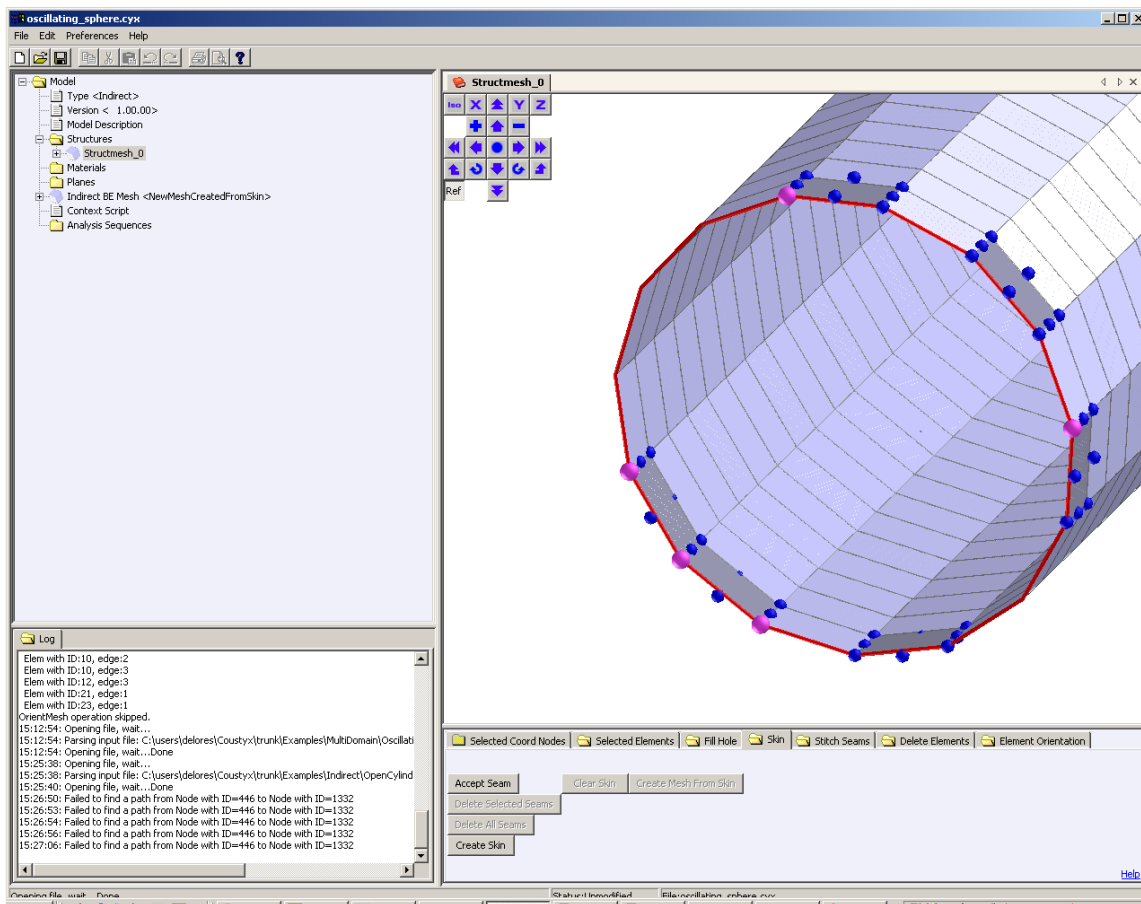
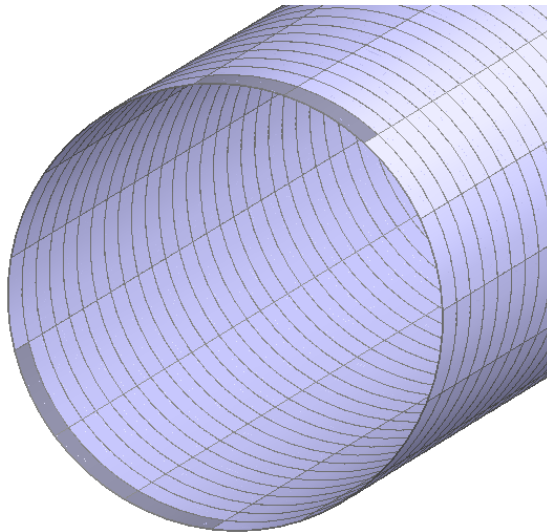


Figure 5.27 Create Seams.

**5.6.2.1. Procedure to Create Seams** Follow the steps below to create a seam:

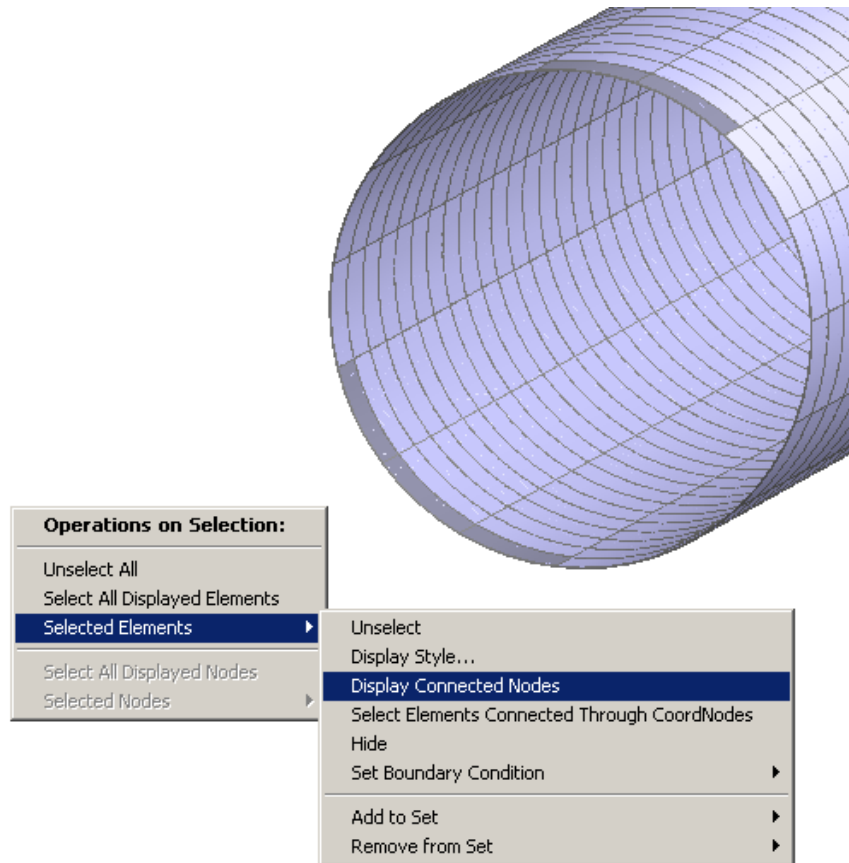
- Select a few of the elements through which the seam is supposed to pass through. To select an element in the GUI: Left-click on the element with the *shift-key* held down. (Figure 5.28)
- To display coordinate nodes of selected elements, right-click with *shift-key* held down and select: **Operations on Selection** → **Selected Elements** → **Display Connected Nodes**. (Figure 5.29)
- 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 5.30)
- Accept the seam by pressing: **Skin** → **Accept Seam**. See Figure 5.31. **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*.



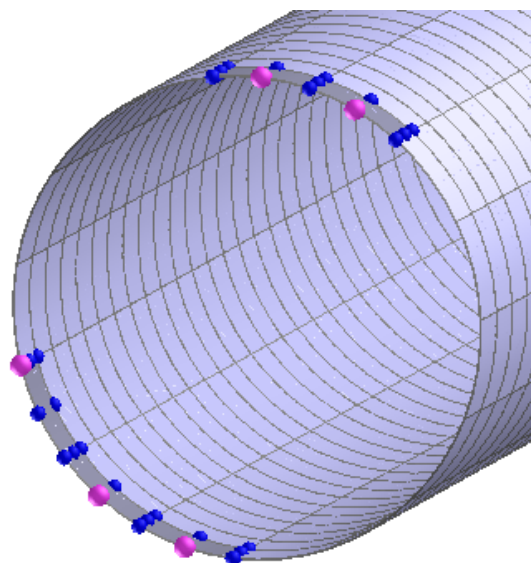
**Figure 5.28** Select Elements for Creating a Seam.

### 5.6.3. 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 5.6.1. If the seam is acceptable, click on **Skin** → **Accept Seam**. (Figure 5.27)
- Repeat the above process until all the number of necessary seams are created.



**Figure 5.29** Display Connected Nodes for Selected Elements.



**Figure 5.30** Choose Nodes for Creating Seams.



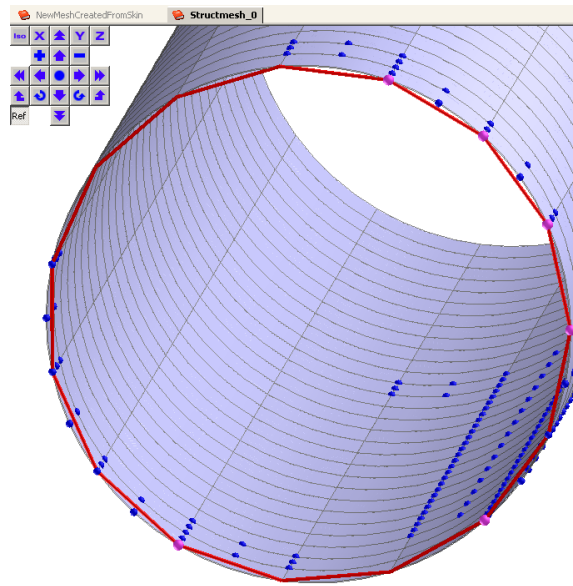


Figure 5.31 Accept Seam.

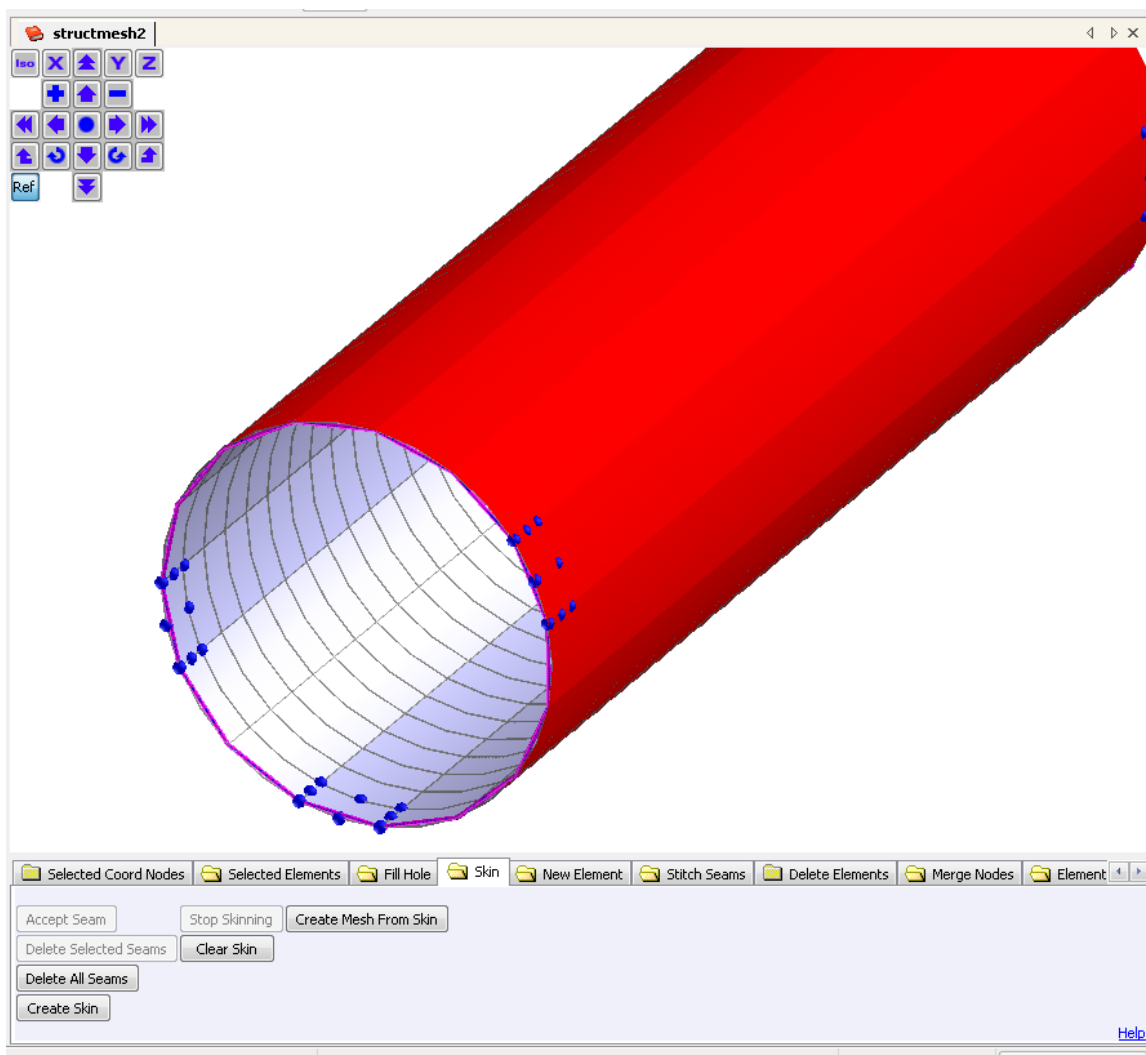
- Choose any one face of an element, preferably on the side of the acoustic domain you are interested in, to start skinning. To select an element in the GUI: Left-click on the element with the *shift-key* held down. To ensure skinning is done accurately, unselect all elements in the GUI by right-clicking 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 5.32.
- 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**.

#### 5.6.4. Create Mesh from Skin

A new boundary element mesh can be created from the generated skin (see Section 5.6.2.1) by pressing: **Skin** → **Create Mesh From Skin** button. Choose the type of variable connectivity (**VarConn Type**) desired for the BE mesh before creating the mesh.

**VarConn Type** This option specifies the type of variable connectivity to be used for all elements in the BE mesh. The variable connectivity describes the interpolation scheme used for acoustic variables on an element. Select **Same as Struct Element Type**, **CONSTANT**, **LINEAR**, or **QUADRATIC** from the drop-down menu. The option **Same as Struct Element Type** sets the variable connectivity the same as the geometry interpolation scheme (coordinate connectivity) used for the corresponding structure element. Note that the option **CONSTANT** is available only for *MultiDomain* models.

This option allows users to specify different interpolation schemes for acoustic variables and coordinates (geometry). Typically, different interpolation schemes are used when the geometry variation is quite different from the acoustic variation on the surface. For example, consider a uniformly pulsating sphere model. The sphere geometry should be modeled with *quadratic* interpolation scheme to accurately



**Figure 5.32** Create Skin.

capture the surface variation. However, we just need a *constant* interpolation scheme to capture the pressure variation on the surface, as the pressure is constant over the surface of a pulsating sphere. The flexibility in specifying independent interpolation schemes for geometry and acoustic variables can help reduce the number of degrees of freedom (which solely depend on number of acoustic variables), hence analysis run times, without effecting the solution accuracy.

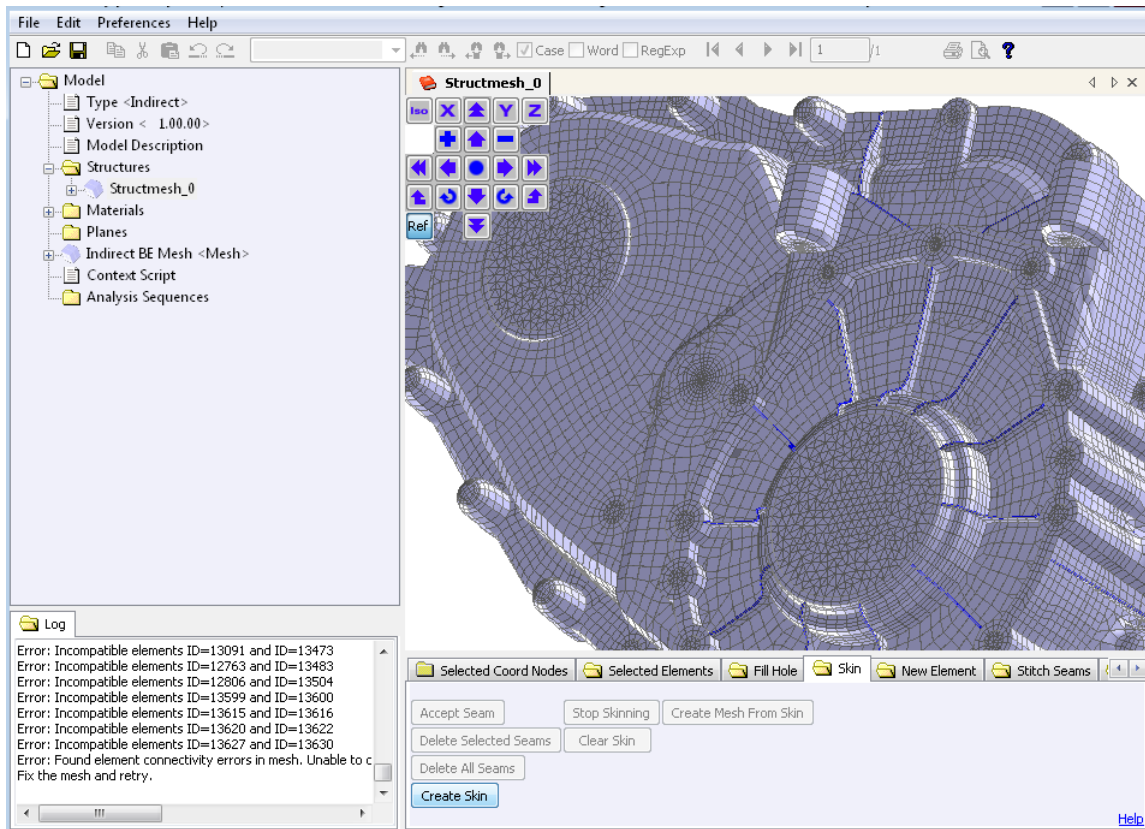


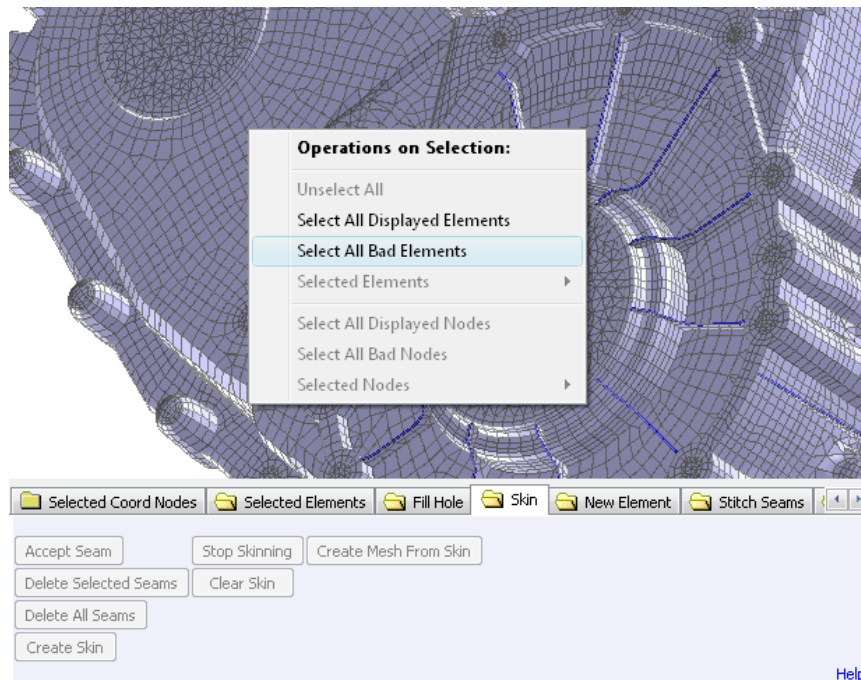
Figure 5.33 Press **Skin** → **Create Skin** to check for incompatible elements.

### 5.6.5. 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. When a 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.

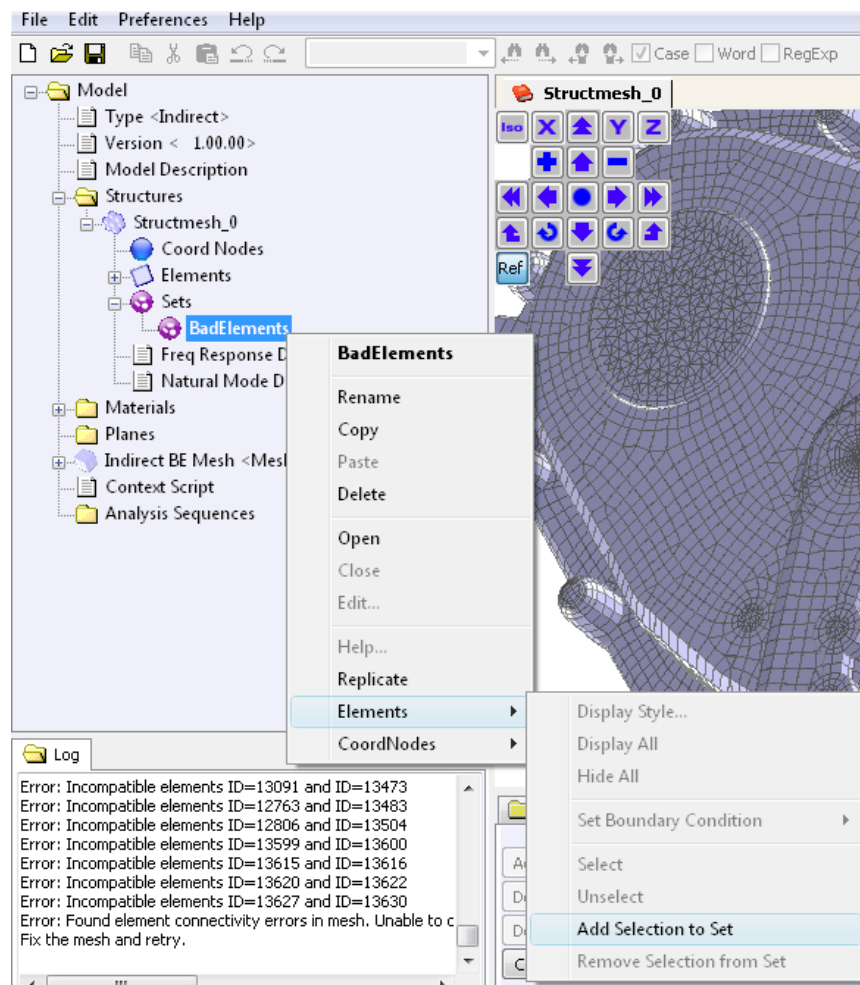
The steps to be followed to treat Bad elements in a FE mesh are:

- 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 5.33)
- To unselect all displayed elements in the GUI, right-click with *shift-key* held down and select: **Operations on Selection** → **Unselect All**.

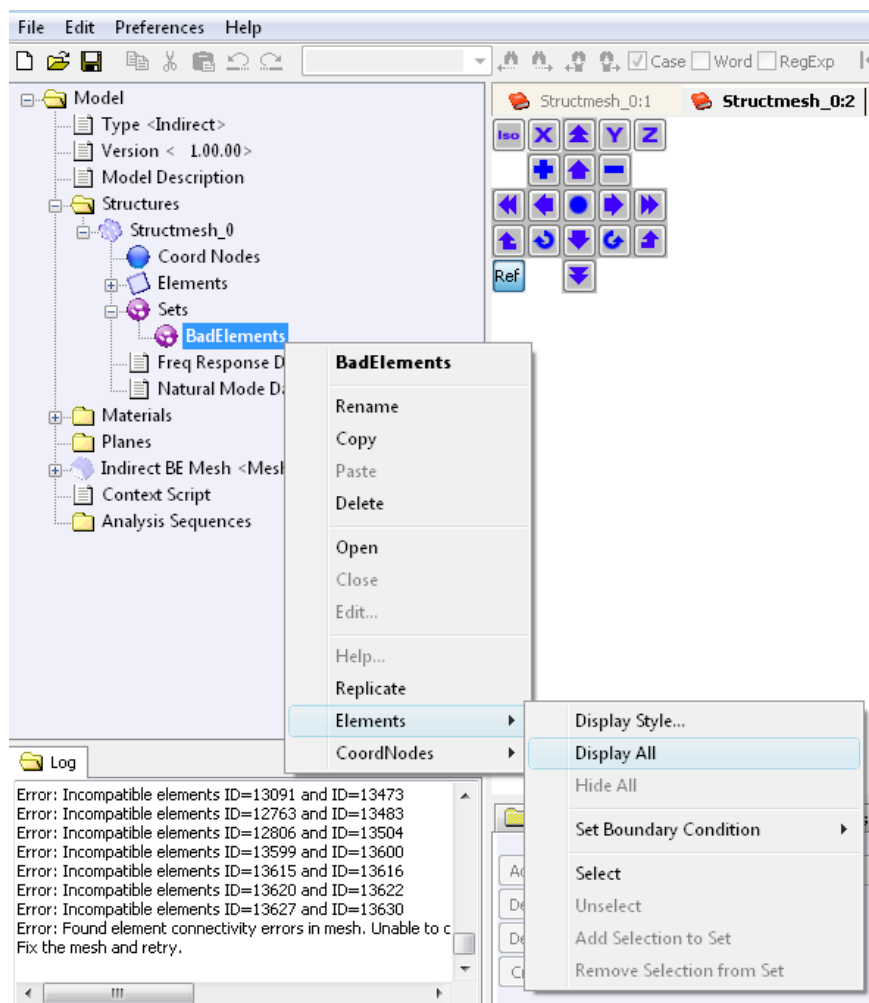


**Figure 5.34** Select all Bad Elements.

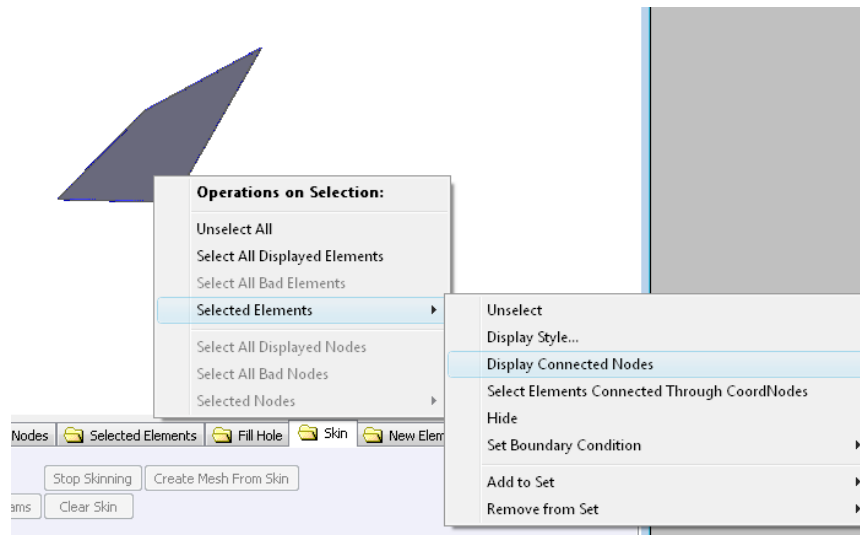
- Select all Bad Elements in the FE mesh by right-clicking on the GUI while holding down the *shift*-key and choosing **Operations on Selection** → **Select all Bad Elements**. (Figure 5.34)
- Create a new set by choosing **Model** → **Structures** → <Structure Mesh Name> → **Sets**. Right-click on **Sets** 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 5.35)
- 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 5.36)
- 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 5.37)
- 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 5.38)
- 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**.



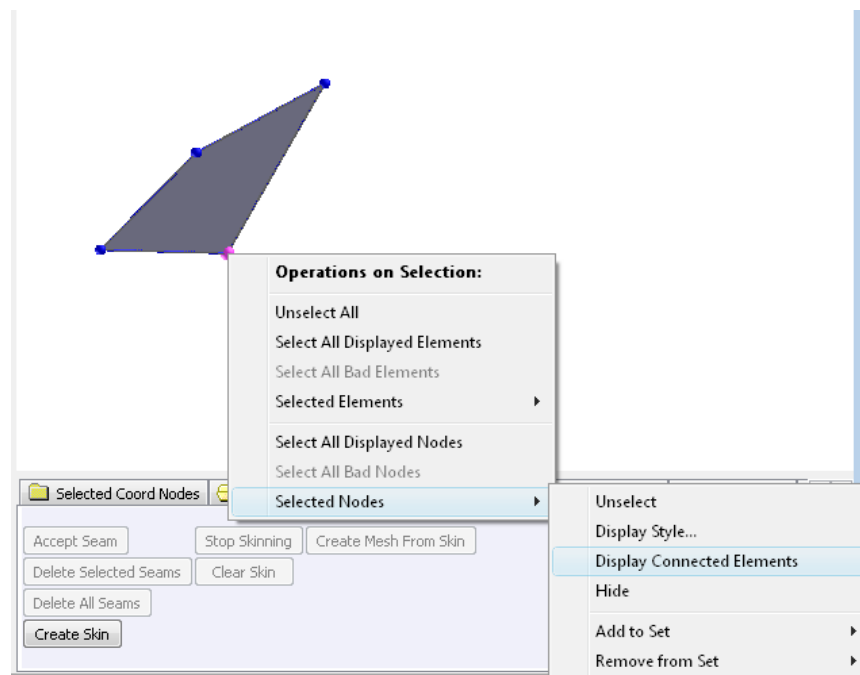
**Figure 5.35** Add Bad Elements to a Set.



**Figure 5.36** Display Set of Bad Elements.



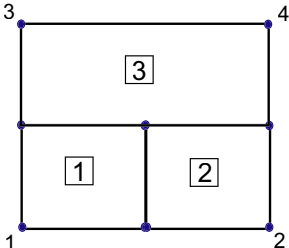
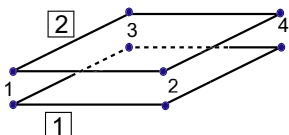
**Figure 5.37** Display Connected Nodes of Selected Element.



**Figure 5.38** Display Connected Elements of Selected Node.



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

Bad Elements	Treatment
	<ul style="list-style-type: none"> <li>▪ Incompatible coordinate connectivity between elements 3 and 1(2).</li> <li>▪ Delete elements 1, 2 and 3.</li> <li>▪ Create a new element using the coordinates 1, 2, 3 and 4.</li> </ul>
	<ul style="list-style-type: none"> <li>▪ Overlapping elements 1 and 2. These elements have the same coordinate nodes 1, 2, 3 and 4.</li> <li>▪ Delete any one of the overlapping elements.</li> </ul>

## 5.7. 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 5.39 shows the window to create a new element in a BE mesh.

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

**CoordConn Type** This option specifies the coordinate connectivity to be used for the new element. The coordinate connectivity describes the interpolation scheme used to define the geometry of the element. Select **LINEAR**, **QUADRATIC**, or **CUBIC** interpolation schemes from the drop-down menu.

**VarConn Type** This option is available only for BE meshes. It specifies the variable connectivity to be used for the new element. The variable connectivity describes the interpolation scheme used for acoustic variables on an element. Select **Same as CoordConn Type**, **CONSTANT**, **LINEAR**, **QUADRATIC**, or **CUBIC** from the drop-down menu. The option **Same as CoordConn Type** sets the variable connectivity the same as the coordinate connectivity. Note that the option **CONSTANT** is available only for *MultiDomain* models.

### 5.7.1. Procedure to Create a New Element

Follow the steps below to create a new element:

- Open any mesh in the GUI by right-clicking on the mesh and selecting **Open**.
- To select elements in the GUI: Left-click on the element with *shift-key* held down.
- To display nodes of selected elements: Right-click with *shift-key* held down and choose **Operations on Selection: Selected Elements** → **Display Connected Nodes**.



- Select the shape of the new element: **New Element** → **TRIANGLE** or **QUADRILATERAL**.
- Select the type of coordinate interpolation to be used for the new element: **New Element** → **Coord-Conn Type** → **LINEAR**, **QUADRATIC**, or **CUBIC**.
- Also, select the type of variable interpolation to be used for the new element: **New Element** → **Var-Conn Type** → **CONSTANT**, **LINEAR**, **QUADRATIC**, or **CUBIC**.
- Begin building the new element by selecting nodes from the GUI. To select a node in the GUI: left-click on it with the *shift-key* held down.
- The **Coord Node ID** table displays the IDs of chosen nodes in the sequence. Follow the sequence suggested by the node numbering in Figure 5.40, while choosing the coordinate nodes. Note that 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, an outline of the new element is shown as a preview in the GUI. Click on the **Accept** button to create the new element. To delete a node from the selection, press **Clear Node** to delete a selected node, or press **Clear All Nodes** to delete all nodes.

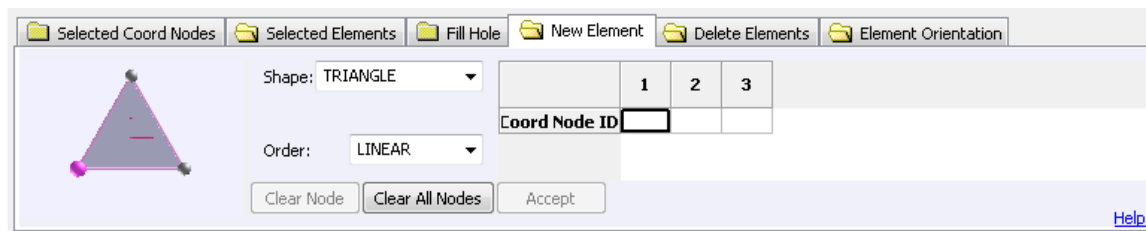


Figure 5.39 Create new element window.

## 5.8. Stitch Seams

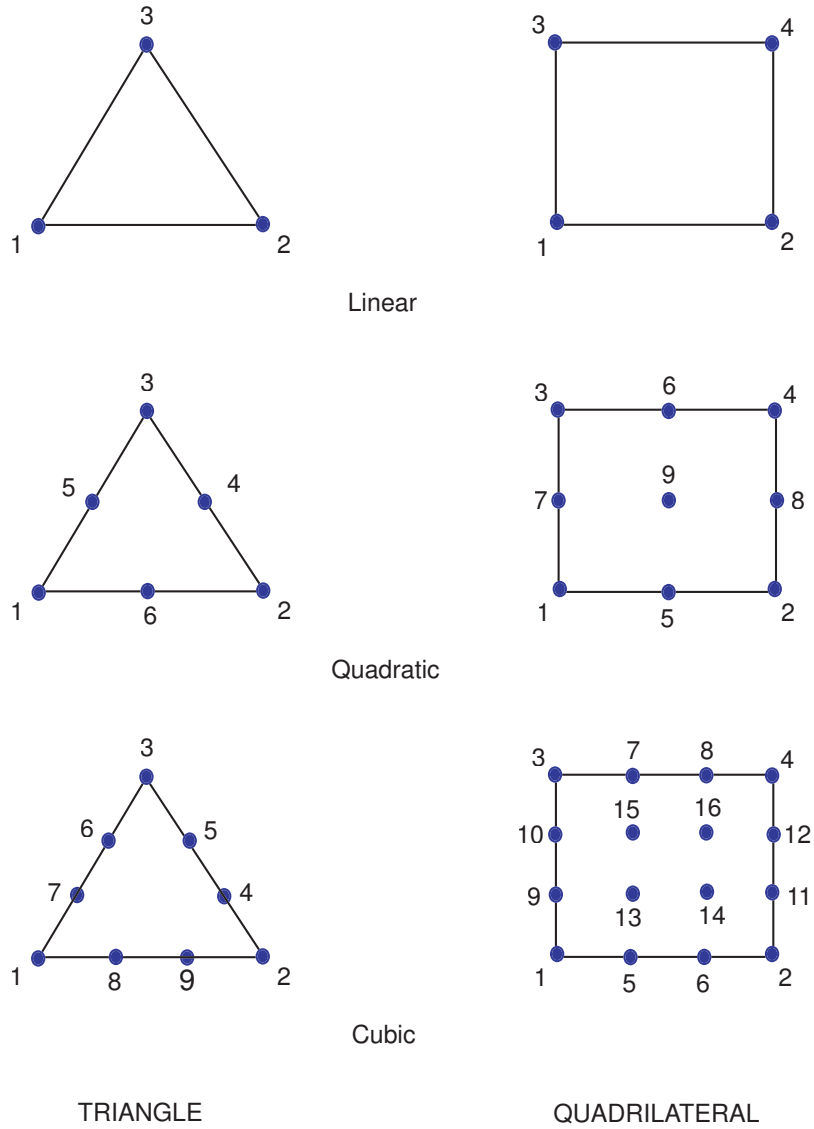
The **Stitch Seams** *tabbed* window is located at the bottom pane the *Mesh Viewer* window. This function could be used to fill gaps between disjointed parts of a structure mesh by generating new triangle elements between the seams defined. Refer to Figure 5.41.

### 5.8.1. Set Parameters

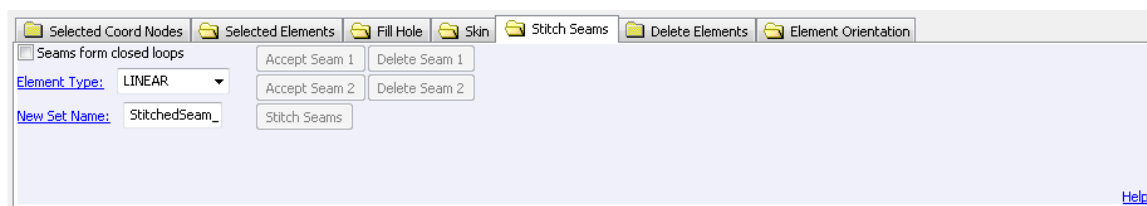
**Seams form closed loops** When this option is enabled, *Coustyx* assumes that the seam forms a closed loop and tries to find a closed path by connecting the selected coordinate nodes. If disabled, *Coustyx* assumes that the seam does not form a closed loop and just tries to connect the selected coordinate nodes to form a curve.

**CoordConn Type** This option specifies the type of coordinate connectivity to be used for the triangle elements created to fill up the gap between disjoint seams. The coordinate connectivity describes the interpolation scheme used to define the geometry of an element. Select **LINEAR** or **QUADRATIC** connectivity from the drop-down menu.

**VarConn Type** This option is available only for BE meshes. It specifies the type of variable connectivity to be used for the triangle elements created to fill up the gap. The variable connectivity describes the interpolation scheme used for acoustic variables on an element. Select **Same as CoordConn Type**,



**Figure 5.40** New elements coordinate node connectivity.



**Figure 5.41** Stitch Seam Dialog Panel

**CONSTANT**, **LINEAR**, or **QUADRATIC** from the drop-down menu. The option **Same as CoordConn Type** sets the variable connectivity the same as the coordinate connectivity. Note that the option **CONSTANT** is available only for *MultiDomain* models.

**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.

### 5.8.2. Procedure to Stitch Seams

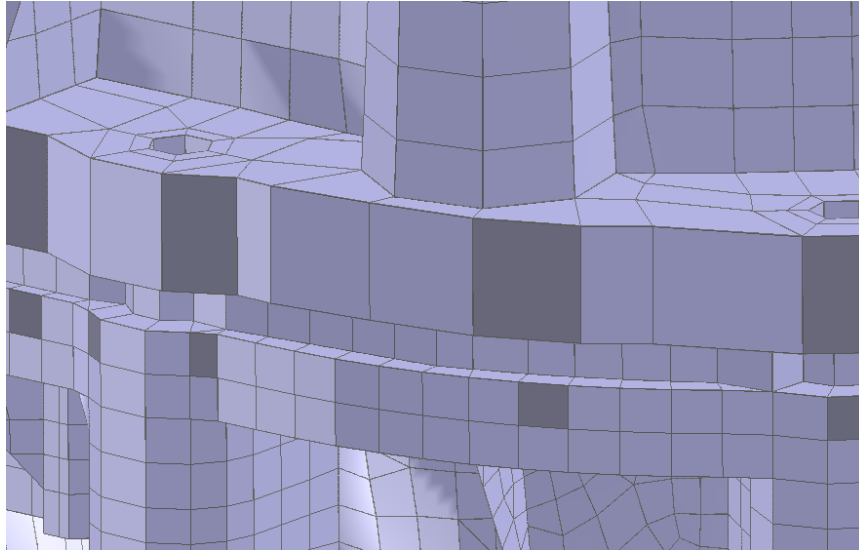
Follow the steps below to stitch a seam:

- 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 5.42)
- 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 5.43.
- 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 5.44.
- Accept the first seam by pressing: **Stitch Seams** → **Accept Seam 1**.
- Repeat steps mentioned above to create Seam 2. See Figure 5.45.
- 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 5.46.
- Select the type of coordinate interpolation to be used for the triangle elements created to fill the gap: **Fill Hole** → **CoordConn Type** → **LINEAR** or **QUADRATIC**.
- Also, select the type of variable interpolation to be used for the triangle elements created to fill the gap: **Fill Hole** → **VarConn Type** → **Same as CoordConn Type**, **CONSTANT**, **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 5.47.

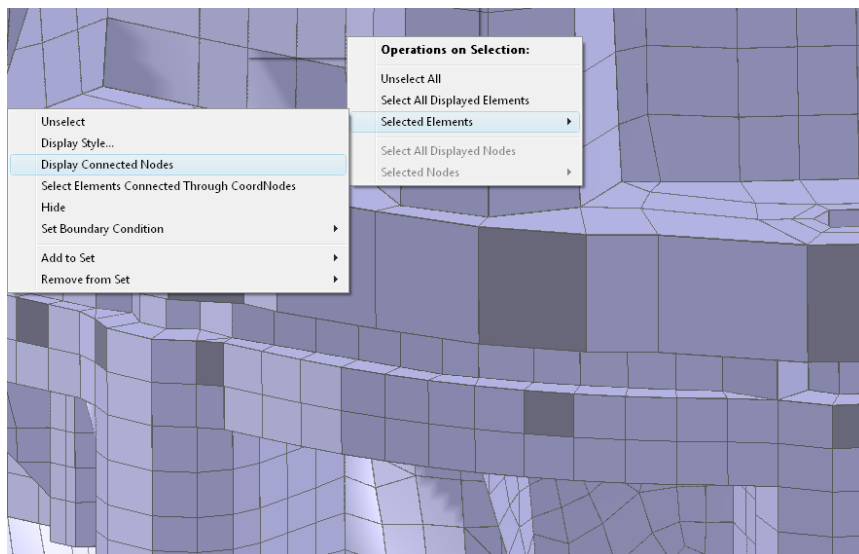
### 5.9. Delete Elements

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 5.48). *Coustyx* provides options to remove unshared coordinate nodes and unshared variable nodes of the deleted element.

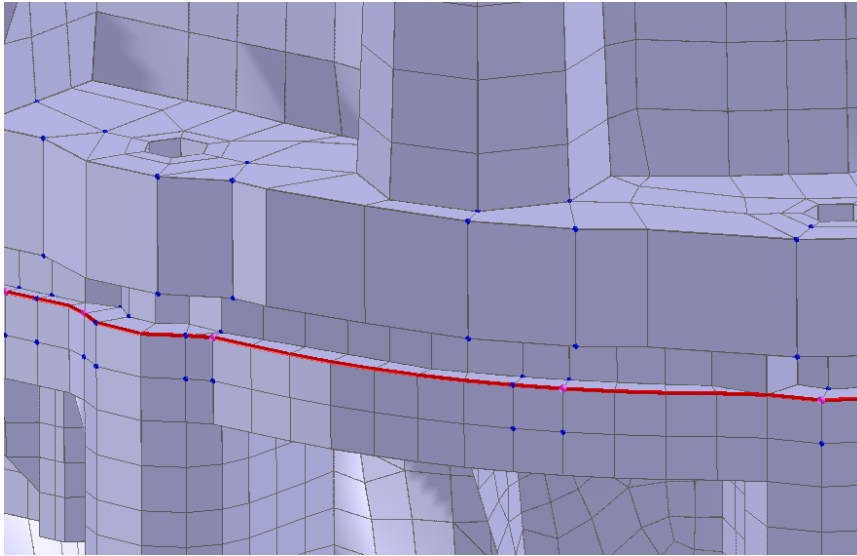
**Delete Unshared Coord Nodes** When this option is enabled, the unshared coordinate nodes of selected elements are deleted along with the elements.



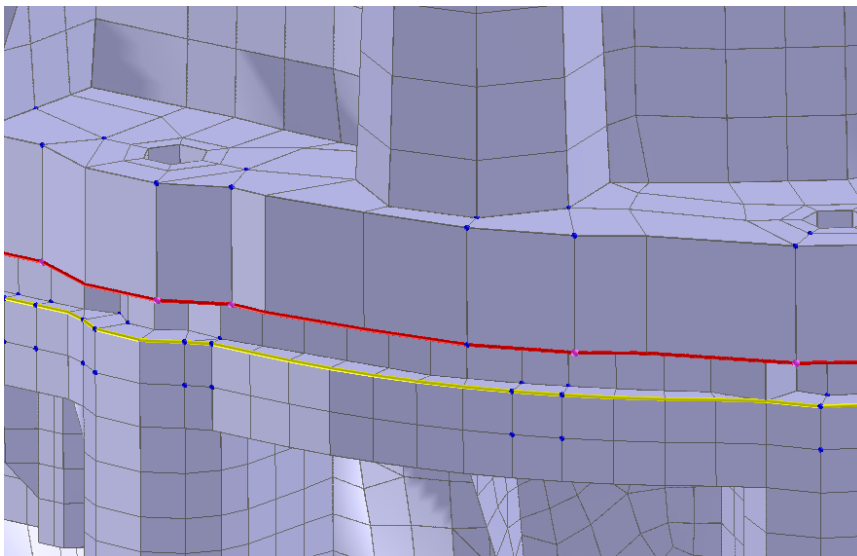
**Figure 5.42** Select Elements.



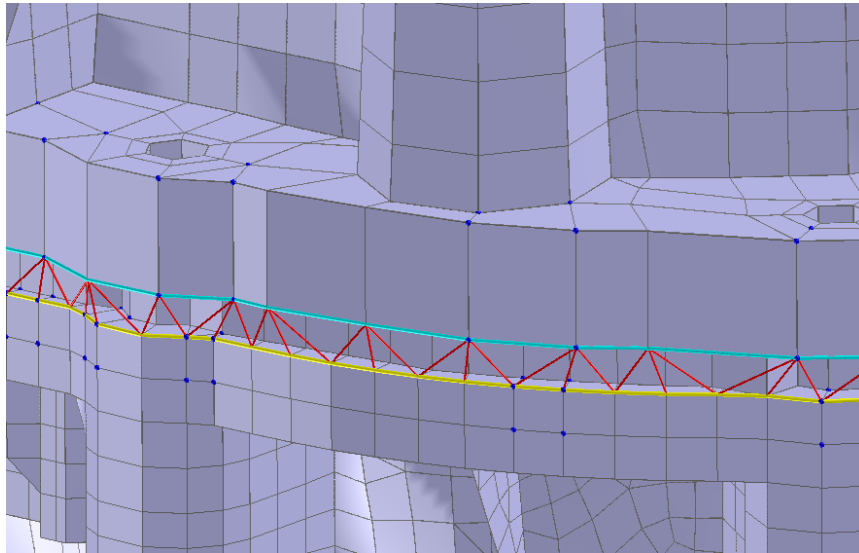
**Figure 5.43** Display Connected Nodes for Elements.



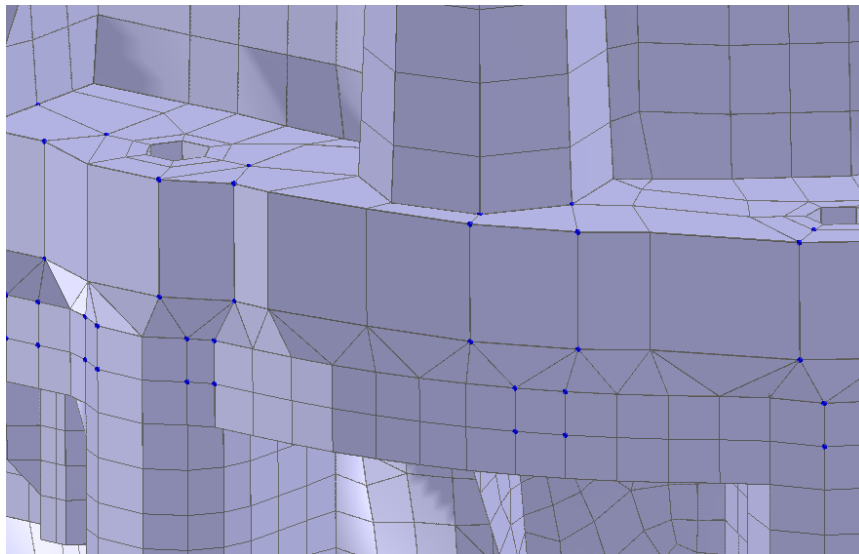
**Figure 5.44** Seam 1.



**Figure 5.45** Seam 2.



**Figure 5.46** Transparent elements shown for inspection before Stitching Seams.



**Figure 5.47** Complete Stitched Seam.

**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.

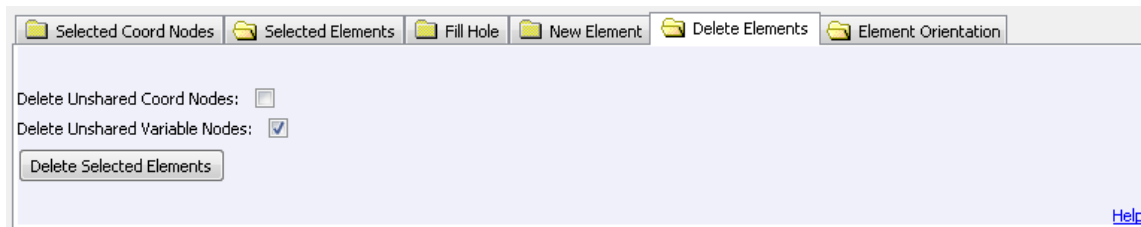


Figure 5.48 Delete element window.

## 5.10. 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 5.49). 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.

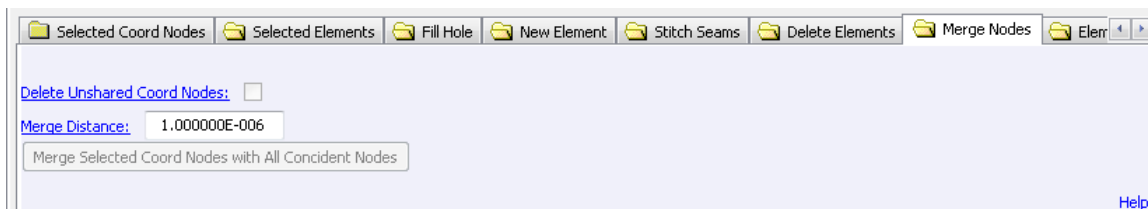


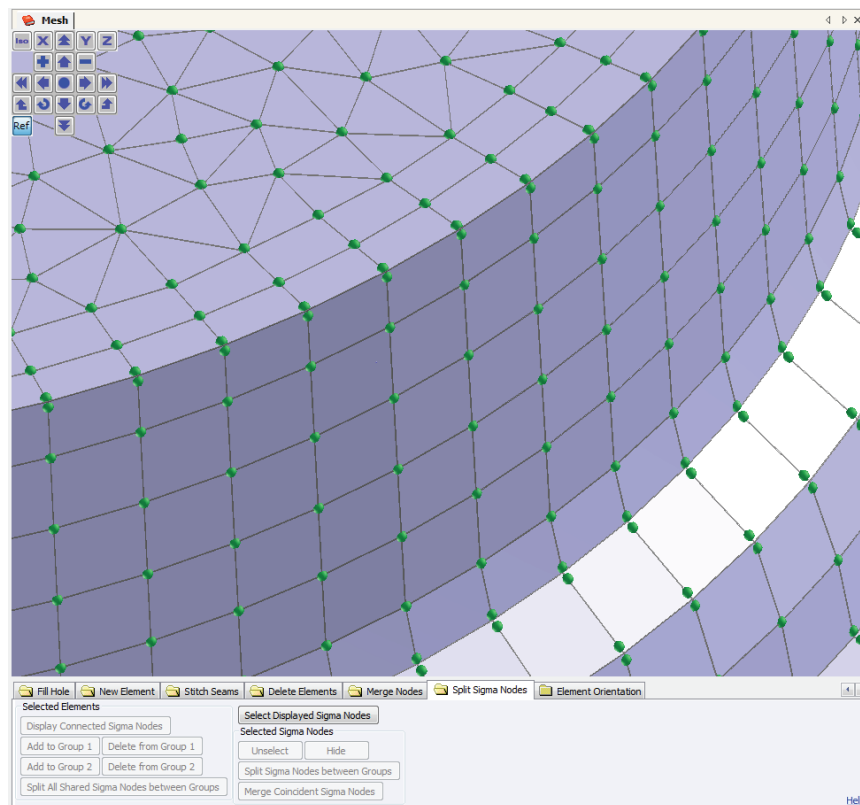
Figure 5.49 Merge Nodes window.

**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.

### 5.11. 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 5.50 shows the window for Split Pn Nodes or Split Sigma Nodes.

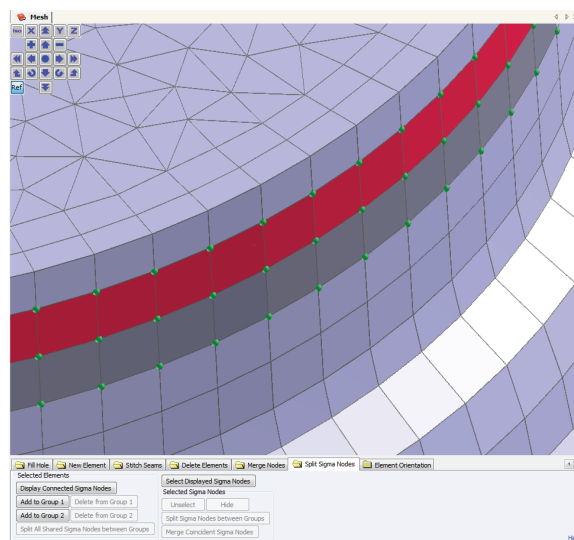


**Figure 5.50** Split Pn Nodes (Split Sigma Nodes) window showing duplicate nodes along edges. Note: During skinning, *Coustyx* automatically creates duplicate nodes for most common cases, such as geometry discontinuities, edges, corners, junctions etc.

- 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**.



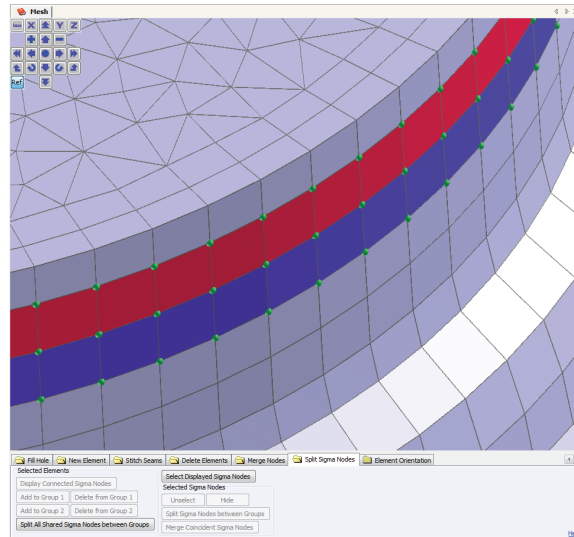
- To merge coincident nodes among the selected nodes, click on **Merge Coincident Pn Nodes (Merge Coincident Sigma Nodes)**. When a node is mistakenly split, this function would be helpful to get back to the original state.
- Left-click with *shift-key* held down to select elements and add them to Group 1 (Group 2) by clicking on **Add to Group 1 (Add to Group 2)**. See Figure 5.51 and Figure 5.52. 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 5.53.
- 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 5.54 and Figure 5.55.
- 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 5.56. 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 5.57.



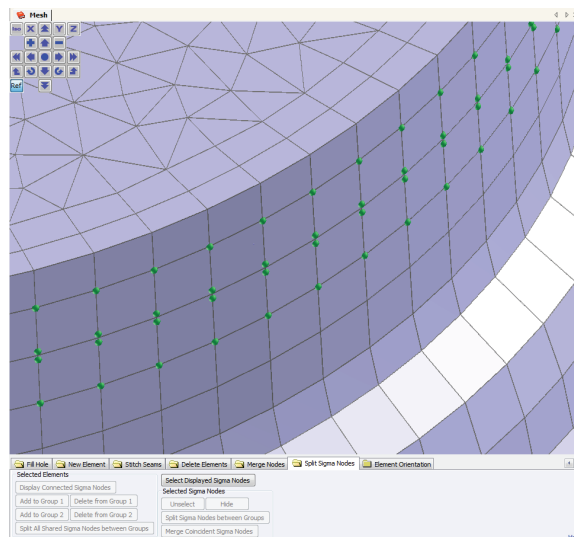
**Figure 5.51** After adding elements to Group 1. Notice that the elements belonging to Group 1 are displayed in red.

## 5.12. Element Orientation

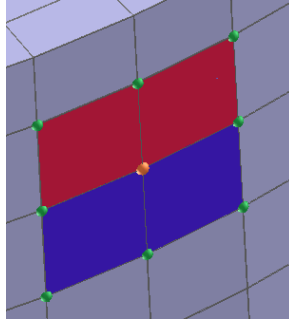
This *tabbed* window is located at the bottom pane of the *Mesh Viewer* window. It is used to view and/or flip the normals of the selected elements in the GUI (refer to Figure 5.58). Left-click on any element in the GUI



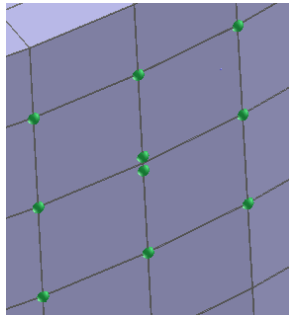
**Figure 5.52** After adding elements to Group 2. Notice that the elements belonging to Group 2 are displayed in blue.



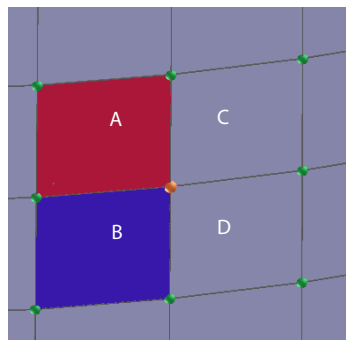
**Figure 5.53** After splitting all shared nodes between Group 1 and Group 2. Notice that the nodes along common edges of elements in Group 1 and Group 2 are split into two.



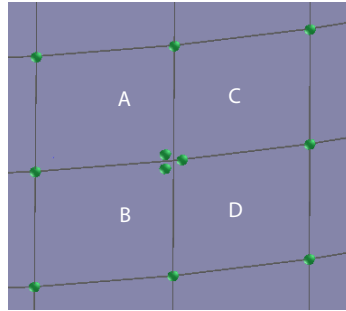
**Figure 5.54** Before splitting a selected node between Group 1 and Group 2.



**Figure 5.55** After splitting a selected node between Group 1 and Group 2.



**Figure 5.56** 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).



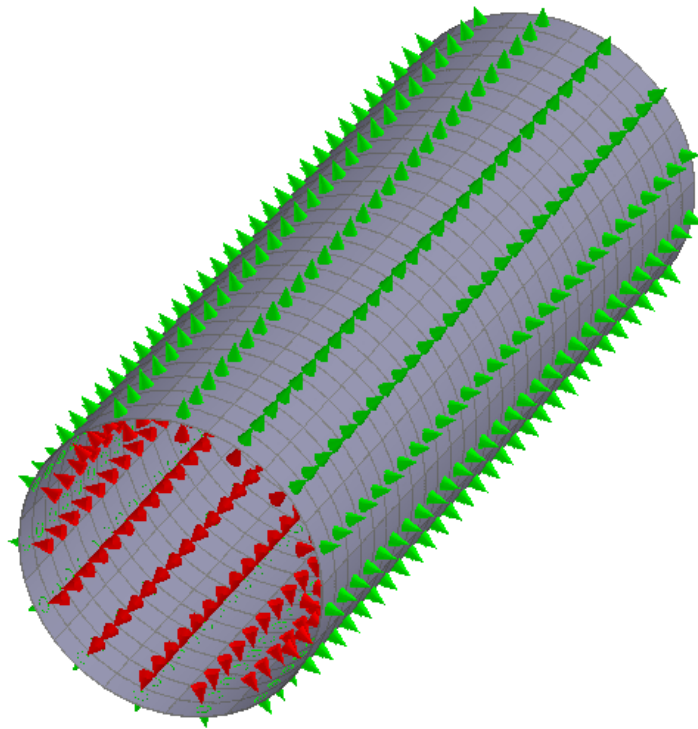
**Figure 5.57** After splitting the shared node among Group 1 (Element A), Group 2 (Element B) and unselected elements (Elements C and D).

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.

**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 5.58** Element orientations window.



## 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.

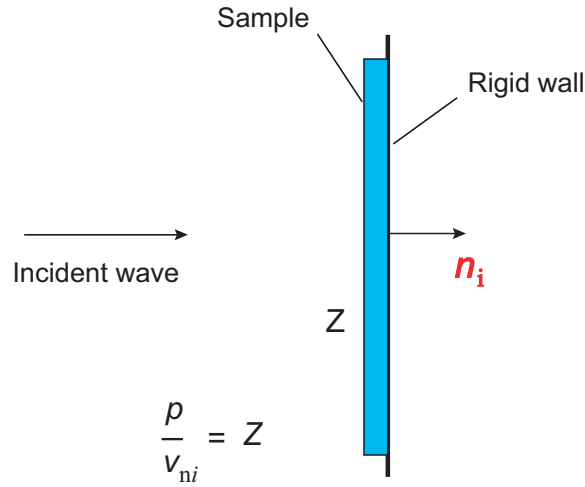
Sound pressure ( $p$ ) at the surface of the acoustic material (Figure 6.1) is related to particle normal velocity ( $v_{ni}$ ) as

$$\frac{p}{v_{ni} - v_{n\_wall}} = Z \quad (6.1)$$

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,  $v_{ni} = \mathbf{v} \cdot \mathbf{n}_i$ , 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].

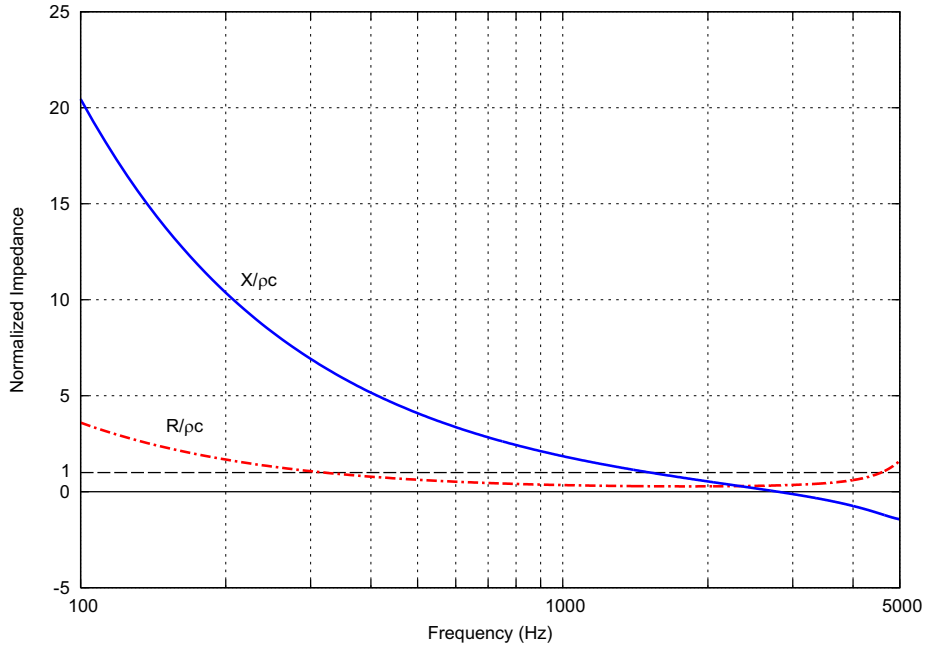
One of the popular methods used to measure impedance ( $Z$ ) of a material is the *impedance-tube* method. The values of  $Z$  are deduced from the standing wave pattern developed when excited by a plane wave in a cylindrical tube (*impedance-tube*) with acoustic material sample at one end (Pierce [1]).

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



**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.

appropriately modified before using in *Coustyx*. For example, 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** 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.

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 be 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

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

where  $\text{Re}$  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.

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.

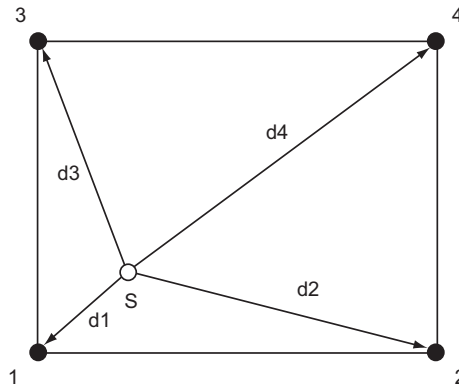


Figure 6.3 Interpolation for mismatched meshes.

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

$$v(S) = \frac{\frac{1}{d_1} v_1 + \frac{1}{d_2} v_2 + \frac{1}{d_3} v_3 + \frac{1}{d_4} v_4}{\frac{1}{d_1} + \frac{1}{d_2} + \frac{1}{d_3} + \frac{1}{d_4}} \quad (6.3)$$

Various options that control the number of interpolation points and the search criteria are discussed below. Figure 6.20.

**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, whichever 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 within 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. Structural Acoustic Coupling

Coustyx offers the ability to model two-way coupling between an elastic structure and the surrounding acoustic medium with no flow. Use this feature whenever the structural vibration could be significantly influenced by acoustic loading. The structure invacuo vibration modes from finite element analysis are combined with the boundary element formulations to solve the coupled structural acoustic problem.

Follow the steps described below to setup a coupled structural acoustic problem:

- **Set coupling type.** Select: **Model** → **Structures** → <Structure Mesh Name> → **Coupling Type**. Right-click and choose the coupling type from the two choices available: **Coupled**, or **Uncoupled**. Refer to Figure 6.4. Note: Select the structure that is going to be used to apply the structure velocity boundary condition.
- **Load natural mode data.** Load natural modes computed from the modal analysis performed using any external FE packages. Note: Make sure the modes are ortho-normalized with respect to the finite element mass matrix.
- **Define & apply structure velocity boundary condition.** Define a structure velocity boundary condition and apply it on the boundary element mesh. Refer to Section 6.5.7.1 and Section 6.6.3.5 for more details on the structure velocity boundary condition type. Note: The structure velocity boundary condition should use the above structure.
- **Run analysis.** Setup the model and run the analysis. Coustyx solves for both the structure velocities and surface pressures (or pressure jumps).

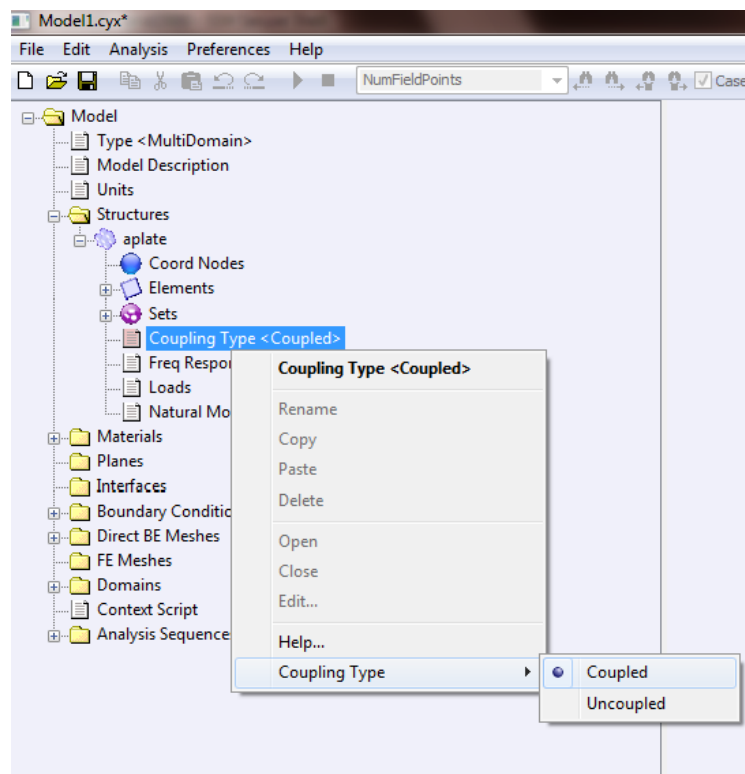


Figure 6.4 Coupling type.

## 6.4. Interfaces

An acoustic interface is used to connect two different fluid domains in a *MultiDomain* model. It relates acoustic variables at coincident points on either side of the interface through transfer matrices.

Follow the steps described below to use interfaces in a *Multidomain* model:

- **Define Interface.** **Model** → **Interfaces**. Define an *interface* with the desired transfer relation between the acoustic pressures and velocities on two sides of the interface surface. Allowed interface types include: [Transparent](#), [Uniform Perforated](#), [Non-uniform Perforated](#), [Uniform Arbitrary](#), and [Non-uniform Arbitrary](#).
- **Define Interface BCs.** **Model** → **Boundary Conditions**. Create two [interface boundary conditions](#) that are linked to the same interface defined above. For one of the *interface BCs*, specify that the domain is on the positive side of the interface; let's call it the *positive side* ('+' side) of the interface; for the other, specify that the domain is on the negative side; let's call it the *negative side* ('-' side) of the interface. Note: Always, an *interface* should be associated with a positive and a negative side.
- **Apply Interface BCs.** Separately apply each of the above *interface BCs* to elements on the interface from the two domains. Thus, one domain would be on the *positive side* ('+' side) of the interface, and the other would be on the *negative side* ('-' side)).

In summary the interface boundary conditions relate the acoustic variables on the interface in two domains through the transfer relation specified in this section. Note: the normal vector that is used in implementing the interface BCs or *interface normal* always points from the *negative side* to the *positive side*.

To create a new Interface, right-click on **Interfaces** and select **New** (Figure 6.5). Below is the description of each of the interface options available for *MultiDomain* models.

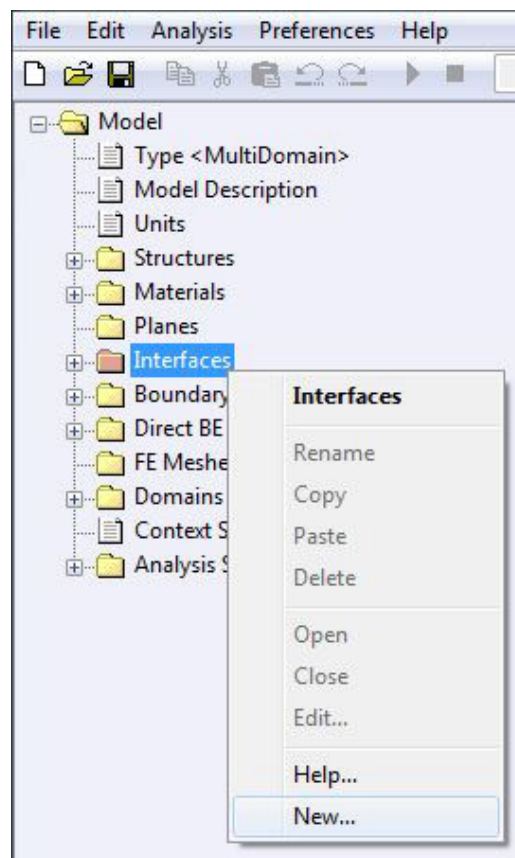


Figure 6.5 New Interface.

### 6.4.1. Transparent Interface

Transparent interface can be applied on a boundary between two domains across which the pressure and normal velocity is continuous (Figure 6.6).

$$p^+ = p^- \quad (6.4)$$

$$v_n^+ = v_n^- \quad (6.5)$$

Note that the transfer relations are presented with respect to the *interface normal*. An *interface normal* is a vector pointing from the *negative side* of the interface to the *positive side*.



Figure 6.6 Transparent Interface.

### 6.4.2. Perforated Interface

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

$$p^+ - p^- = -\rho_o c_o \zeta [v_n^-] \quad (6.6)$$

$$v_n^+ = v_n^- \quad (6.7)$$

where  $\zeta$  is the non-dimensional transfer impedance of the perforated surface,  $\rho_o$  is the density and  $c_o$  is the speed of sound in the acoustic domain on the *negative* side of the interface,  $p^+$  and  $p^-$  are surface pressures on *positive* and *negative* sides of the interface respectively, and  $v_n^-$  is the acoustic normal velocity on negative side of the interface. Note that the transfer relations are presented with respect to the *interface normal*. An *interface normal* is a vector pointing from the *negative side* of the interface to the *positive side*.

Use **Perforated Model Type** drop-down menu to choose different modelling choices for a perforated interface.

**6.4.2.1. Sullivan and Crocker** The non-dimensional transfer impedance  $\zeta$  used in the Sullivan and Crocker model ([2]) is,

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

where  $k_o$  is the wave number in air,  $t_w$  is the plate thickness,  $d_h$  is the hole diameter, and  $\chi$  is the porosity of the plate (Note: the actual relation from [2] is modified to accomodate  $e^{-j\omega t}$  convention used in *Coustyx*). Note that the above relation would be valid only for air-air contact and when the porosity is not too different from 0.042 (4.2%) used by Sullivan and Crocker in their experiments.

**Porosity** Define porosity  $\chi$  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.

**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.

**6.4.2.2. Transfer Impedance** Coustyx allows one to define their own transfer impedance models obtained from analytical expressions or empirical models derived from experiments. Note that the transfer impedance  $\zeta$  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).

**6.4.2.3. Uniform Perforated Interface** This interface type is applied over perforated elements with position-independent parameters. Refer to Section 6 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 6.7)

**6.4.2.4. Non-uniform Perforated Interface** This interface type is applied over perforated elements with position-dependent parameters. Refer to Section 6 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. (Figure 6.8.)

### 6.4.3. Uniform Arbitrary Interface

This option could be used to define uniform transfer relation that don't fall in any of the other interface types. This is the most general type of interface type possible. The user can specify the relation between the pressure and normal velocity on *positive* side of the interface to the *negative* side by the following transfer function. (Figure 6.9)

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} p^+ \\ p^- \end{bmatrix} + \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix} \begin{bmatrix} v_n^+ \\ v_n^- \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \quad (6.9)$$

where  $p^+$ ,  $v_n^+$ , and  $p^-$ ,  $v_n^-$  are the pressure and normal velocity on positive and negative sides of the interface respectively. Note that the transfer relations are presented with respect to the *interface normal*. An *interface normal* is a vector pointing from the *negative side* of the interface to the *positive side*.

The variables  $\alpha_{11}$ ,  $\alpha_{12}$ ,  $\alpha_{21}$ ,  $\alpha_{22}$ ,  $\beta_{11}$ ,  $\beta_{12}$ ,  $\beta_{21}$ ,  $\beta_{22}$ ,  $\gamma_1$ ,  $\gamma_2$  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.

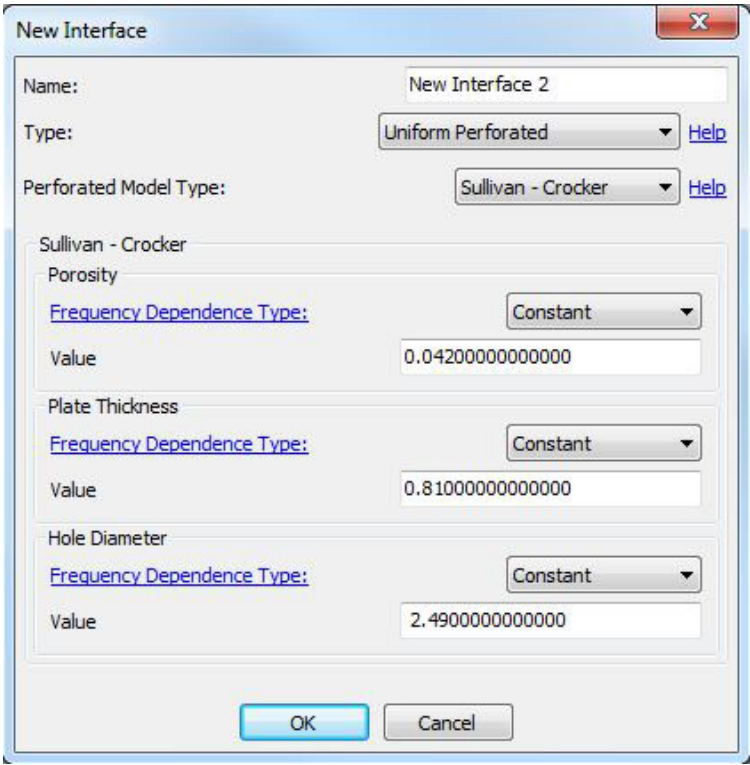


Figure 6.7 Uniform Perforated Interface

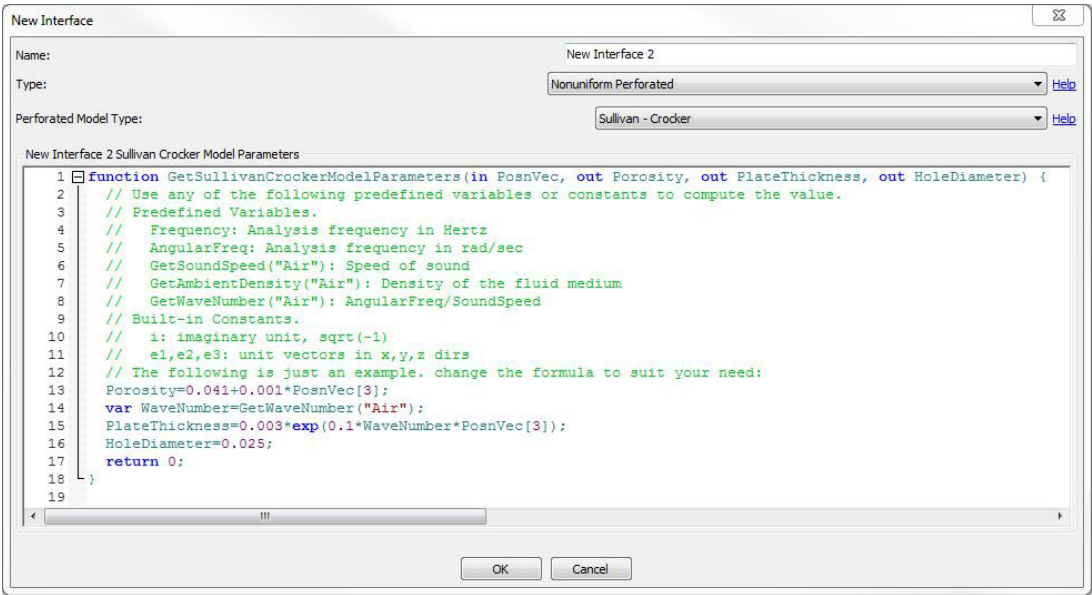


Figure 6.8 Non-Uniform Perforated Interface



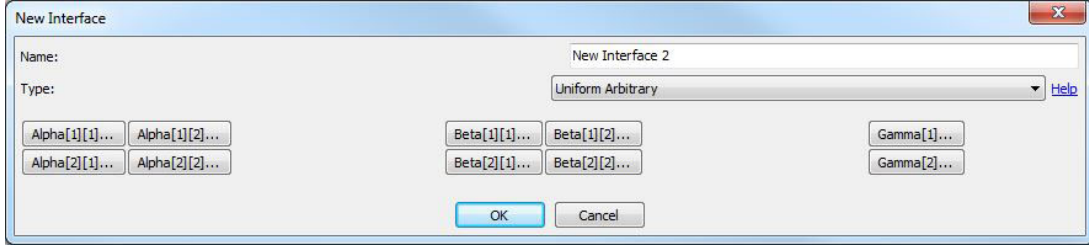


Figure 6.9 Uniform Arbitrary Interface

All the entries in the coefficient matrices of Equation 6.9 are not independent. They should satisfy the following relations for *Coustyx* to apply the arbitrary BC correctly.

$$\begin{aligned}
 (\alpha_{11}\alpha_{22} - \alpha_{12}\alpha_{21}) &\neq 0 \\
 (\beta_{11}\beta_{22} - \beta_{12}\beta_{21}) &\neq 0 \\
 (\alpha_{21} + \alpha_{22})(\beta_{11} + \beta_{12}) - (\alpha_{11} + \alpha_{12})(\beta_{21} + \beta_{22}) &\neq 0 \\
 \alpha_{11}\beta_{21} - \alpha_{21}\beta_{11} - \alpha_{12}\beta_{22} + \alpha_{22}\beta_{12} &= 0
 \end{aligned} \tag{6.10}$$

#### 6.4.4. Non-uniform Arbitrary Interface

This option could be used to define non-uniform transfer relation that do not fall into any of the other interface types. The general transfer relation between pressures and normal velocities on either side of the interface at any point is given by Equation 6.9.

The variables  $\alpha_{11}$ ,  $\alpha_{12}$ ,  $\alpha_{21}$ ,  $\alpha_{22}$ ,  $\beta_{11}$ ,  $\beta_{12}$ ,  $\beta_{21}$ ,  $\beta_{22}$ ,  $\gamma_1$ ,  $\gamma_2$  can be defined by the function call *GetAlphaBetaGamma* (Figure 6.10). The input arguments position vector *PosnVec*, and normal vector *NormalVec*, along with other predefined variables such as *AngularFreq* ( $\omega$ ), 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.

$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix}, \quad \gamma = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \tag{6.11}$$

### 6.5. 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 6.11), in order to assign the boundary velocities correctly.

- **Domain Normal** is defined to be always pointing away from the domain of interest. Figure 6.11 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



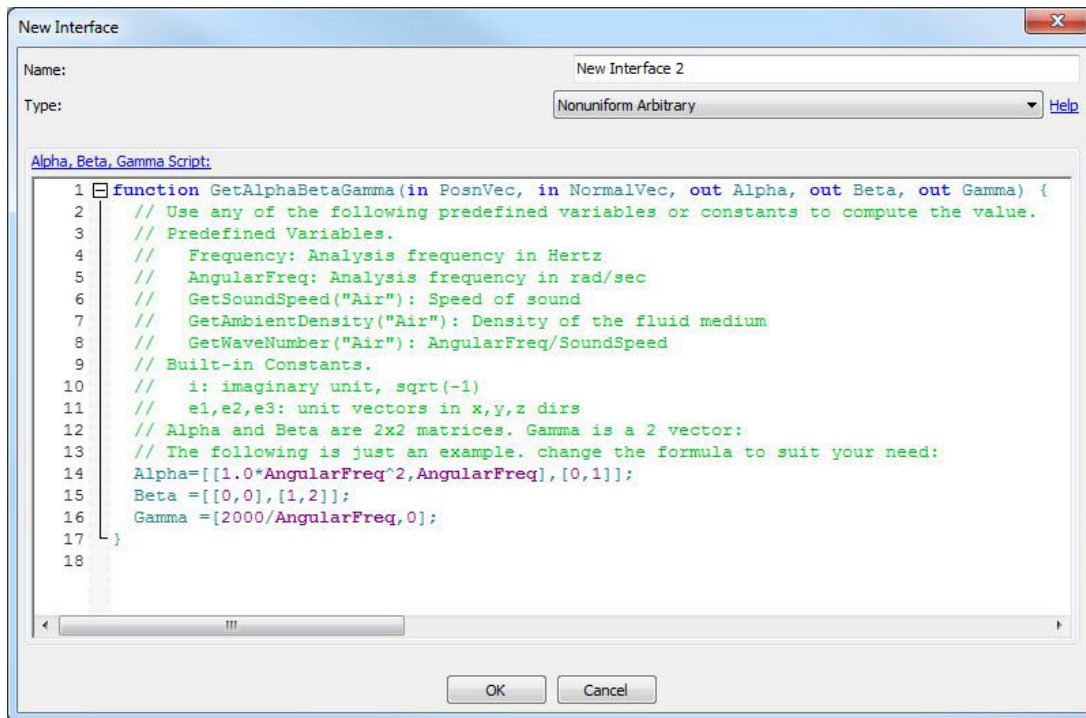


Figure 6.10 Sample script for the function GetAlphaBetaGamma().

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 6.11, the element normal is arbitrarily assumed to be pointing inward. The direction of mesh normal is irrelevant while defining boundary condition.

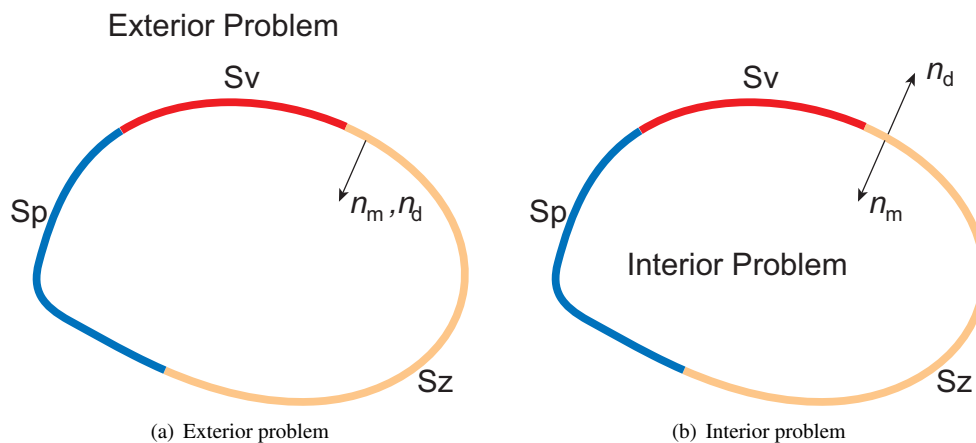
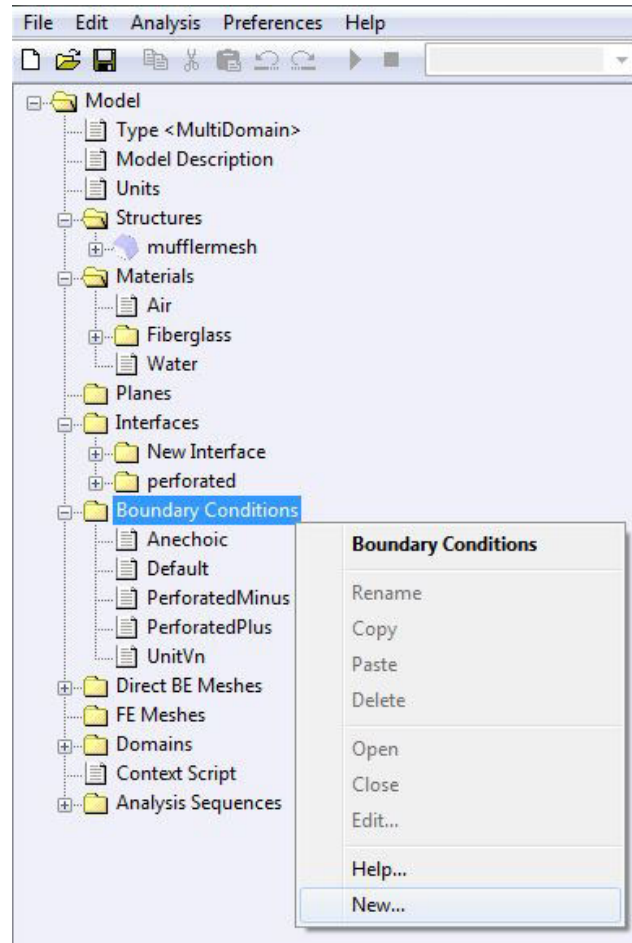


Figure 6.11 Definition of domain ( $n_d$ ) and mesh normals ( $n_m$ ). Note that domain normal always points away from the domain of interest. Mesh normal, however, can point towards or away from the fluid. All boundary conditions in a *MultiDomain* model are defined with respect to the domain normal.

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 6.12). Below is the description of each of the boundary condition options provided in *MultiDomain* model.



**Figure 6.12** New Boundary Condition.

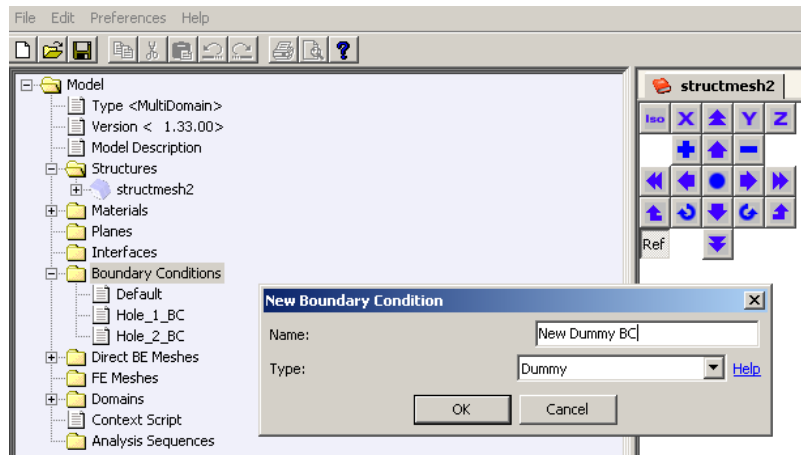
### 6.5.1. Dummy BC

This Boundary Condition is applied on elements which have no interaction with the fluid medium. They don't contribute to the acoustic radiation. (Figure 6.13.) For example, *Dummy BC* could be used on elements lying on a ground plane, as these elements don't interact with the fluid medium.

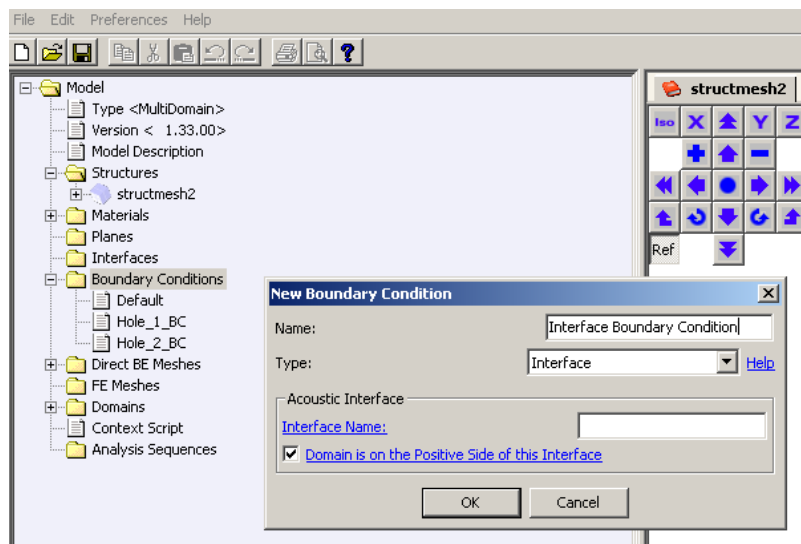
### 6.5.2. Interface BC

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 6.14.)

**6.5.2.1. Interface Name** Defines the name of the acoustic interface. (Figure 6.14.)



**Figure 6.13** Dummy Boundary Condition.



**Figure 6.14** Interface Boundary Condition.

**6.5.2.2. 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 6.14.)

### 6.5.3. 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 be dependent on the frequency. The pressure values can be specified by selecting any of the frequency dependence types: *Constant*, *Table*, or *Script* (Figure 6.15.)

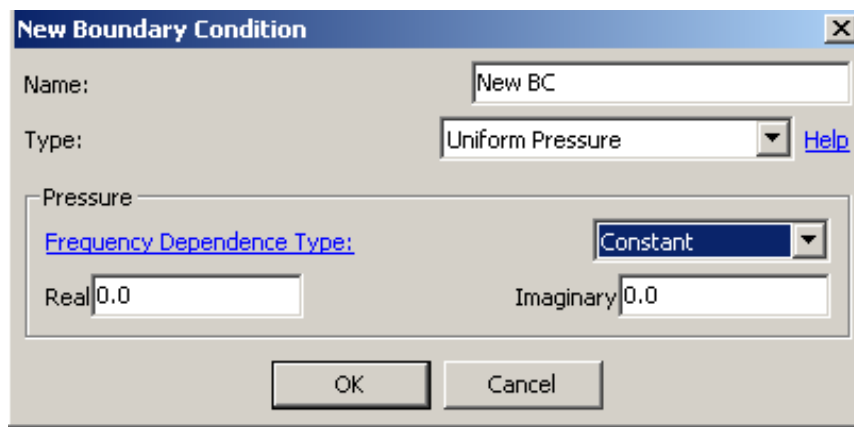


Figure 6.15 Uniform Pressure BC.

### 6.5.4. 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/functions that can be used in the script are *AngularFreq* ( $\omega$ ) – frequency in radians/sec, *GetSoundSpeed*("MaterialName") ( $c$ ) – speed of sound in the medium with the same units as those defined in materials, *GetWaveNumber*("MaterialName") ( $k = \frac{\omega}{c}$ ), and *GetAmbient-Density*("MaterialName") ( $\rho$ ) – density of the medium with the same units as those defined in materials. (Figure 6.16.)

### 6.5.5. 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 6.17.) 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 6.11). 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 6.17)

**6.5.5.1. 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 for the definition

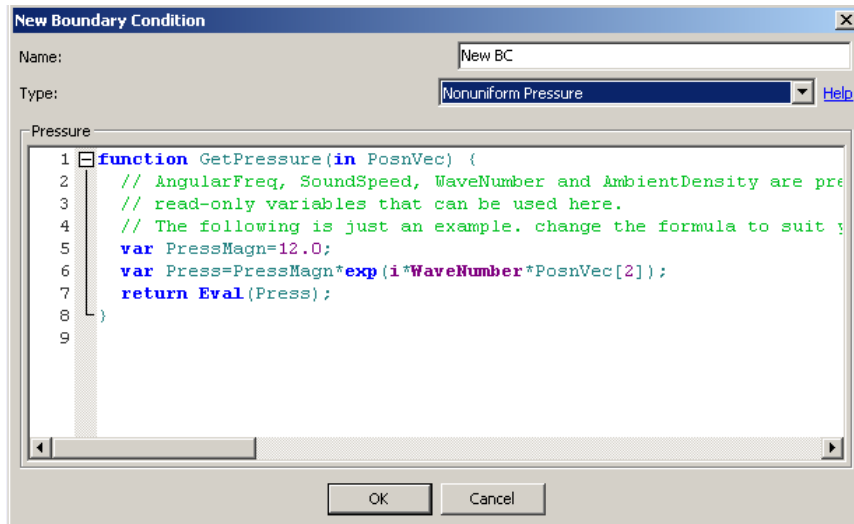


Figure 6.16 Non-Uniform Pressure BC.

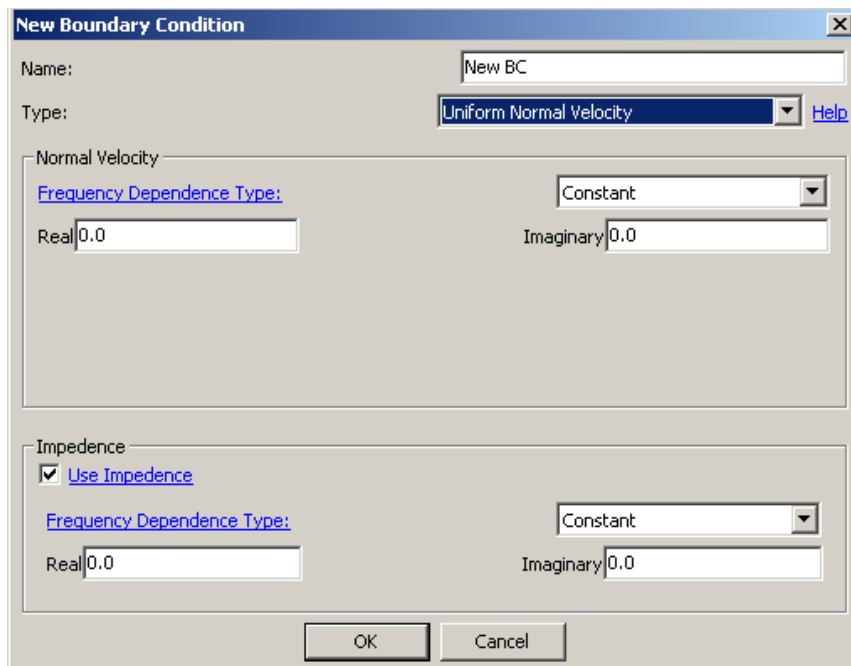


Figure 6.17 Uniform Normal Velocity BC.

of impedance implemented in *Coustyx*. The impedance value is defined by selecting any of the frequency dependent types: *Constant*, *Table*, or *Script*.

### 6.5.6. 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 6.18.)

The screenshot shows the 'New Boundary Condition' dialog box. The 'Name' field is 'New BC'. The 'Type' dropdown is set to 'Uniform Velocity'. The 'Velocity' section has a 'Frequency Dependence Type' dropdown set to 'Constant'. Below this, there are input fields for Real(X), Real(Y), Real(Z), Imaginary(X), Imaginary(Y), and Imaginary(Z), all of which are currently set to 0.0. The 'Impedance' section has a checked 'Use Impedance' checkbox, a 'Frequency Dependence Type' dropdown set to 'Constant', and input fields for Real and Imaginary impedance, both set to 0.0. At the bottom are 'OK' and 'Cancel' buttons.

**Figure 6.18** Uniform Velocity BC.

**6.5.6.1. 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 6.11). 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 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*.

### 6.5.7. 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 6.11) 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 6.19.)

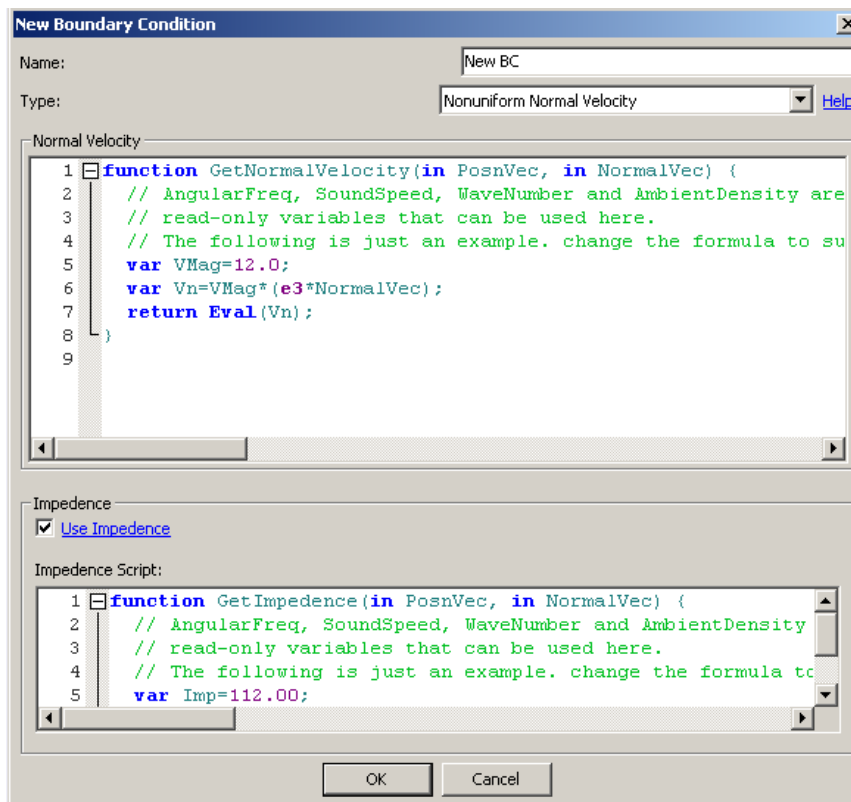


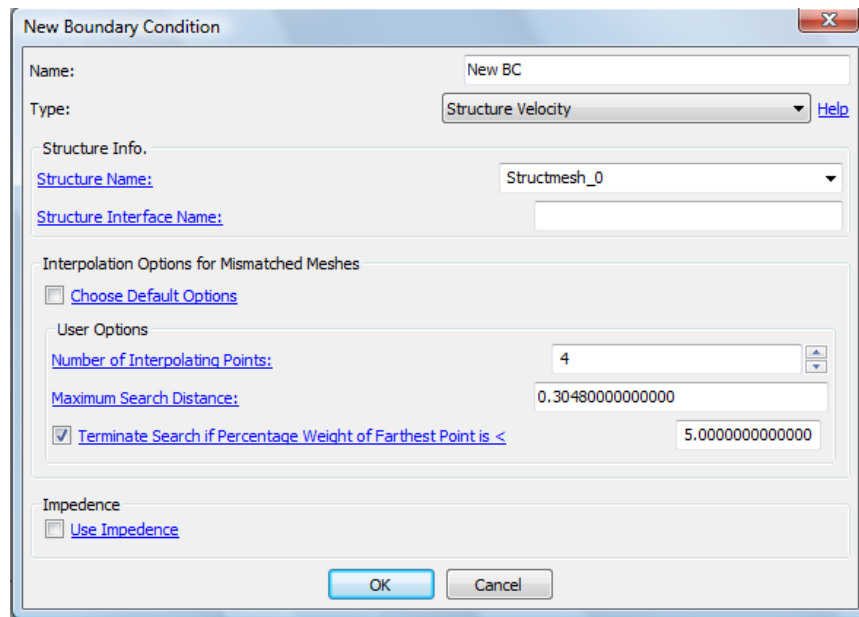
Figure 6.19 Non-Uniform Normal Velocity BC.

Other predefined variables/functions that can be used in the script are *AngularFreq* ( $\omega$ ) – frequency in radians/sec, *GetSoundSpeed*("MaterialName") ( $c$ ) – speed of sound in the medium with the same units as those defined in materials, *GetWaveNumber*("MaterialName") ( $k = \frac{\omega}{c}$ ), and *GetAmbientDensity*("MaterialName") ( $\rho$ ) – density of the medium with the same units as those defined in materials.

**6.5.7.1. 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 for the definition of impedance implemented in *Coustyx*. The impedance is assigned through a script which uses the function call *GetImpedance*. The function takes in the predefined variables *PosnVec* and *NormalVec* as the input arguments. Other predefined variables/functions such as *AngularFreq*, *GetSoundSpeed*("MaterialName") etc., can also be used to compute impedance. (Figure 6.40)

### 6.5.8. 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 6.20.)



**Figure 6.20** Structure Velocity BC.

**6.5.8.1. Structure Name** Select the structure name from which the Boundary Conditions are applied. (Figure 6.20.)

**6.5.8.2. 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 6.20.)

**6.5.8.3. Interpolation Options for Mismatched Meshes** Refer to Section 6.

**6.5.8.4. 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 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*. (Figure 6.20.)

### 6.5.9. 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

$$\alpha p + \beta v_n = \gamma \quad (6.12)$$



The dialog box is titled "New Boundary Condition". It has a "Name" field containing "New BC" and a "Type" dropdown menu set to "Arbitrary Uniform". Below this are three sections for frequency dependence: Alpha, Beta, and Gamma. Each section contains a "Frequency Dependence Type" dropdown menu set to "Constant", and two input fields for "Real" and "Imaginary" values, both set to "0.0". At the bottom of the dialog are "OK" and "Cancel" buttons.

Figure 6.21 Arbitrary Uniform BC

where  $p$  is the pressure,  $v_n$  is the normal velocity at a point on the element, and  $\alpha, \beta, \gamma$  are variables defined by any of the frequency dependent types: *Constant*, *Table*, or *Script*. The values of  $\alpha, \beta$  and  $\gamma$  don't vary with position over the element. (Figure 6.21.)

#### 6.5.10. 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

$$\alpha(\mathbf{x}, \mathbf{n}_x)p + \beta(\mathbf{x}, \mathbf{n}_x)v_n = \gamma(\mathbf{x}, \mathbf{n}_x) \quad (6.13)$$

where  $p$  is the pressure,  $v_n$  is the normal velocity at a point on the element, and  $\alpha, \beta, \gamma$  are variables that vary with position ( $\mathbf{x}$ ) and normal ( $\mathbf{n}_x$ ) over the element.

The values for  $\alpha, \beta$ , and  $\gamma$  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*( $\alpha$ ), *Beta*( $\beta$ ), and *Gamma*( $\gamma$ ) which are used in the BC. Other predefined variables/functions that can be used in the script are *AngularFreq* ( $\omega$ ) – frequency in radians/sec, *GetSoundSpeed*("MaterialName") ( $c$ ) – speed of sound in the medium with the same units as those defined in materials, *GetWaveNumber*("MaterialName") ( $k = \frac{\omega}{c}$ ), and *GetAmbientDensity*("MaterialName") ( $\rho$ ) – density of the medium with the same units as those defined in materials. (Figure 6.22.)

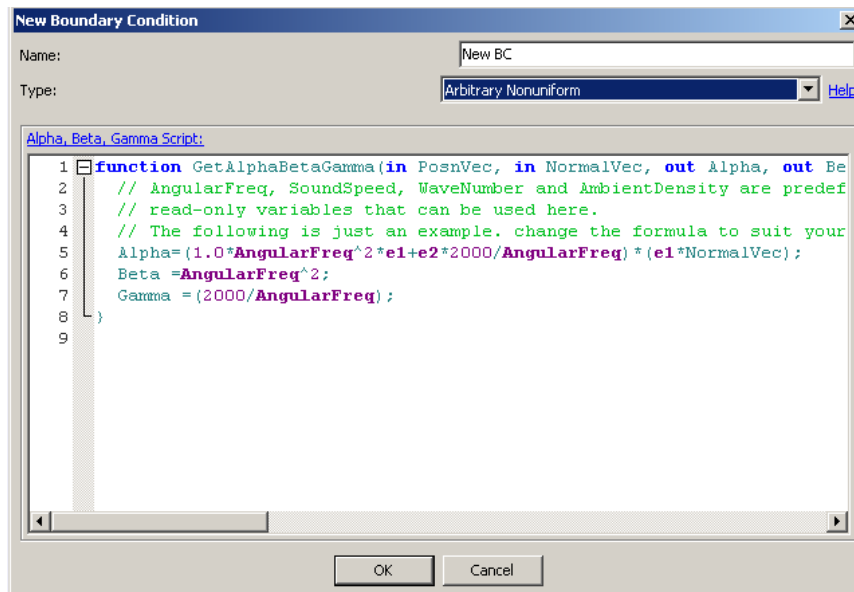


Figure 6.22 Arbitrary Non-Uniform BC

## 6.6. Indirect BE Model BCs

The primary variables in *Indirect* BE models are *pressure jumps* or *double layer potential* ( $\mu$ ), and *velocity jumps* or *single layer potential* ( $\sigma$ ). 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 6.11).

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 6.23 for the element sides definition.

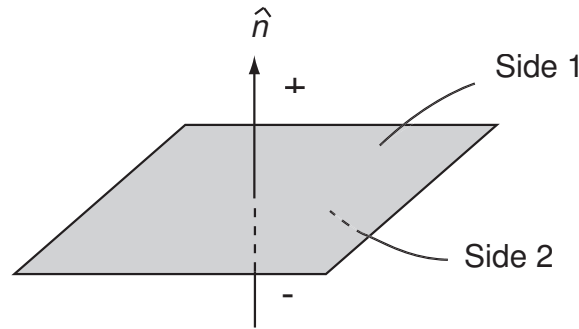
The Boundary Conditions offered in *Indirect* models could be categorized into three broad groups for better understanding:

- **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. [3] 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  $\mu$  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 6.23** 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 6.12). Below is the description of each of the boundary condition options provided in *Indirect* model.

### 6.6.1. 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 6.24)

### 6.6.2. 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

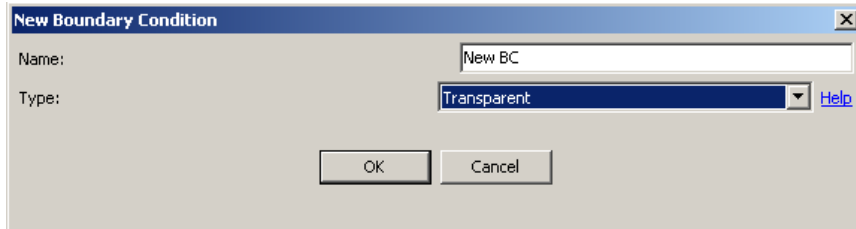


Figure 6.24 Transparent BC

the fluid medium. Refer Section 6 for the definition of impedance implemented in *Coustyx*. The relation between pressure and velocity at anechoic termination is given by

$$\frac{p}{v_n} = Z_o = \rho_o c \quad (6.14)$$

where  $\rho_o$  is the mean density of the surrounding medium, and  $c$  is the speed of sound in the medium. (Figure 6.25.)

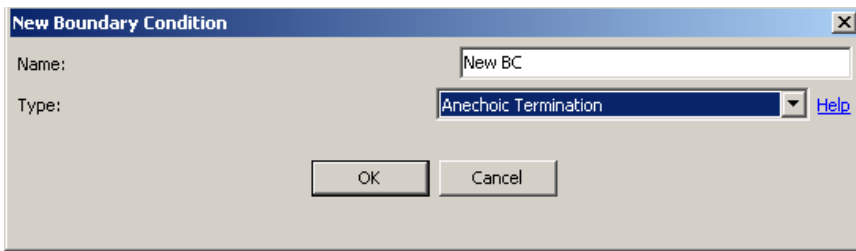


Figure 6.25 Anechoic Termination BC

### 6.6.3. Perforated BC

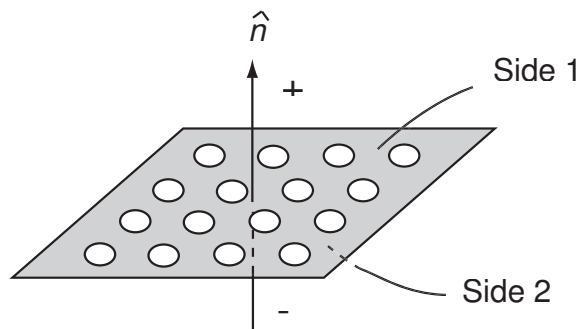


Figure 6.26 Perforated Plate

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

$$p^+ - p^- = -\rho_o c \zeta [v_n - \bar{v}_{sn}] \quad (6.15)$$

Figure 6.27 Uniform Perforated

where  $\zeta$  is the non-dimensional transfer impedance of the perforated surface,  $\rho_o$  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 ([2]) or they can define their own transfer impedance from the following choices in the drop-down menu **Perforated Model Type**:

**6.6.3.1. Sullivan and Crocker** The non-dimensional transfer impedance  $\zeta$  used in the Sullivan and Crocker model ([2]) is,

$$\zeta = \frac{[0.006 - ik_o(t_w + 0.75d_h)]}{\chi} \quad (6.16)$$

where  $k_o$  is the wave number,  $t_w$  is the plate thickness,  $d_h$  is the hole diameter, and  $\chi$  is the porosity of the plate (Figure 6.26) (Note: the actual relation from [2] 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  $\chi$  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 6.27.)

**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.

**6.6.3.2. 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  $\zeta$  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).

**6.6.3.3. Use Structure Velocity** Structure velocity can be used by enabling this option. Equation 6.15 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 6.6.3.5 for more details on how to define a structure velocity. If this option is disabled structure velocity in the Equation 6.15 is set to zero. (Figure 6.27.)

**6.6.3.4. Uniform Perforated BC** This Boundary Condition is applied over perforated elements with position-independent parameters. Refer to Section 6.6.3 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 6.27)

The option **Use Structure Velocity** is enabled to define structure velocity used in Equation 6.15. Refer to Section 6.6.3.3 for further details.

**6.6.3.5. Non-uniform Perforated BC** This Boundary Condition is applied over perforated elements with position-dependent parameters. Refer to Section 6.6.3 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* ( $\omega$ ) – frequency in radians/sec, *GetSoundSpeed*(“MaterialName”) ( $c$ ) – speed of sound in the medium with the same units as those defined in materials, *GetWaveNumber*(“MaterialName”) ( $k = \frac{\omega}{c}$ ), and *GetAmbientDensity*(“MaterialName”) ( $\rho$ ) – density of the medium with the same units as those defined in materials. (Figure 6.28.)

The option **Use Structure Velocity** is enabled to define structure velocity used in Equation 6.15. Refer to Section 6.6.3.3 on how to use this option.

#### 6.6.4. 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 6.23 for side 1 and side 2 definitions).

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

where  $p_o$  is the uniform pressure applied on both sides. (Figure 6.29.)

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

#### 6.6.5. 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

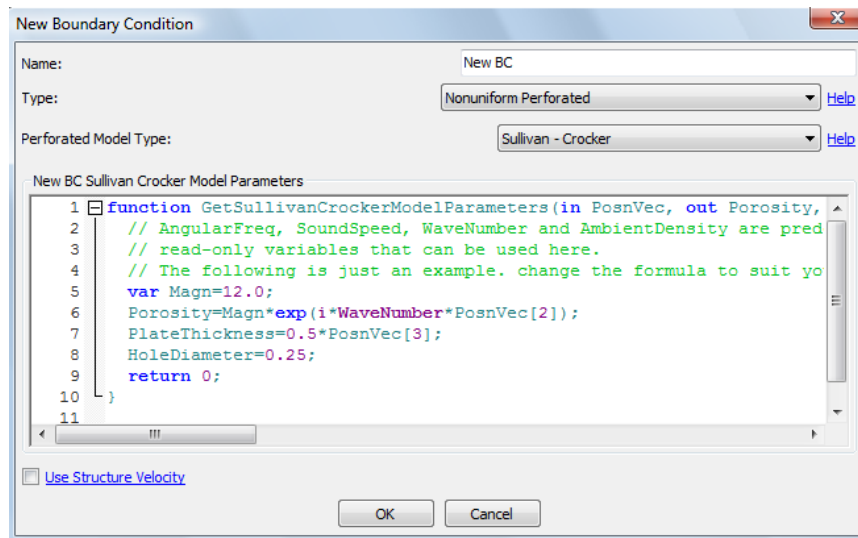


Figure 6.28 Non-Uniform Perforated

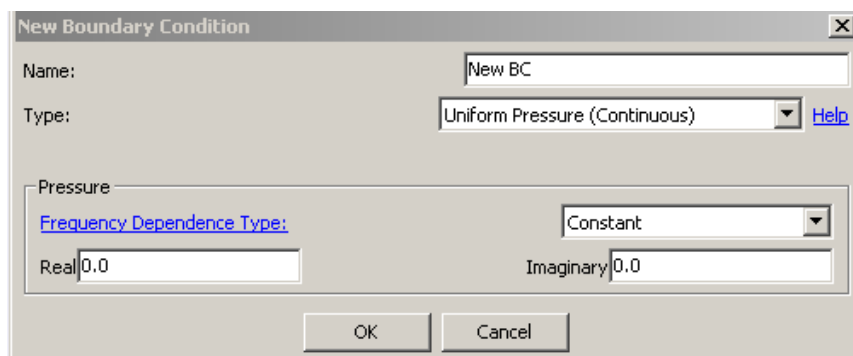


Figure 6.29 Uniform Pressure (Continuous)

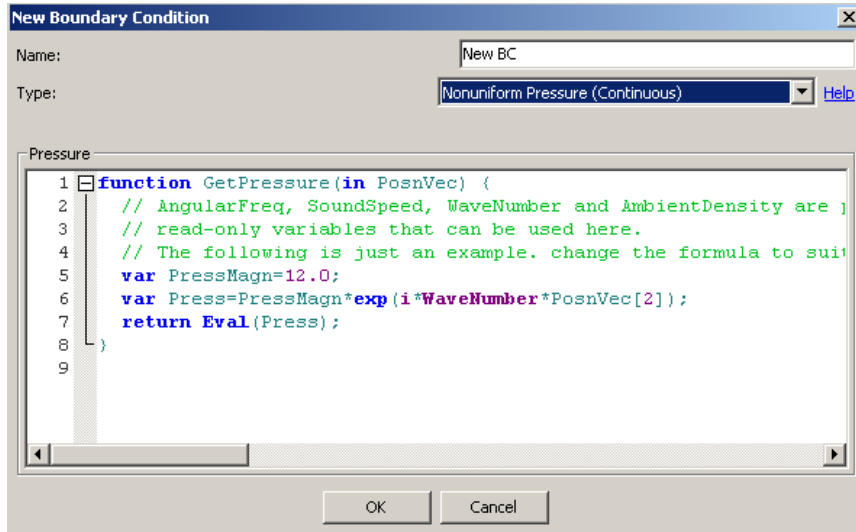


Figure 6.30 Non-Uniform Pressure (Continuous)

( $p^-$ ) are identical (refer to Figure 6.23 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* ( $\omega$ ), *GetSoundSpeed*("MaterialName") ( $c$ ), *GetWaveNumber*("MaterialName") ( $k = \frac{\omega}{c}$ ), and *GetAmbientDensity*("MaterialName") ( $\rho$ ) can also be used in the script. (Figure 6.30.)

#### 6.6.6. Uniform Normal Velocity (Continuous) BC

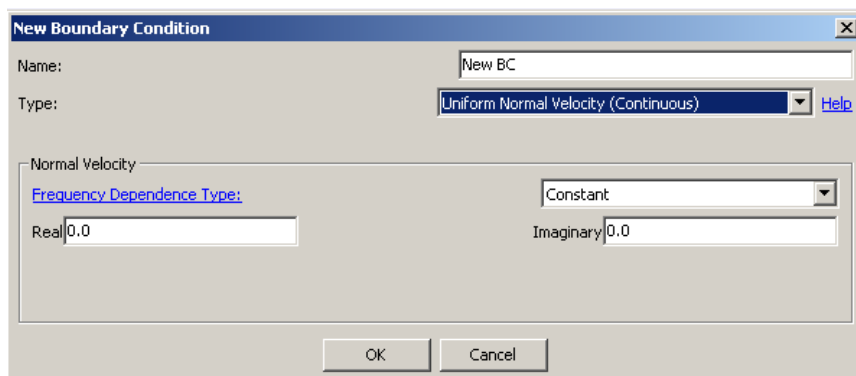


Figure 6.31 Uniform Normal Velocity (Continuous)

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 6.23 for side 1 and side 2 definitions).

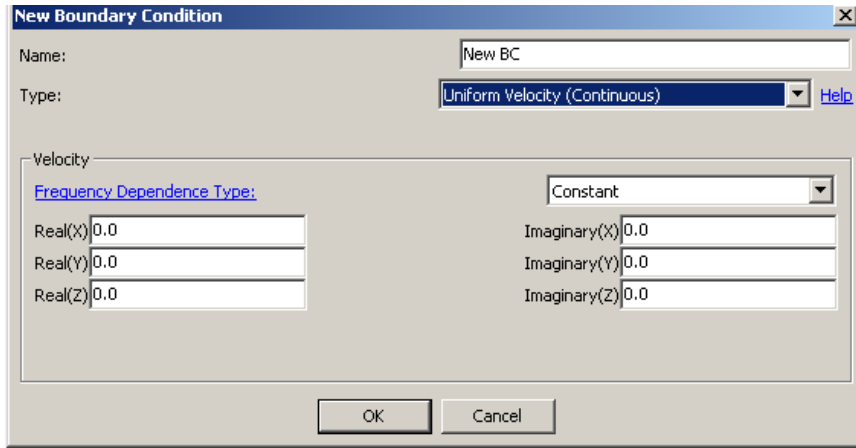
$$v_n^+ = v_n^- = v_{no}$$



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

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 6.31)

### 6.6.7. Uniform Velocity (Continuous) BC



**Figure 6.32** Uniform Velocity (Continuous)

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 6.23 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 6.32)

### 6.6.8. Non-uniform Normal Velocity (Continuous) BC

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 6.23 for side 1 and side 2 definitions). (Figure 6.33)

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

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* ( $\omega$ ), *GetSoundSpeed*("MaterialName") ( $c$ ), *GetWaveNumber*("MaterialName") ( $k = \frac{\omega}{c}$ ), *GetAmbientDensity*("MaterialName") ( $\rho$ ), and reference frame unit vectors **e1**, **e2**, **e3** can also be used in the script. (Figure 6.33)

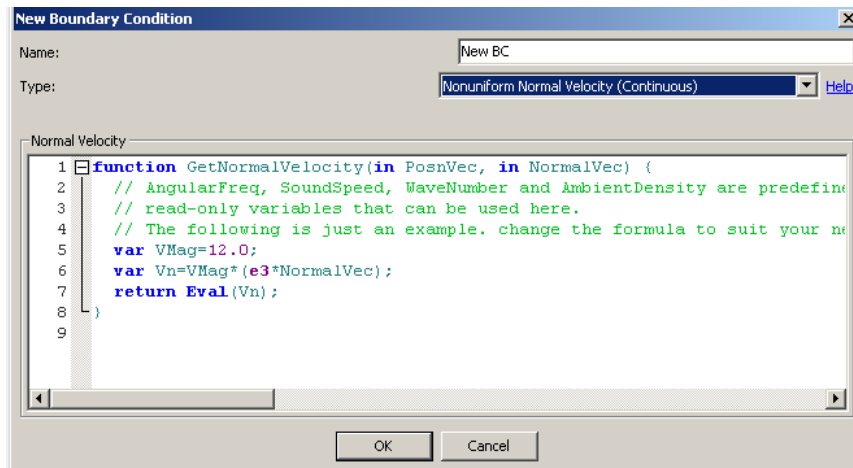


Figure 6.33 Non-Uniform Normal Velocity (Continuous)

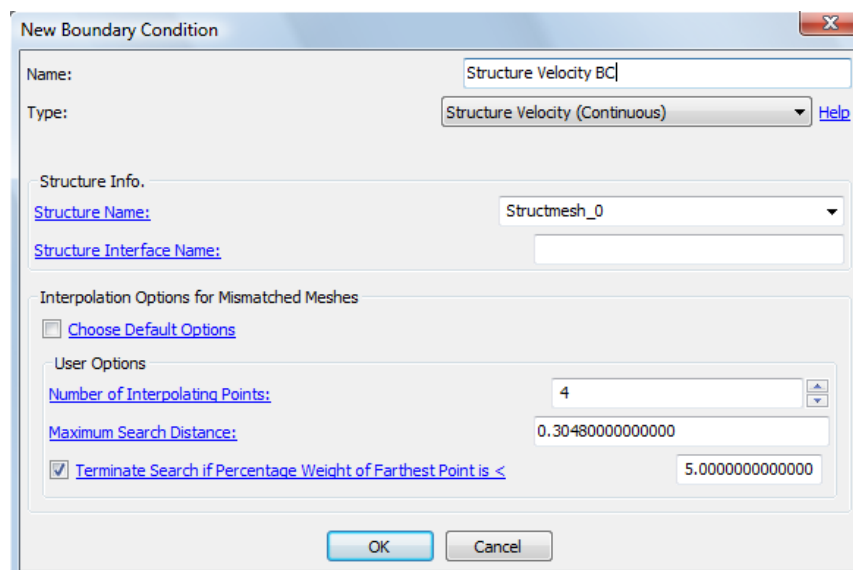


Figure 6.34 Structure Velocity (Continuous)

### 6.6.9. Structure Velocity (Continuous) BC

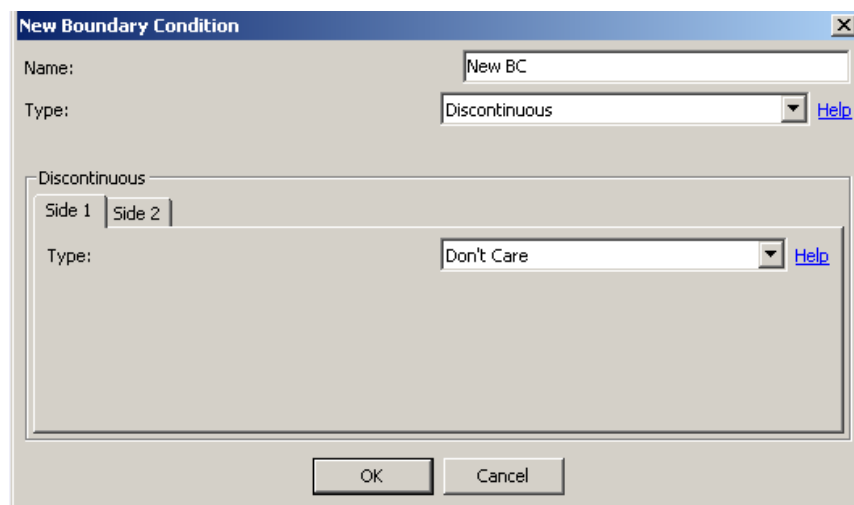
Structure Velocity BC is the most common Boundary Condition in the industry. This can be loaded from frequency response analysis of the structure FEA done externally. Refer to Section 5.1.2 for details on how to load frequency response data. (Figure 6.34)

**6.6.9.1. Structure Name** Select the structure name from which the BCs are applied. (Figure 6.34)

**6.6.9.2. 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 6.34)

**6.6.9.3. Interpolation Options for Mismatched Meshes** Refer to Section 6.

### 6.6.10. Discontinuous BC



**Figure 6.35** Discontinuous BC (Don't Care)

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.

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 specifying different BCs on *Side 1* and *Side 2*. Refer Figure 6.23 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 6.35)

**6.6.10.1. 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 6.35)

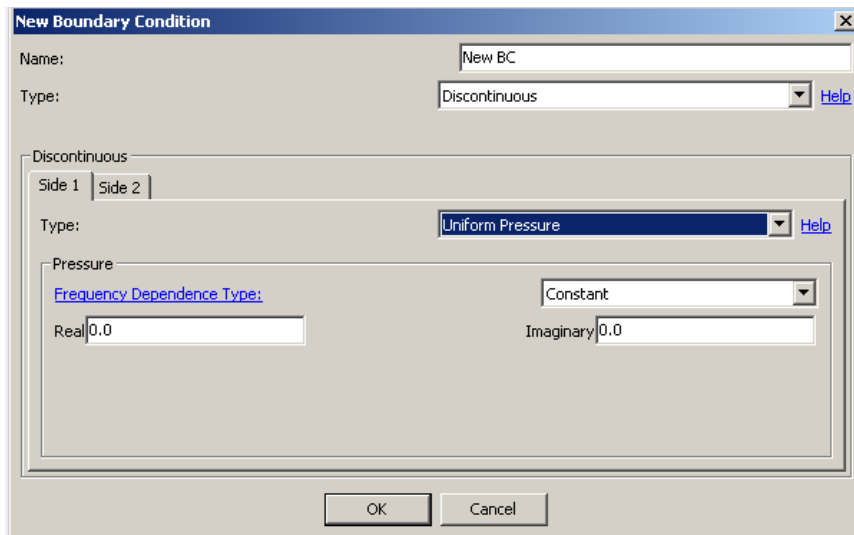


Figure 6.36 Discontinuous Uniform Pressure

**6.6.10.2. 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 be specified by selecting any of the frequency dependence types: *Constant*, *Table*, or *Script*. (Figure 6.36)

**6.6.10.3. 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* ( $\omega$ ), *GetSoundSpeed*("MaterialName") ( $c$ ), *GetWaveNumber*("MaterialName") ( $k = \frac{\omega}{c}$ ), and *GetAmbientDensity*("MaterialName") ( $\rho$ ). (Figure 6.37)

**6.6.10.4. 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 6.23). (Figure 6.38)

**Use Impedance** can be used to specify impedance. Refer to Section 6.5.5.1 for more details. (Figure 6.38)

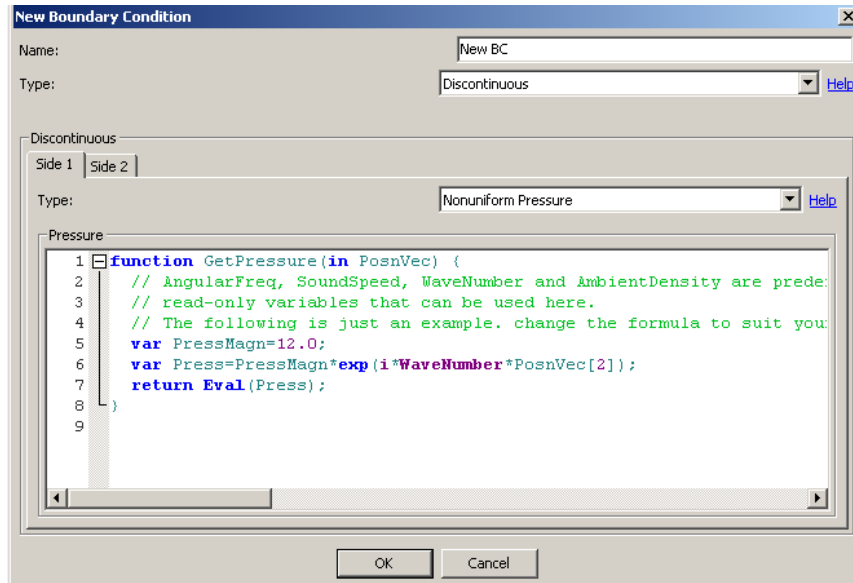


Figure 6.37 Discontinuous Non-Uniform Pressure

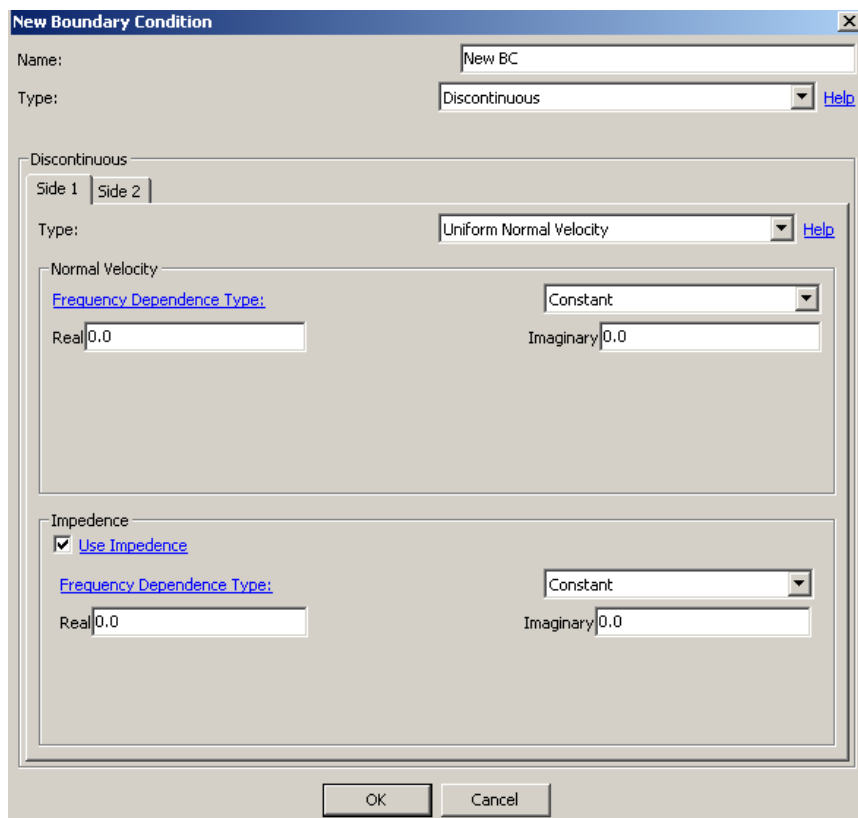


Figure 6.38 Discontinuous Uniform Normal Velocity

The image shows a software dialog box titled "New Boundary Condition". It contains the following fields and options:

- Name:** A text field containing "New BC".
- Type:** A dropdown menu set to "Discontinuous" with a "Help" link.
- Discontinuous section:**
  - Side 1 | Side 2:** Two tabs, with "Side 1" currently selected.
  - Type:** A dropdown menu set to "Uniform Velocity" with a "Help" link.
  - Velocity section:**
    - Frequency Dependence Type:** A dropdown menu set to "Constant".
    - Real(X), Real(Y), Real(Z):** Three text fields, each containing "0.0".
    - Imaginary(X), Imaginary(Y), Imaginary(Z):** Three text fields, each containing "0.0".
- Impedance section:**
  - ☒ **Use Impedance** (with a "Help" link).
  - Frequency Dependence Type:** A dropdown menu set to "Constant".
  - Real:** A text field containing "0.0".
  - Imaginary:** A text field containing "0.0".

At the bottom of the dialog are "OK" and "Cancel" buttons.

**Figure 6.39** Discontinuous Uniform Velocity

**6.6.10.5. 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*.

**Use Impedance** can be used to specify impedance. Refer to Section 6.5.6.1 for more details. (Figure 6.39)

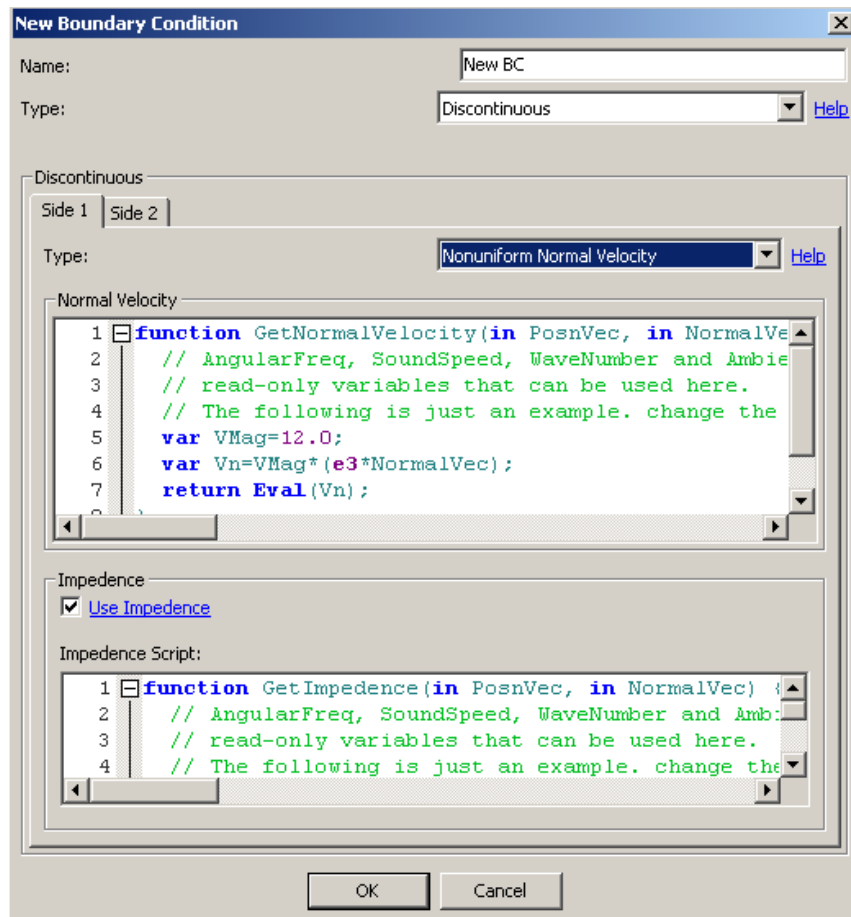


Figure 6.40 Discontinuous Non-Uniform Normal Velocity

**6.6.10.6. 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* ( $\omega$ ), *GetSoundSpeed*("MaterialName") ( $c$ ), *GetWaveNumber*("MaterialName") ( $k = \frac{\omega}{c}$ ), *GetAmbientDensity*("MaterialName") ( $\rho$ ), and reference frame unit vectors **e1**, **e2**, **e3**. (Figure 6.40)

**Use Impedance** can be used to specify impedance. Refer to Section 6.5.7.1 for more details. (Figure 6.40)

**6.6.10.7. 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 6.41). 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 6.5.8.1,

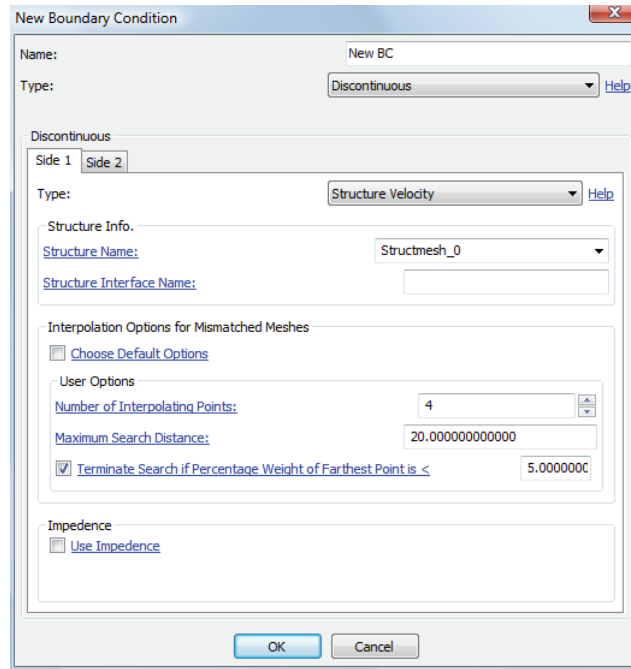


Figure 6.41 Structure Velocity

and Section 6.5.8.2. Refer to Section 6 for details about **Interpolation Options for Mismatched Meshes**. (Figure 6.41)

### 6.6.11. 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 6.42)

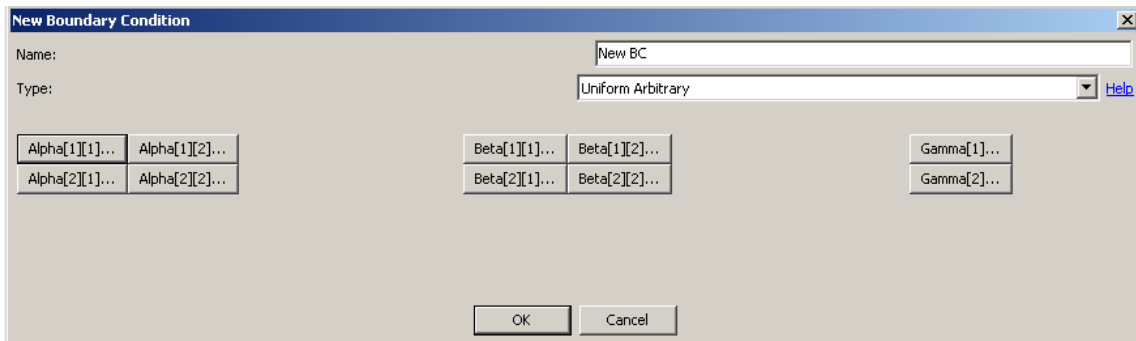


Figure 6.42 Uniform Arbitrary BC

$$\begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} p^+ \\ p^- \end{bmatrix} + \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix} \begin{bmatrix} v_n^+ \\ v_n^- \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \quad (6.17)$$



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  $\alpha_{11}, \alpha_{12}, \alpha_{21}, \alpha_{22}, \beta_{11}, \beta_{12}, \beta_{21}, \beta_{22}, \gamma_1, \gamma_2$  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 6.17 are not independent. They should satisfy the following relations for *Coustyx* to apply the arbitrary BC correctly.

$$\begin{aligned}(\alpha_{11}\alpha_{22} - \alpha_{12}\alpha_{21}) &\neq 0 \\(\beta_{11}\beta_{22} - \beta_{12}\beta_{21}) &\neq 0 \\(\alpha_{21} + \alpha_{22})(\beta_{11} + \beta_{12}) - (\alpha_{11} + \alpha_{12})(\beta_{21} + \beta_{22}) &\neq 0 \\\alpha_{11}\beta_{21} - \alpha_{21}\beta_{11} - \alpha_{12}\beta_{22} + \alpha_{22}\beta_{12} &= 0\end{aligned}\tag{6.18}$$

### 6.6.12. 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 6.17.

The variables  $\alpha_{11}, \alpha_{12}, \alpha_{21}, \alpha_{22}, \beta_{11}, \beta_{12}, \beta_{21}, \beta_{22}, \gamma_1, \gamma_2$  can be defined by the function call *GetAlphaBetaGamma* (Figure 6.43). The input arguments position vector *PosnVec*, and normal vector *NormalVec*, along with other predefined variables such as *AngularFreq* ( $\omega$ ), *GetSoundSpeed*("MaterialName") ( $c$ ), *GetWaveNumber*("MaterialName") ( $k = \frac{\omega}{c}$ ), *GetAmbientDensity*("MaterialName") ( $\rho$ ), 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 6.44, Figure 6.45, and Figure 6.46.

$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix}, \quad \gamma = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}\tag{6.19}$$

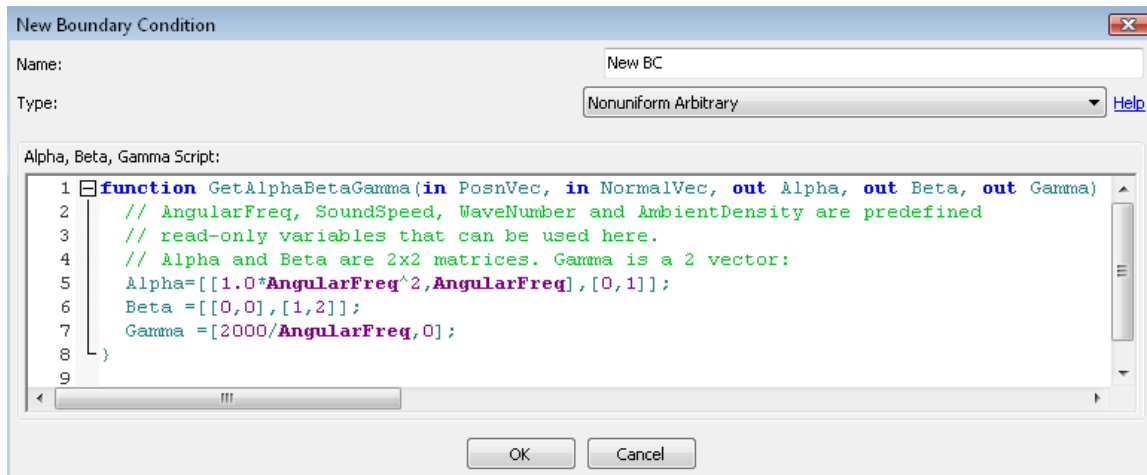


Figure 6.43 Sample script for the function GetAlphaBetaGamma().

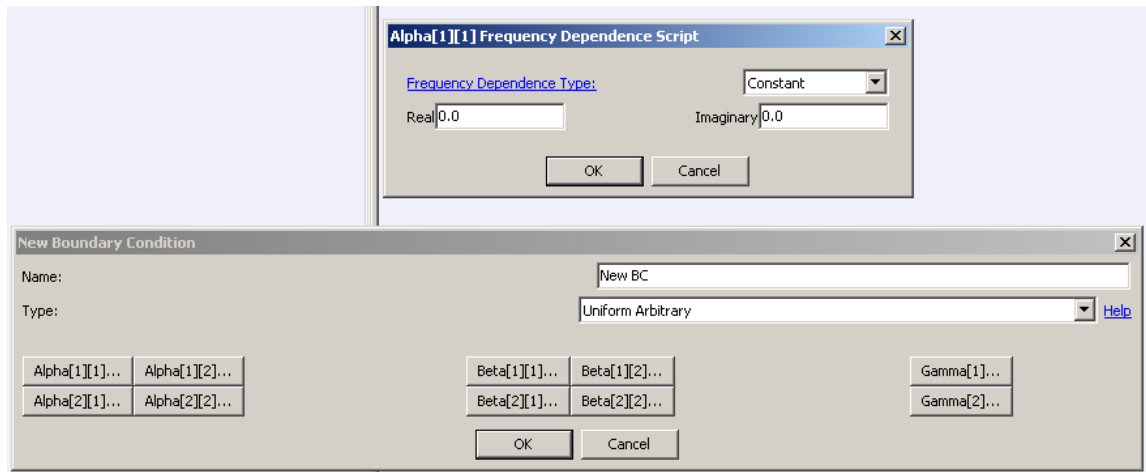


Figure 6.44 Alpha Frequency Dependence Script

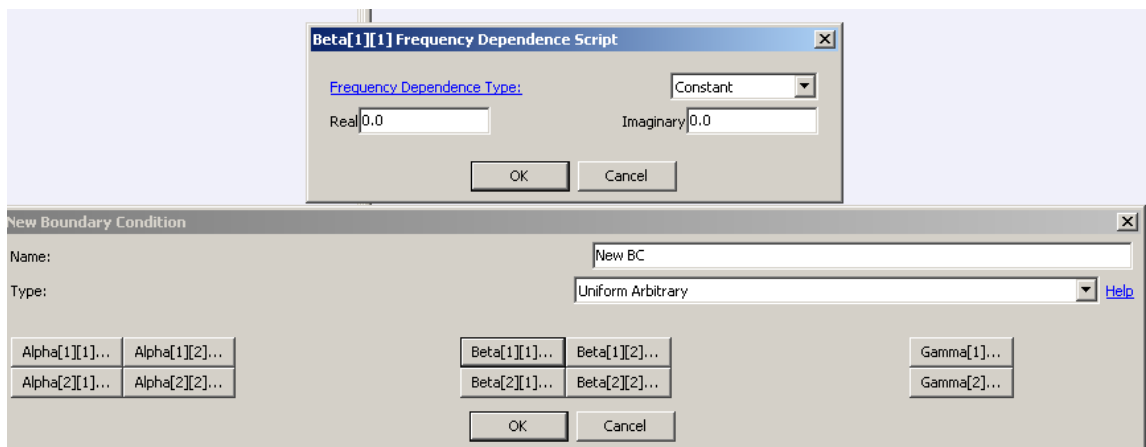


Figure 6.45 Beta Frequency Dependence Script

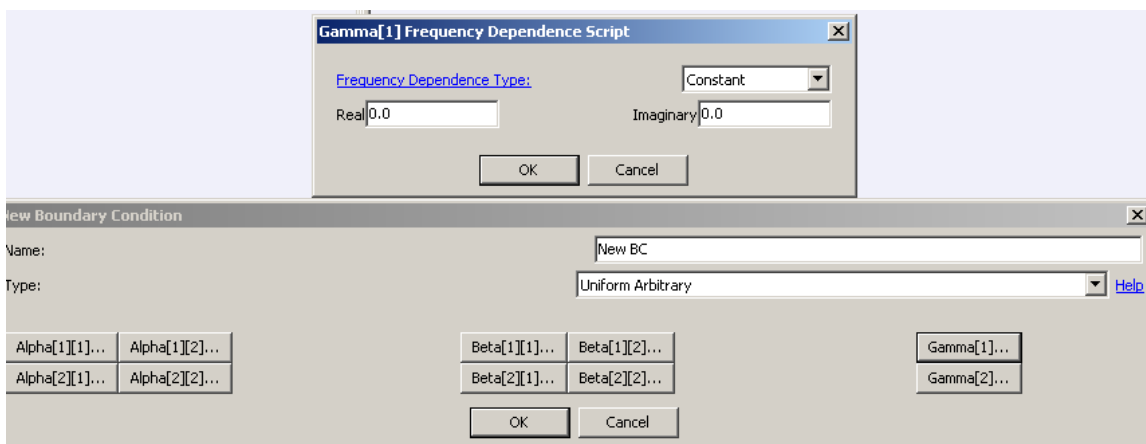


Figure 6.46 Gamma Frequency Dependence Script

## 6.7. 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 6.5 and Section 6.5.8.4.

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

### 6.7.1. Apply BCs directly to Elements

To apply BCs directly to elements follow these steps:

- Open the BE mesh in the GUI.

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 6.47). Note that the boundary condition list is only visible if BCs are defined before hand.

### 6.7.2. 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 6.48)

For an *Indirect* model: right-click on **Model** → **Indirect BE Mesh** → **Sets** and select **New**. (Figure 6.48)

- 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*.

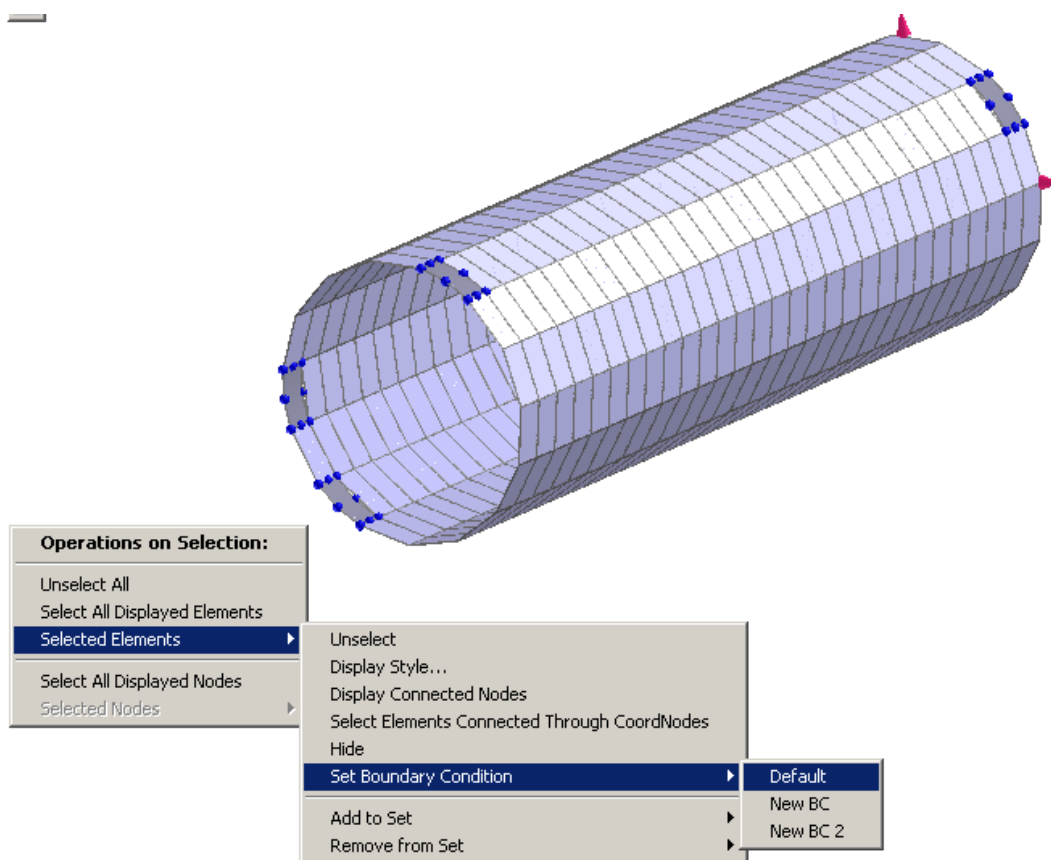
Right-click with the *shift-key* held down and select: **Operations on Selection** → **Selected Elements** → **Add to Set** → *<Set Name>* (Figure 6.49). Note that the list of sets are only visible if sets are defined before hand.

- Apply BC through sets.

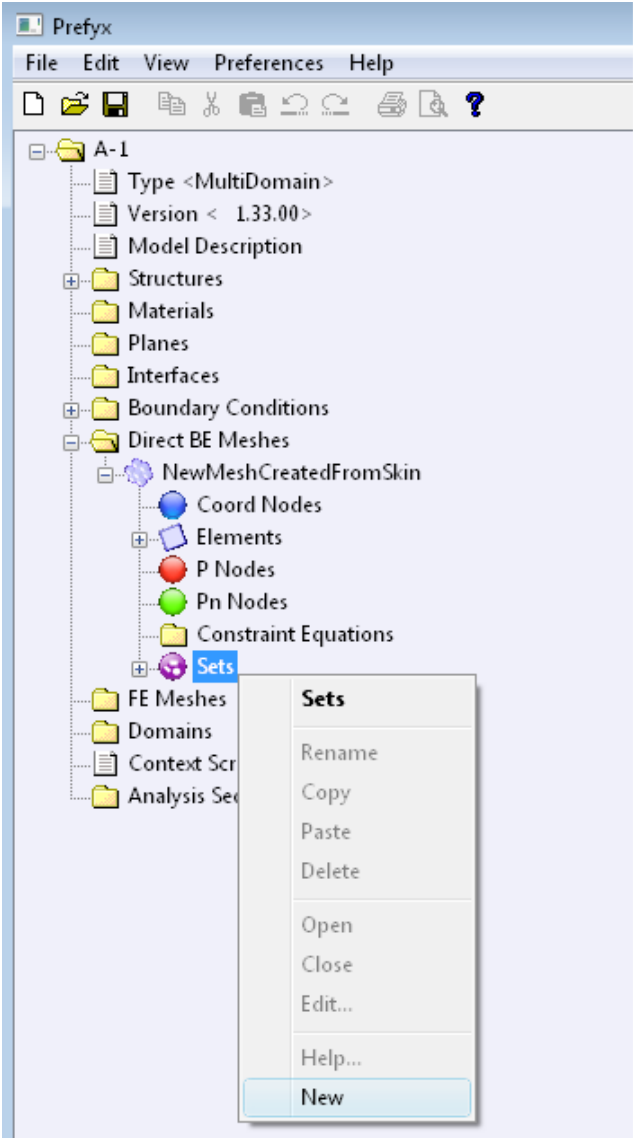
Select the set for a *MultiDomain* model from **Model** → **Direct BE Meshes** → *<Direct BE Mesh>* → **Sets** → *<Set Name>*.

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

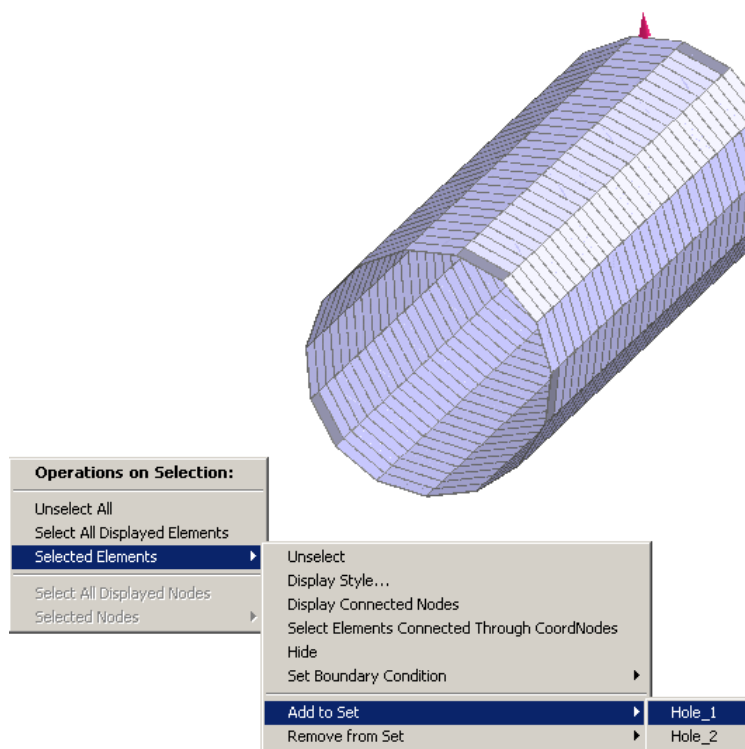
Apply BC to all the elements in the Set through the Set by right-clicking on *<Set Name>* and selecting **Elements** → **Set Boundary Condition** → *<Boundary Condition Name>* (Figure 6.50).



**Figure 6.47** Applying boundary conditions through elements.



**Figure 6.48** Creating a new set, a group of elements and nodes.



**Figure 6.49** Add elements to a set.

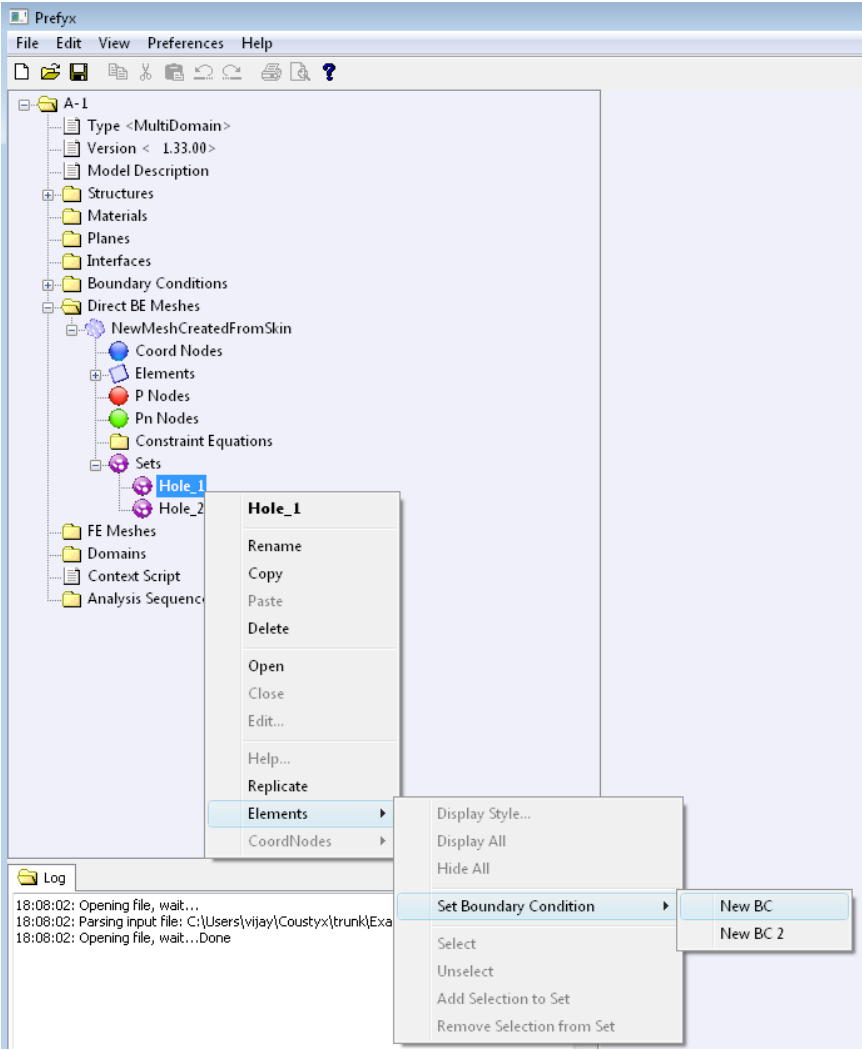


Figure 6.50 Applying boundary conditions through sets.





## REFERENCES

---

- [1] A. D. Pierce. *Acoustics - An introduction to its physical principles and applications*. Acoustical Society of America, 1991. Pages 107–113.
- [2] 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.
- [3] 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.



## 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.
- **Generate Command Script (.csr) File.** Generates a *Coustyx* script .csr file. Figure 7.6. This script file can be used to run the analysis from the command prompt using the following command:

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

Note that the users can externally modify the script file and automate their runs.

If the analysis sequence has been edited manually, a prompt will appear to verify that the command script will ignore any manual changes. (Figure 7.7).

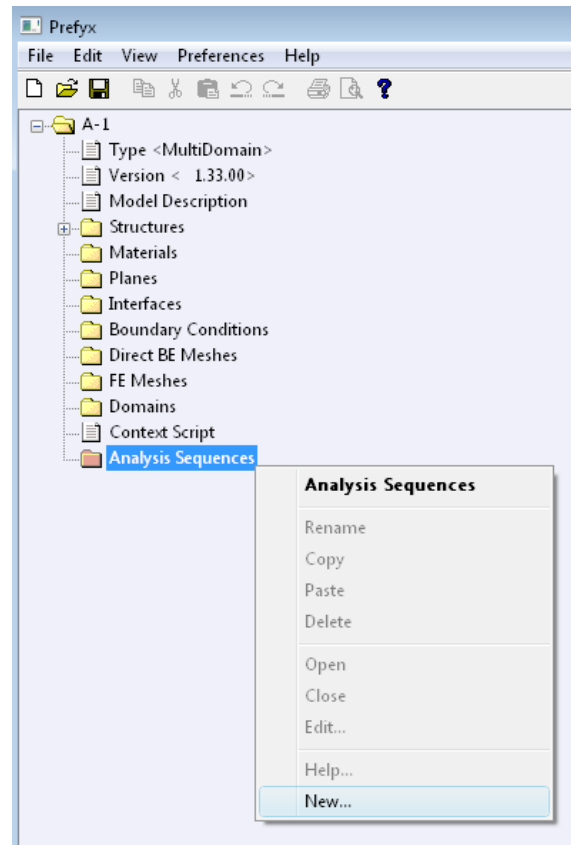


Figure 7.1 Creating new analysis sequence.

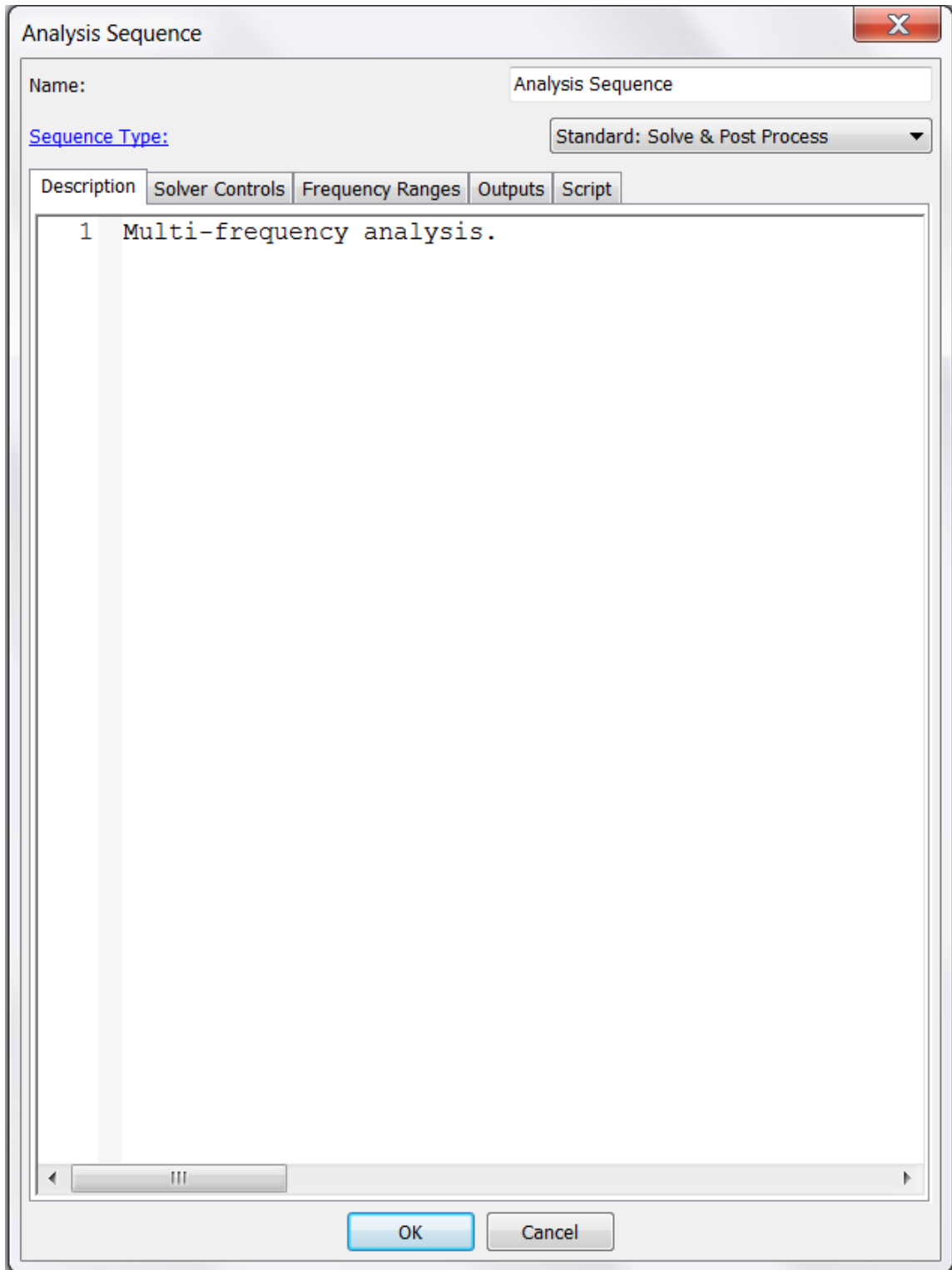
## 7.1. Sequence Type

Select one of the five sequence types from the drop-down menu: **Acoustic Response: Solve & Post Process**, **Acoustic Response: Load Solution & Post Process**, **Coupled Vibro-Acoustic Response**, **Coupled Vibro-Acoustic Response: Diffuse Field Excitation**, or **Muffler Transmission Loss: Three Point Method**.

**Acoustic Response: Solve & Post Process** Use this sequence type to first solve the acoustic problem and then perform post-processing from the surface solution.

**Acoustic Response: Load Solution & Post Process** Use this sequence type to load solution from the results file created by an earlier analysis and perform post-processing. Refer to Section 7.3.1.5 on how to create a binary results file. Enter the file name with full path in the **FileName:** box or use **Browse...** button to find the desired file.

**Coupled Vibro-Acoustic Response** Use this sequence to solve a two-way coupled vibro-acoustic problem and output coupled structures frequency responses in addition to other standard acoustic field outputs. Note that the selection of this sequence does not preempt other model setup requirements that make



**Figure 7.2** Analysis sequence edit dialog box.

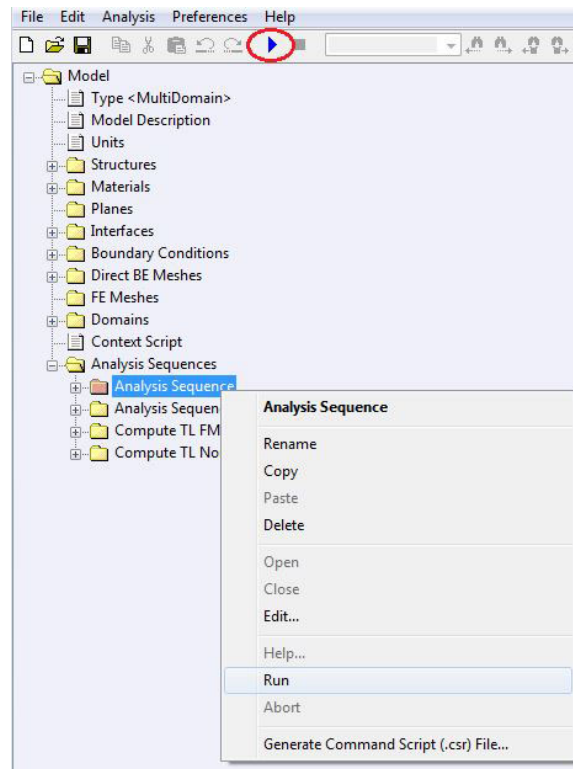


Figure 7.3 Run new analysis sequence.

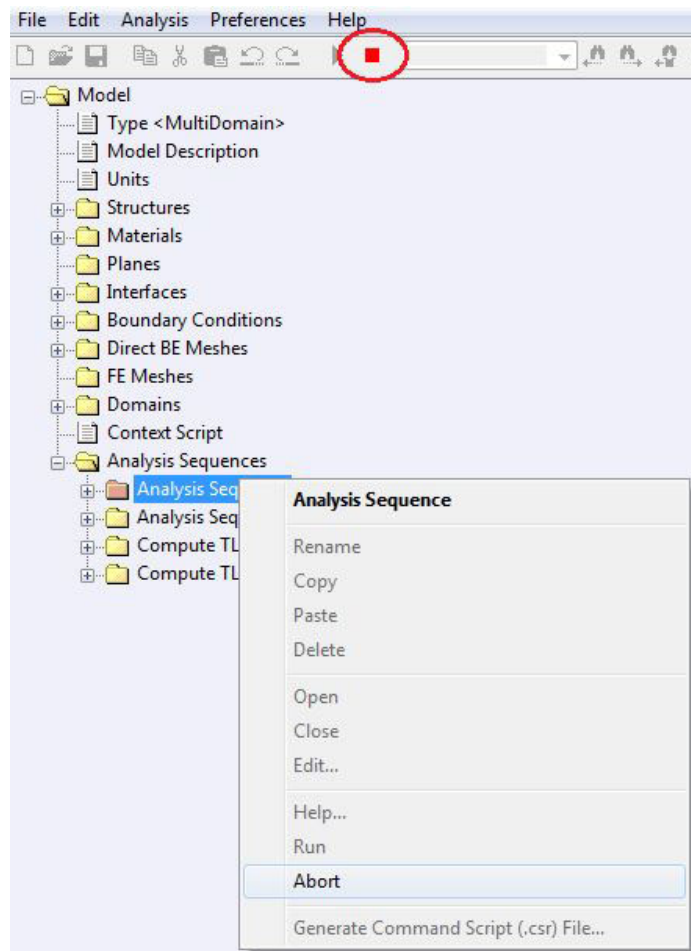
a coupled analysis possible, such as applying structure velocity boundary condition from a coupled structure, etc.

**Coupled Vibro-Acoustic Response: Diffuse Field Excitation** Use this sequence to solve for structure response due to diffuse sound field loading. The diffuse sound field is modeled by a superposition of uncorrelated plane waves directed towards the structure from multiple equally probable directions. This analysis would be useful to predict random response of space structures, such as satellite antenna reflectors, solar panels, etc., which are exposed to large broadband (random) acoustic loads during lift-off and ascent.

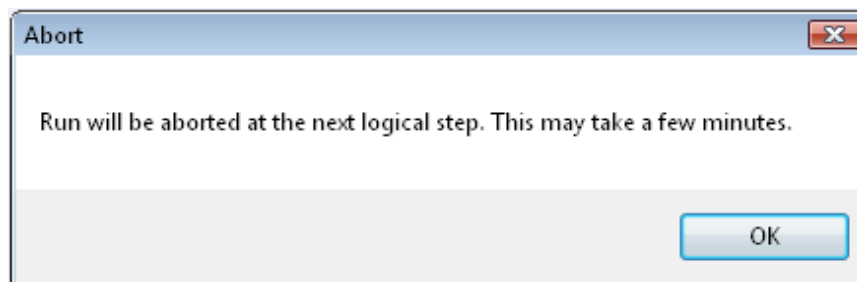
**Muffler Transmission Loss: Three Point Method** Use this sequence to solve for transmission loss of mufflers or intake manifolds using the industry standard three-point method. Note that the selection of this sequence does not preempt the model setup requirements, such as setting up the right boundary conditions to perform such analysis, etc.

Figure 7.2 shows an Analysis Sequence edit dialog box. The dialog box consists of *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* the selected sequence type are found in the *tabbed* windows: **Solver Controls**, **Frequency Ranges (Frequency Filter)**, **Just-In-Time Loading** and **Diffuse Sound Field**. 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 selections in the previous *tabbed* windows. Advanced users can directly modify the script to suit their requirements.

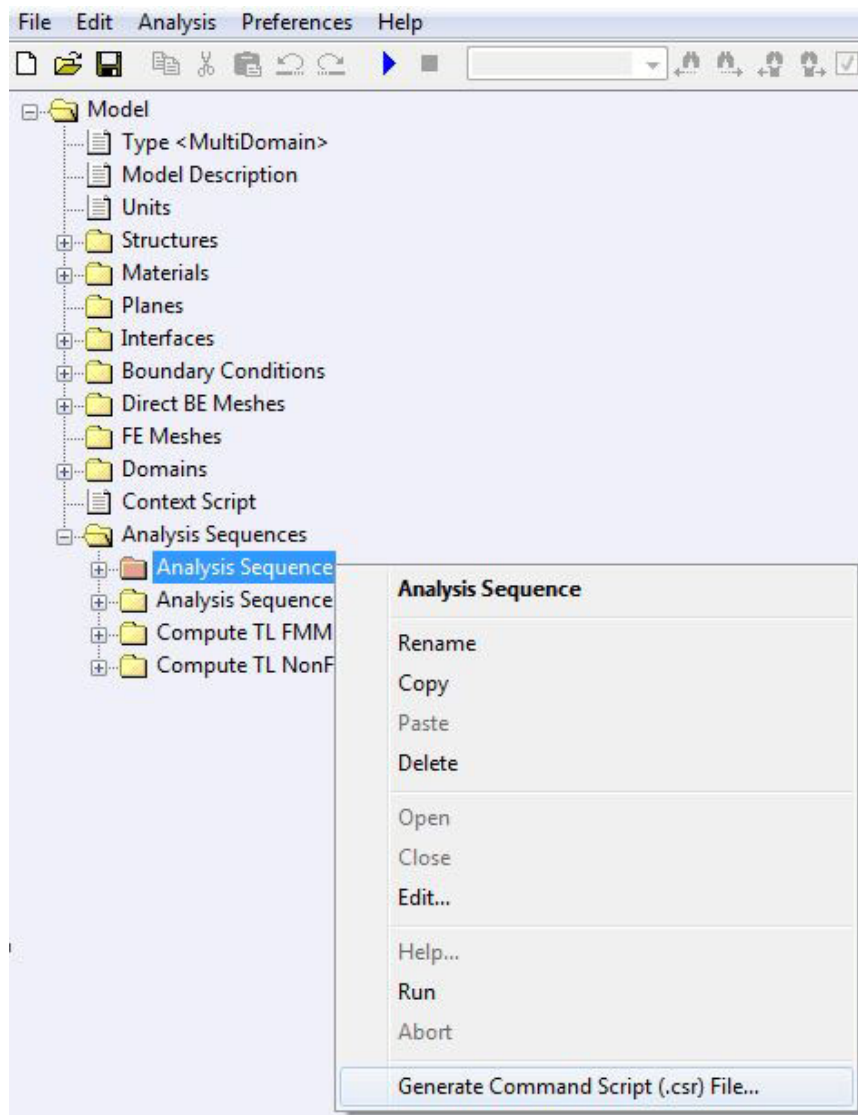
Users should exercise reasonable caution in the selection of the input parameters due to their influence on the solution accuracy and the speed of the analysis.



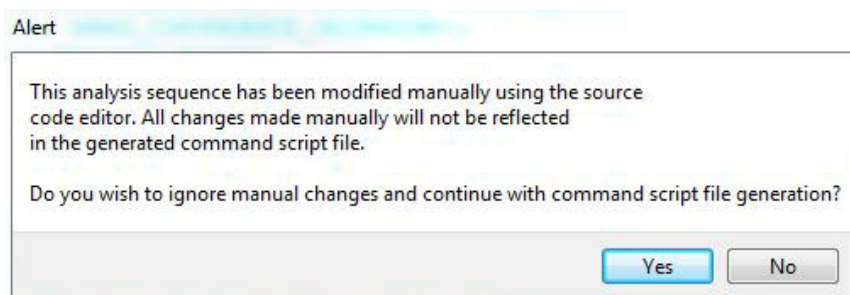
**Figure 7.4** Abort the analysis run.



**Figure 7.5** Abort message.



**Figure 7.6** Generate command script.



**Figure 7.7** Generate command script alert message.



## 7.2. Inputs

Input parameters required to run an analysis are spread over **Solver Controls**, **Frequency Ranges (Frequency Filter)**, **Just-In-Time Loading** and **Diffuse Sound Field** *tabbed* windows.

### 7.2.1. Solver Controls

The **Solver Controls** window is shown in Figure 7.8.

The screenshot shows the 'Analysis Sequence 1' dialog box with the 'Solver Controls' tab selected. The window has a title bar with a close button (X). The main content area is divided into several sections with labels and input fields:

- Name:** Analysis Sequence 1
- Sequence Type:** Standard: Solve & Post Process (dropdown menu)
- Tabbed Interface:** Description, Solver Controls (selected), Frequency Ranges, Just-In-Time Loading, Outputs, Script
- Parallel Processing:**
  - ☒ Use Maximum Possible Number of CPUs
  - Number of CPUs requested:** 1 (spin box)
- Formulation Type:** HIE Collocation (dropdown menu)
- FMM:**
  - ☒ Use FMM
  - ☒ Precompute Near Field Matrices
  - FMM Transition Method:** Speed (dropdown menu)
  - Number of FMM Levels: (7 Levels Suggested)** 7 (spin box)
- Solution Method:** GMRES (dropdown menu)
- GMRES:**
  - Preconditioner Type:** Near Field (dropdown menu)
  - Number Bands including Diagonal:** 1 (spin box)
  - Convergence Criterion:** Residual (dropdown menu)
  - Residual as a Percentage:** 0.5000000000000000 (text field)
  - Change in Sound Power as a Percentage:** 0.5000000000000000 (text field)
  - Number of Vectors in Krylov Subspace at Restart:** 1000 (spin box)
  - Initial Guess:** Zero Vector (dropdown menu)
- Integration:**
  - ☒ Use Fixed Integration Order:
  - Integration Order for Triangular Elements:** 3 (spin box)
  - Integration Order for Quadrilateral Elements:** 3 (spin box)
  - Variable Order Integration Scheme:** Medium (dropdown menu)

At the bottom of the dialog box are 'OK' and 'Cancel' buttons.

**Figure 7.8** Solver controls window.

**7.2.1.1. 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 7.8)

**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 7.8)

**7.2.1.2. 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. The variational formulation minimizes the sound power functional to yield unknown surface variables. For a *MultiDomain* model this formulation overcomes the thin-shape breakdown experienced by HIE Collocation for closely spaced meshes. However, this formulation uses double integrals and is slower than HIE Collocation.

**HIE Collocation** Select this option to perform the analysis using the Helmholtz Integral Equation (HIE) Collocation method. In HIE Collocation, the surface unknowns are determined by satisfying the Helmholtz Integral Equation *exactly* at the collocation points. This formulation avoids double integrals and hence is faster. For exterior problems, the HIE Collocation formulation suffers from the irregular frequency or non-uniqueness issue, that is, at frequencies corresponding to the eigenvalues of the complementary interior problem, the boundary element solution yields incorrect answers. This is remedied by using CHIEF overdetermination points.

**Burton Miller Galerkin** Select this option to perform the analysis using the Burton Miller formulation. In the Burton Miller formulation, the weighted residual in the boundary integral equations is minimized. This leads to system coefficient matrices that are not symmetric. This is a theoretically robust way of handling the irregular frequency issue.

**Burton Miller Collocation** Select this option to perform the analysis using the Burton Miller (BM) Collocation method. In this formulation, in addition to the Helmholtz integral equation, the normal derivative integral equation is also used. A linear combination of these equations is guaranteed to yield accurate solutions at all frequencies, so the user need not worry about spurious solutions at certain frequencies. In this method, the HIE and NDIE equations are exactly satisfied at the collocation points which are located close to the nodes but are on the interior of element (so certain smoothness requirements of the pressure field are satisfied). This formulation gives good solution at all frequencies and is very robust. Compared to the Burton Miller Galerkin method, this method does not involve double integrals and is faster. To get good accuracy with the BM Collocation in FMM, it is suggested to use FMM transition method of Accuracy or Best Blend.

**7.2.1.3. 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 7.8)

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

**Speed** This option uses the signature function representation for the multipole coefficients which results in *diagonal* translation operators. Select this option for faster run times  $O(p^2)$ . However, for very small model dimensions, and very low frequency, this approach is susceptible to sub-wavelength breakdown. Hence the solution may be less accurate at very low frequencies.

**Accuracy** This option uses multipole coefficients as the basis in the FMM upward and downward passes. We use Rotation-Coaxial Translation-Rotation approach to result in a  $O(p^3)$  method as opposed to  $O(p^4)$  method. Select this option to obtain accurate solutions at all frequencies. This method, however, is slower  $O(p^3)$  than the **Speed** option.

**Best Blend** This option combines the best features of **Speed** and **Accuracy** options to yield precise error control without sacrificing speed. It uses multipole coefficient representation at finer levels in FMM hierarchy, and switches to signature function representation at coarser levels, as necessary.

**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 7.9).

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 7.10 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 7.10), far cells appear only for number of FMM levels  $\geq 3$ ). 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  $30 \times 30$  elements on each side. Select the *number of 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.

**7.2.1.4. 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 7.12).

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

Analysis Sequence 1

Name: Analysis Sequence 1

Sequence Type: Standard: Solve & Post Process

Description Solver Controls **Frequency Ranges** Just-in-Time Loading Outputs Script

Parallel Processing

☒ Use Maximum Possible Number of CPUs

Number of CPUs requested: 1

Formulation Type: HIE Collocation

FMM

☒ Use FMM

☒ Precompute Near Field Matrices

FMM Transition Method: Speed

**Number of FMM Levels: (7 Levels Suggested)** 7

Solution Method: GMRES

GMRES

Preconditioner Type: Near Field

Number Bands including Diagonal: 1

Convergence Criterion: Residual

Residual as a Percentage: 0.5000000000000000

Change in Sound Power as a Percentage: 0.5000000000000000

Number of Vectors in Krylov Subspace at Restart: 1000

Initial Guess: Zero Vector

Integration

☒ Use Fixed Integration Order:

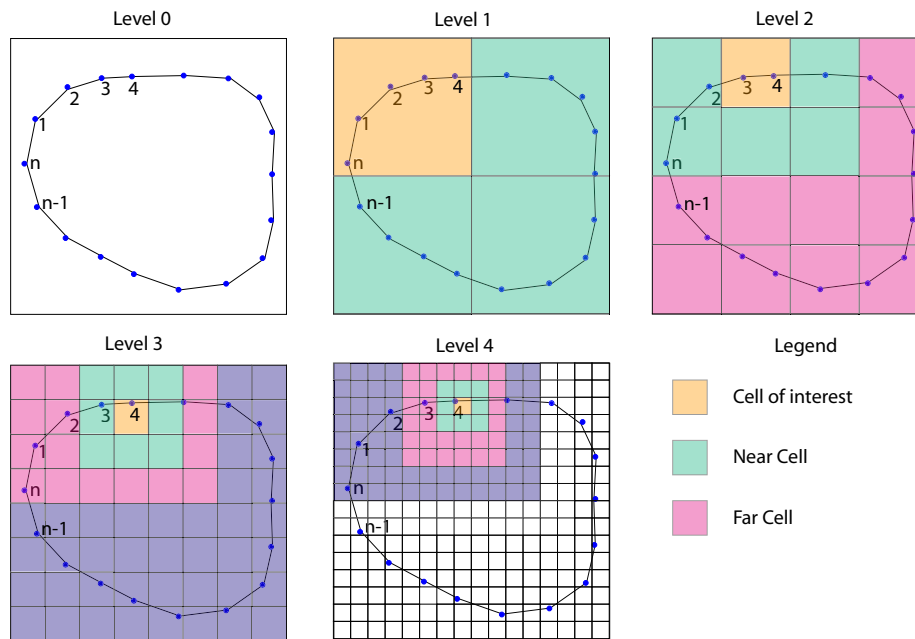
Integration Order for Triangular Elements: 3

Integration Order for Quadrilateral Elements: 3

Variable Order Integration Scheme: Medium

OK Cancel

**Figure 7.9** Suggested number of FMM levels.



**Figure 7.10** Computational cell hierarchy constructed at different FMM levels.

**7.2.1.5. GMRES** The contents of this panel are activated only when GMRES is selected as the **Solution Method**. *Couстыx* 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 7.8).

**Preconditioner Type** Preconditioners are used in iterative solvers to improve spectral properties of the system matrices for faster convergence. *Couстыx* implements three preconditioners which can be selected based on the user requirements for reducing memory usage, faster convergence, parallel-processing, etc. (Figure 7.8).

**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.

Analysis Sequence 1

Name: Analysis Sequence 1

Sequence Type: Standard: Solve & Post Process

Description Solver Controls Frequency Ranges Just-In-Time Loading Outputs Script

Parallel Processing

☒ Use Maximum Possible Number of CPUs

Number of CPUs requested: 1

Formulation Type: HIE Collocation

FMM

☒ Use FMM

☒ Precompute Near Field Matrices

FMM Transition Method: Speed

Number of FMM Levels: (7 Levels Suggested) 7

Solution Method: GMRES

GMRES

Preconditioner Type: Near Field

Number Bands including Diagonal: 1

Convergence Criterion: Residual

Residual as a Percentage: 0.5000000000000000

Change in Sound Power as a Percentage: 0.5000000000000000

Number of Vectors in Krylov Subspace at Restart: 1000

Initial Guess: Zero Vector

Integration

☒ Use Fixed Integration Order:

Integration Order for Triangular Elements: 3

Integration Order for Quadrilateral Elements: 3

Variable Order Integration Scheme: Medium

OK Cancel

Figure 7.11 Solution Method FMM

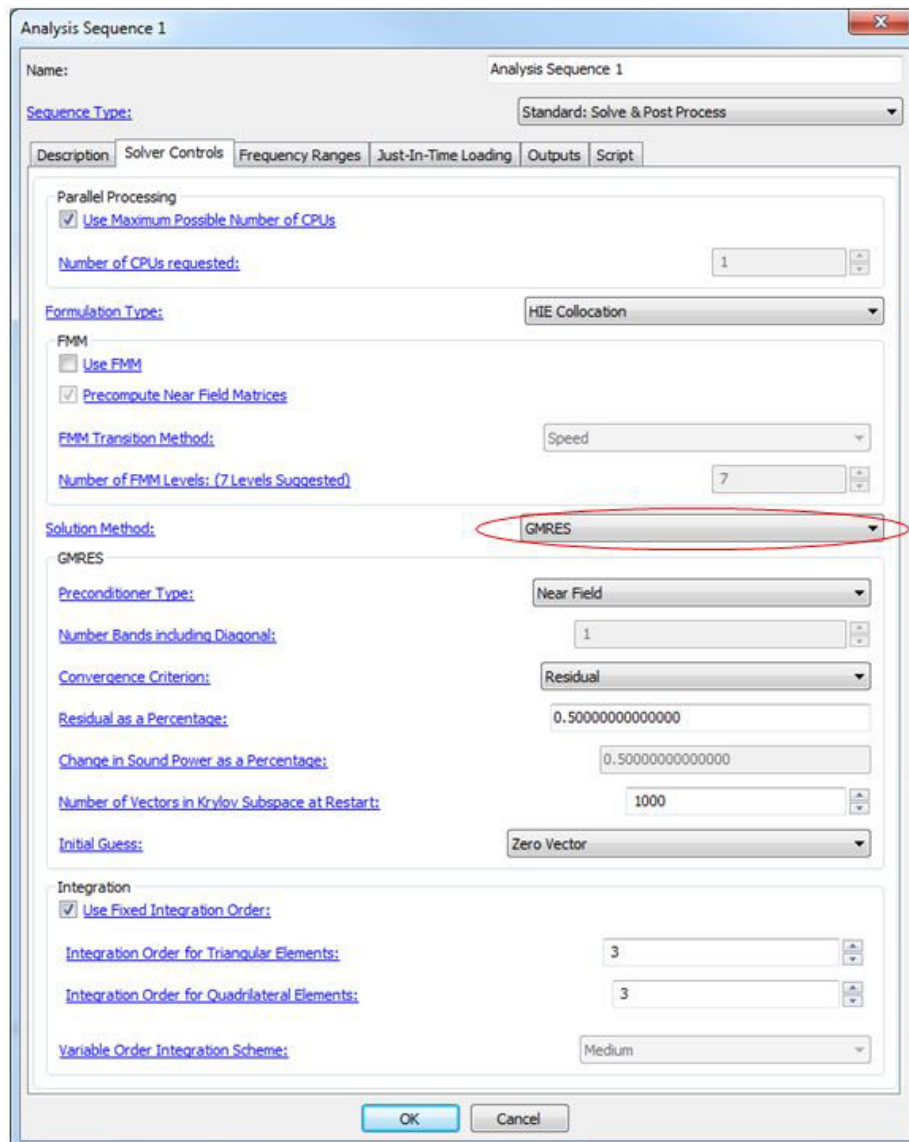


Figure 7.12 GMRES



Analysis Sequence 1

Name: Analysis Sequence 1

Sequence Type: Standard: Solve & Post Process

Description Solver Controls Frequency Ranges Just-In-Time Loading Outputs Script

Parallel Processing

☒ Use Maximum Possible Number of CPUs

Number of CPUs requested: 1

Formulation Type: HIE Collocation

FMM

☐ Use FMM

☒ Precompute Near Field Matrices

FMM Transition Method: Speed

Number of FMM Levels: (7 Levels Suggested) 7

Solution Method: Direct

GMRES

Preconditioner Type: Near Field

Number Bands including Diagonal: 1

Convergence Criterion: Residual

Residual as a Percentage: 0.5000000000000000

Change in Sound Power as a Percentage: 0.5000000000000000

Number of Vectors in Krylov Subspace at Restart: 1000

Initial Guess: Zero Vector

Integration

☒ Use Fixed Integration Order:

Integration Order for Triangular Elements: 3

Integration Order for Quadrilateral Elements: 3

Variable Order Integration Scheme: Medium

OK Cancel

Figure 7.13 Direct Solution Method



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 7.8)

**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 7.8)

**Residual** This option computes the residue from the solution vector at each iteration. It is the most commonly used convergence criterion. (Figure 7.8)

**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 7.8)

**Residual as a Percentage** This option is enabled only when **Residual** option is selected as the **Convergence Criterion**. (Figure 7.8). 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 7.8). 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 7.8).

**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.

**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  $n_{pts} \times n_{pts}$ 

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

**7.2.1.6. 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 7.8).

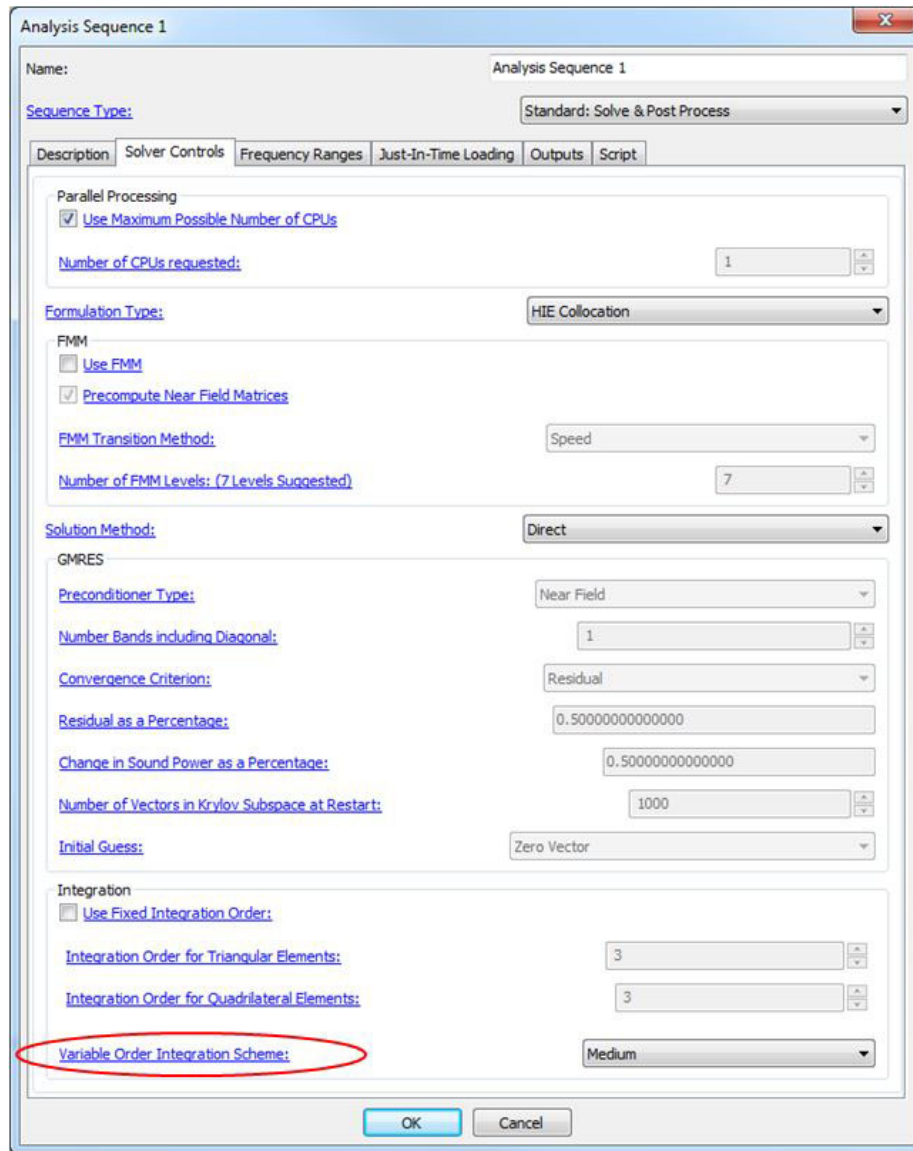
**Use Fixed Integration Order** This option is to set fixed number of quadrature points to be used for all elements. For a FMM case this option is selected by default. The user can't un-select this option when **Use FMM** is already selected.(Figure 7.8).

**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 7.8).

**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 7.8).

**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. 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 7.14)



**Figure 7.14** Variable Order Integration Scheme

**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 7.14)

**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 7.14)

**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 7.14)

**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.

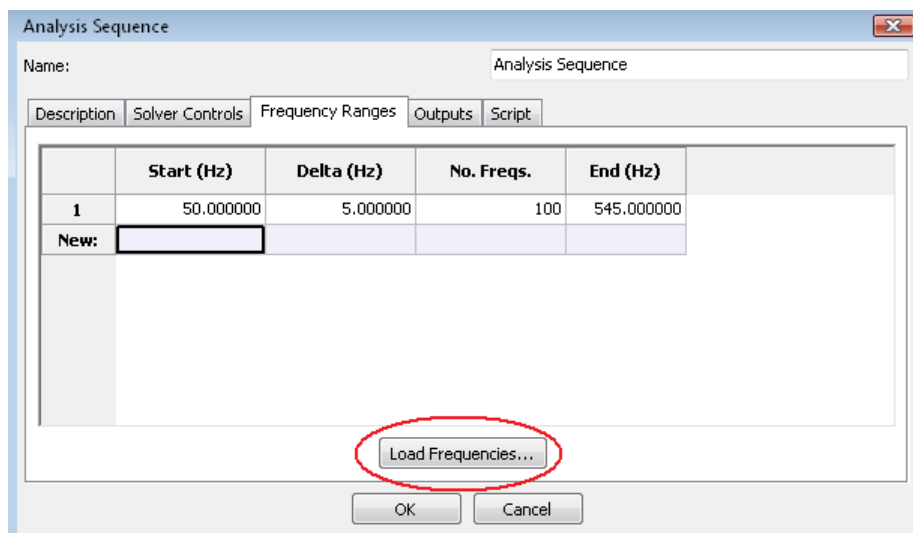
	Quadrilateral			Triangle		
	Medium	Fine	Finest	Medium	Fine	Finest
$D/L < 2$	5	9	11	9	9	9
$2 \leq D/L < 5$	5	7	9	9	9	9
$5 \leq D/L < 20$	3	5	5	9	9	9
$20 \geq D/L$	3	3	3	9	9	9

### 7.2.2. Frequency Ranges (Frequency Filter)

The **Frequency Ranges** window appears when the analysis sequence is of the type: **Standard: Solve & Post Process**, whereas **Frequency Filter** window appears when the sequence is of the type: **Load Solution & Post Process**.

The **Frequency Ranges** window in Figure 7.15 shows a table with columns: **Start(Hz)**, **Delta(Hz)**, **No.Freqs.**, **End(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.

- **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.



**Figure 7.15** Frequency ranges window.

The **Frequency Filter** window in Figure 7.16 shows a table with only two columns, **Start(Hz)** and **End(Hz)**. The solutions for the frequencies lying in between the start and end frequencies in the table are extracted from the results file.

Multiple frequencies can either be added manually to the table or can be loaded into the table by pressing on **Load Frequencies...** button (Figure 7.15). More details on various options provided to load frequencies are discussed next.

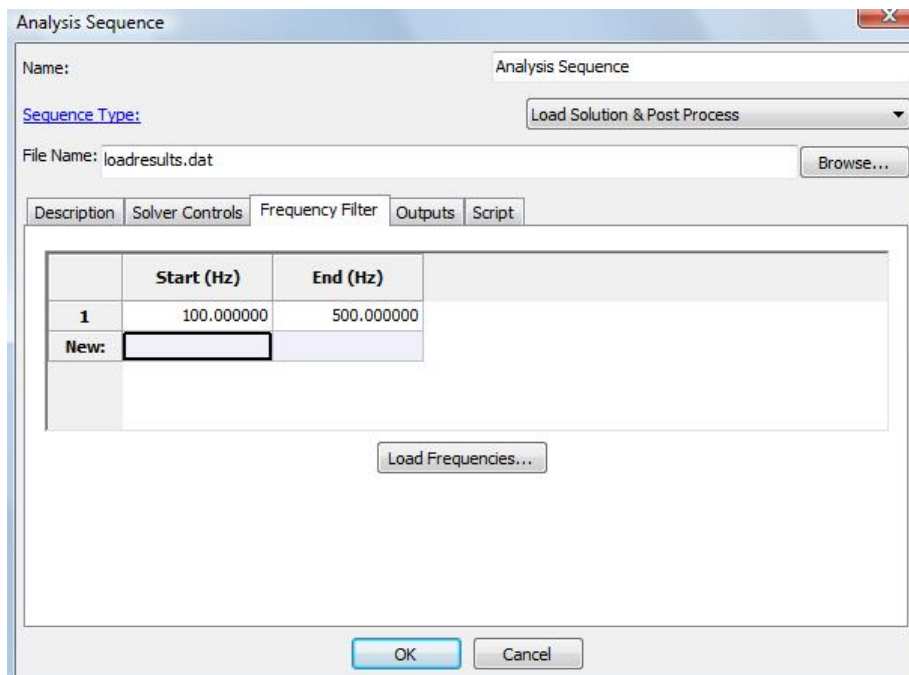


Figure 7.16 Frequency filter window.

- **Source** is the option type from which frequencies are loaded. Frequencies can be loaded from a [Coustyx Results File](#), [ASCII File](#), [Frequency Response Data File](#), [Natural Mode Data File](#), [Structure Frequency 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.

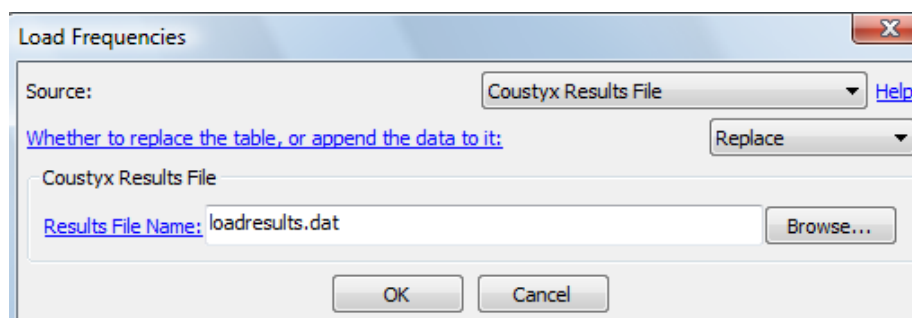
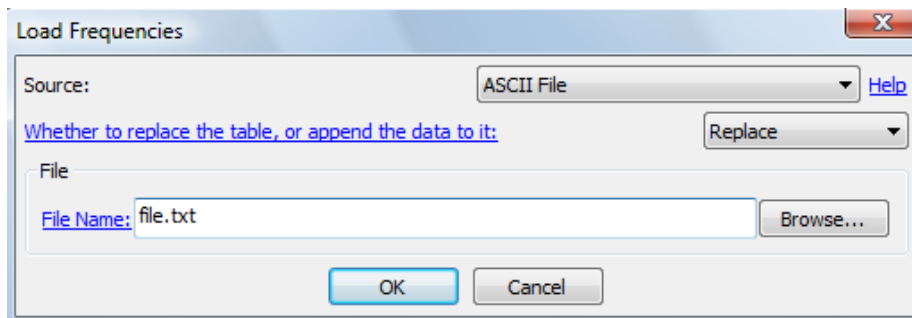


Figure 7.17 Load frequencies from Coustyx Results file window.

**7.2.2.1. Coustyx Results File** Analysis frequencies could be loaded from Coustyx binary results file using this option (Figure 7.17). Refer to Section 7.3.1.5 on how to create Coustyx binary results file.

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

- **Results File Name** is the full path of the Coustyx binary results file used to load analysis frequencies.

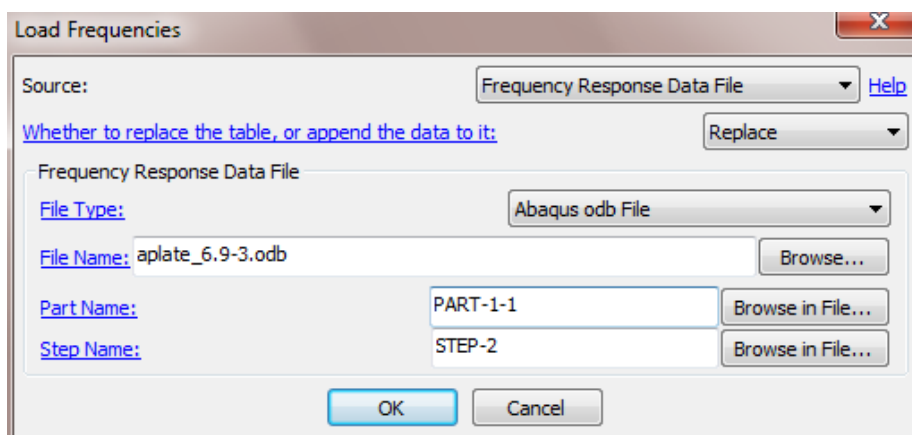


**Figure 7.18** Load frequencies from an ASCII file window.

**7.2.2.2. ASCII File** Analysis frequencies could be loaded from an *ASCII* file using this option (Figure 7.18). 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 (Frequency Filter) table, press OK. To discard changes press CANCEL.

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

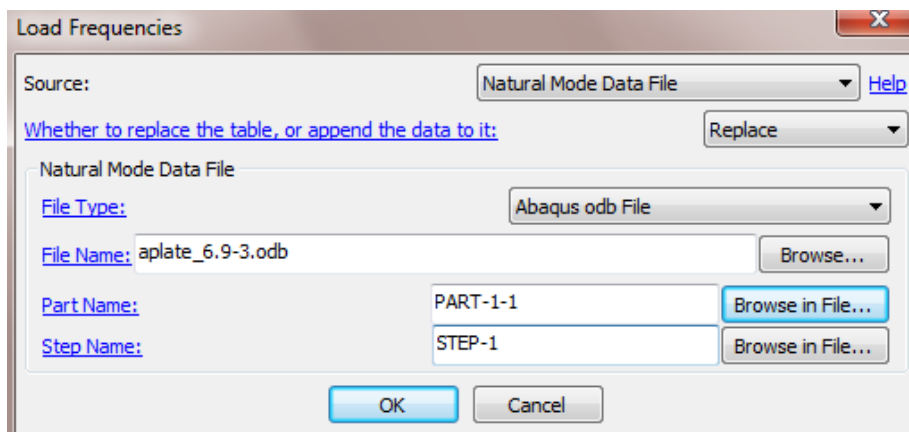


**Figure 7.19** Load frequencies from frequency response data file.

**7.2.2.3. Frequency Response Data File** Analysis frequencies could be extracted from a frequency response data file using this option (Figure 7.19). Select appropriate parameters described below and press

OK to get a new dialog box with the list of available frequencies from the frequency response data file. See Figure 5.6. 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.

- **File Type** Select appropriate FEA output data file type from the drop down menu. Supported FEA output data file formats are: **Abaqus odb File**, **Nastran OP2 File**, **Nastran Punch File**, **Ansys rst File**, **Ansys rfrq File**, and **I-DEAS Universal File**.
- **File Name** Provide the full path of the FEA data file name. Use the **Browse** button to browse folders for the file name.
- **Part Name** This option is applicable only for **Abaqus odb File** format. Provide the part name from which to load the data. Use the **Browse in File** button to browse for all the available parts in the file provided in **File Name**.
- **Step Name** This option is applicable only for **Abaqus odb File** format. Provide the step name from which to load the data. Use the **Browse in File** button to browse for all the available parts in the file provided in **File Name**.

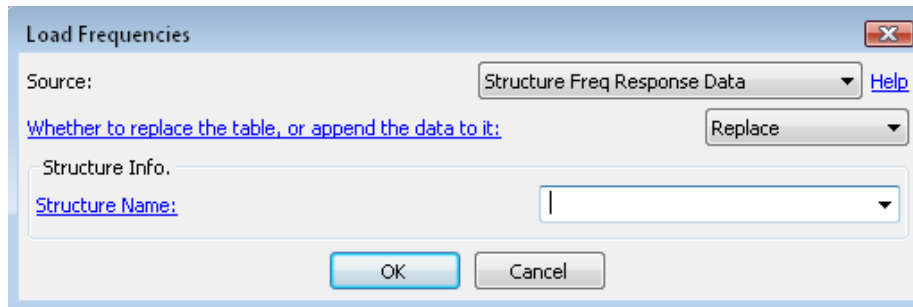


**Figure 7.20** Load frequencies from the natural mode data file.

**7.2.2.4. Natural Mode Data File** Analysis frequencies could be extracted from a natural mode data file using this option (Figure 7.20). Select appropriate parameters and press OK to load natural frequencies. A new dialog box with a list of available natural frequencies from the natural mode data file appear. See Figure 5.8. 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.

- **File Type** Select appropriate FEA output data file type from the drop down menu. Supported FEA output data file formats are: **Abaqus odb File**, **Nastran OP2 File**, **Nastran Punch File**, **Ansys rst File**, **Ansys rfrq File**, and **I-DEAS Universal File**.
- **File Name** Provide the full path of the FEA data file name. Use the **Browse** button to browse folders for the file name.
- **Part Name** This option is applicable only for **Abaqus odb File** format. Provide the part name from which to load the data. Use the **Browse in File** button to browse for all the available parts in the file provided in **File Name**.

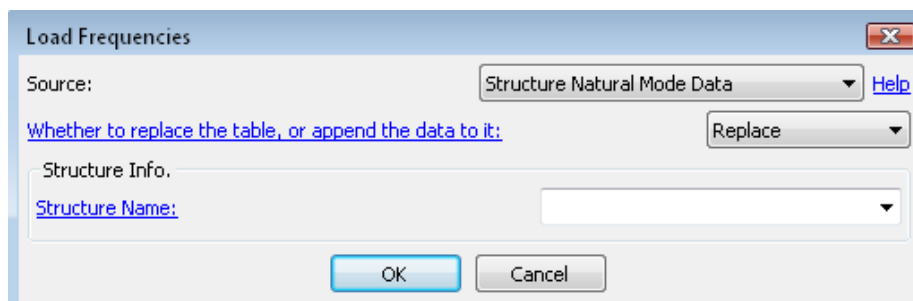
- **Step Name** This option is applicable only for **Abaqus odb File** format. Provide the step name from which to load the data. Use the **Browse in File** button to browse for all the available parts in the file provided in **File Name**.



**Figure 7.21** Load frequencies from the structure frequency response data.

**7.2.2.5. Structure Frequency Response Data** Analysis frequencies could be extracted from a structure's frequency response data using this option (Figure 7.21). 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 5.6. 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.



**Figure 7.22** Load frequencies from the structure natural mode data.

**7.2.2.6. Structure Natural Mode Data** Analysis frequencies could be extracted from a structure's natural mode data using this option (Figure 7.22). 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 5.8. 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.

- **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.

**7.2.2.7. Octave-Band Center Frequencies** The frequency scale for acoustic analysis is usually divided into contiguous *proportional* frequency bands. The partitioning of a *b*th 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

Table 7.4: (continued)

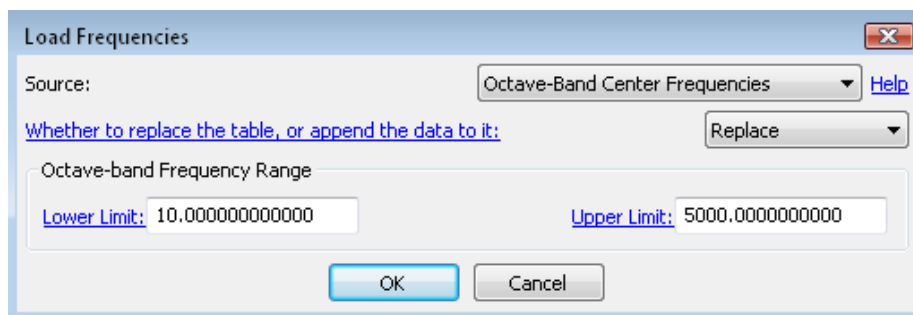
1 Octave	1/3 Octave	1/6 Octave	1/12 Octave	1/24 Octave
2000	1250	1120	1120	1090
				1120
				1150
				1180
		1250	1250	1220
				1250
				1280
				1320
		1400	1400	1360
				1400
				1450
				1500
		1600	1600	1550
				1600
				1650
				1700
	2000	1800	1800	1750
				1800
				1850
				1900
		2000	2000	1950
				2000
				2060
				2120
		2240	2240	2180
				2240
				2300
				2360
		2500	2500	2430
				2500
				2580
				2650
	3150	2800	2800	2720
				2800
				2900
				3000
		3150	3150	3070
				3150

Table 7.4: (continued)

1 Octave	1/3 Octave	1/6 Octave	1/12 Octave	1/24 Octave
4000	4000	3550	3350	3250
				3350
			3550	3450
				3550
			3750	3650
				3750
		4000	4000	3870
				4000
			4250	4120
				4250
			4500	4370
				4500
	5000	4500	4750	4620
				4750
			5000	4870
				5000
		5600	5300	5150
				5300
			6000	5450
				5600
			6300	5800
				6000
	6300	6300	6150	6150
				6300
			6700	6500
				6700
		7100	6900	6900
				7100
			7300	7300
				7500
			7750	7750
				8000
8000	8000	8000	8250	8250
				8500
			8500	8500
				8750
		9000	9000	9000
				9250
			9500	9500
				9500

Table 7.4: (continued)

1 Octave	1/3 Octave	1/6 Octave	1/12 Octave	1/24 Octave
	10000	10000	10000	9750 10000

**Figure 7.23** 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 7.23.
- 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

**7.2.2.8. 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.

**7.2.2.9. 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.

**7.2.2.10. 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.

**7.2.2.11. 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.

### 7.2.3. Just-In-Time Loading

This window allows users set the option to load frequency response data (from FEA data files) during an analysis run. Only the data related to the analysis frequency or nearby frequencies (when frequency interpolation is needed) is extracted from the FEA data files and loaded into the model for the analysis. All the data loaded is cleared at the end of the analysis. One of the main advantages of using this option is that the user can now use frequency response data from really large files without ever loading the entire file (which probably would require lot more memory). Multiple structures and their corresponding data files could be specified by adding each one of the them to the **Structure List** table. Use buttons **Add** or **Delete** to add a new structure or delete existing one from the table respectively. Figure 7.24 shows the **Just-In-Time Loading** window.

**Load Frequency Response from File Just-In-Time** Check this option to load frequency response data during the analysis run. Note that this selection would automatically clear any existing frequency response data for the structures specified in the table.

**Structure Name** Select the name of the structure for which you would like to load the frequency response data.

**File Type** Select the FEA output data file type. Choose from any of the following output file formats: **Abaqus odb File**, **Nastran OP2 File**, **Nastran Punch File**, **Ansys rst File**, **Ansys rfrq File**, and **I-DEAS Universal File**.

**File Name** Provide the full path of the FEA data file name. Use the **Browse** button to browse folders for the file name.

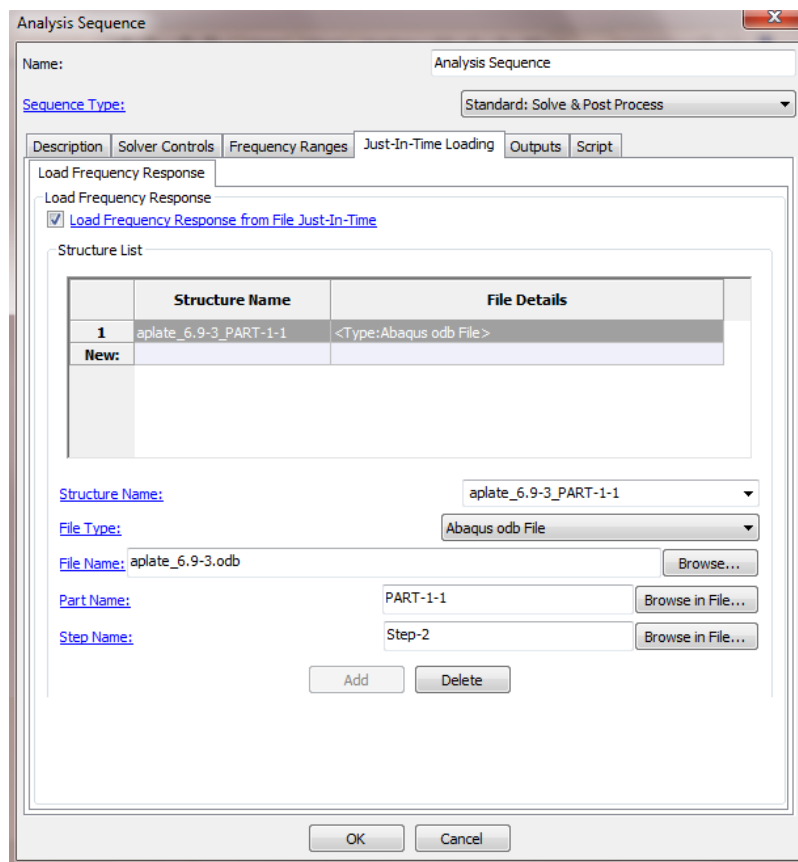
**Part Name** This option is applicable only for **Abaqus odb File** format. Provide the name of the part associated with the structure. Use the **Browse in File** button to browse for all the available parts in the file provided in **File Name**.

**Step Name** This option is applicable only for **Abaqus odb File** format. Provide the name of the step from which the frequency response data should be loaded. Use the **Browse in File** button to browse for all the available steps in the file provided in **File Name**.

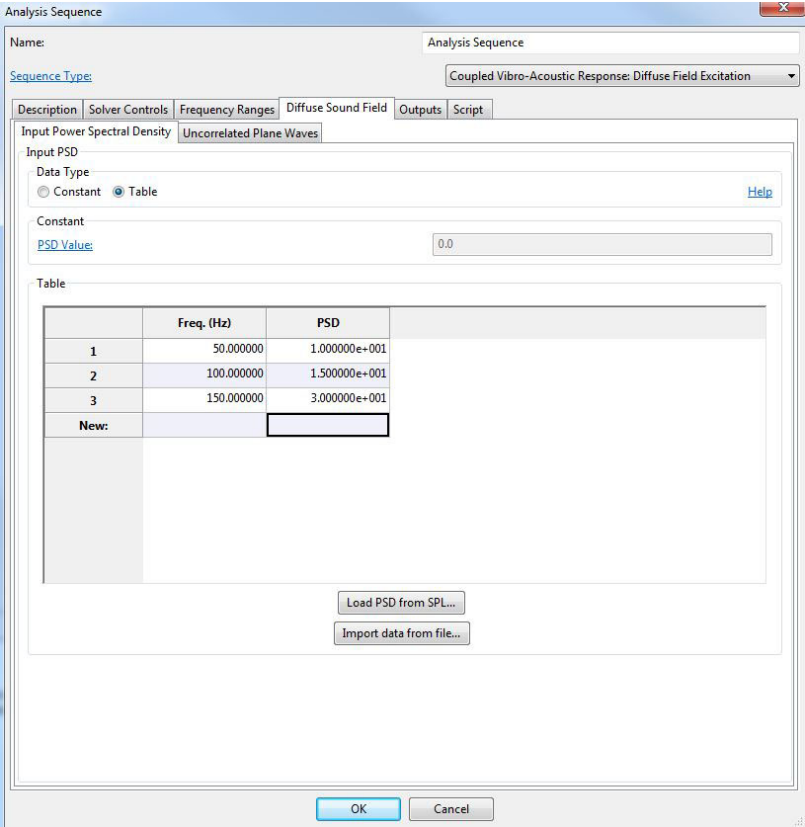
### 7.2.4. Diffuse Sound Field

This window allows users to set the input options for the **Coupled Vibro-Acoustic Response: Diffuse Field Excitation** sequence type. The input power spectral density of the diffuse sound field is specified in the *tabbed* window: **Input Power Spectral Density**. The diffuse sound field is modeled as a superposition of uncorrelated plane waves that are specified in the *tabbed* window **Uncorrelated Plane Waves**. Note that the analysis run would automatically delete any acoustic sources defined in the model.

**7.2.4.1. Input Power Spectral Density** The input power spectral density (PSD) of the diffuse sound field could be specified as a constant (same PSD value at all frequencies) or as a frequency dependent table. Linear interpolation is performed to obtain input PSD value at any other frequency lying between the frequencies specified in the table. Note that the input PSD value shall be specified in appropriate model units consistent



**Figure 7.24** Just-In-Time Loading window.



**Figure 7.25** Diffuse Sound Field: Input Power Spectral Density tab.

with  $\text{Pascal}^2/\text{Hz}$ . The table can be modified by entering values manually or through other options: **Load PSD from SPL** or **Import data from file**.

**Load PSD from SPL** Many at times the diffuse sound field exciting the structure is defined by sound pressure level (SPL) values in octave bands. The **Load PSD from SPL** button opens up a new window that allows users to specify the diffuse sound field in terms of octave band SPL that are then converted to PSD values for Coustyx analysis (refer to Figure 7.26).

The frequency values in the table are determined based on the selection of the **Octave Band Order**, and the band frequency range determined by the **Lower Limit** and **Upper Limit**. Note that the **Reference Pressure** should be input in the correct model units. Refer to sections on **Octave Band Order**, **Lower Limit**, **Upper Limit** and **Reference Pressure** for more details. Press the button **Update Frequencies** to update the list of center frequencies in the table to reflect the changes made in the octave band frequency range through **Lower Limit** and **Upper Limit**.

In addition to the manual entry to the table, it can also be modified by loading SPL values from an ascii file using the button **Import data from file**. The file must contain rows corresponding to the number of octave band center frequencies in the table and a single column with SPL data in dB.

**Figure 7.26** Load PSD from SPL window.

The formulae used to convert SPL values in octave bands to PSD values are described below:

Assuming power spectral density to be constant over the entire band, PSD of the diffuse field within a band,  $S_{diff,pp}$ , is evaluated as,

$$S_{diff,pp} = \frac{(p_{rms}^2)_{band}}{(\Delta f)_{band}} = \frac{p_{ref}^2 10^{0.1(SPL)_{band}}}{(\Delta f)_{band}}, \text{ pa}^2/\text{Hz} \quad (7.1)$$

where  $(SPL)_{band}$  is the sound pressure level within the band,  $(p_{rms}^2)_{band}$  is the mean square pressure of the band,  $p_{ref}$  is the reference pressure, and  $(\Delta f)_{band}$  is the bandwidth.

If the data is provided in  $1/n$  th octave band and  $f_c$  is the band center frequency, then

$$(\Delta f)_{band} = \left( 2^{1/2n} - 2^{-1/2n} \right) f_c \quad (7.2)$$



where  $f_s = 2^{-1/2n} f_c$  is the start frequency of the band, and  $f_e = 2^{1/2n} f_c$  is the end frequency of the band.

**Import data from file** Import the frequency dependent input PSD table from an ascii file with components separated by commas, tabs or spaces. The file must contain rows of data for each frequency with two columns specifying frequency (in Hz) and input PSD respectively. 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**.

**Scale factor** All PSD values in the *ASCII* file are multiplied by the Scale factor before being read into the table. 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. A Scale factor of 1 imports the values as they are.

**7.2.4.2. Uncorrelated Plane Waves** The diffuse field in Coustyx is modeled as a superposition of plane waves with random amplitudes directed towards the structure from all equally probable directions (refer to Figure 7.27).

Diffuse field represented as a superposition of field due to  $I$  uncorrelated plane waves with random amplitudes is given by,

$$p_{diff}(\vec{r}, t) = \lim_{I \rightarrow \infty} \sum_{i=1}^I p_i(\vec{r}, t) \quad (7.3)$$

where  $p_i(r, t)$  is field at position  $r$  due to a plane wave  $i$ .

As the field due to plane waves are assumed to be uncorrelated, the power spectral density of the diffuse field is assumed to be the sum of power spectral densities of fields due to each plane wave and is given by,

$$S_{diff,pp} = \sum_{i=1}^I S_{p_i p_i} = \sum_{i=1}^I w_i S_{diff,pp} \quad (7.4)$$

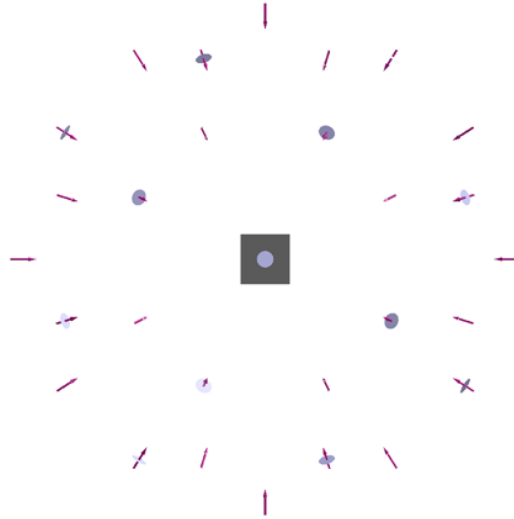
where  $S_{diff,pp}$  is the input PSD of the diffuse sound field,  $S_{p_i p_i}$  is the input PSD due to plane wave  $i$ ,  $w_i$  is the weighting factor (based on the ratio of the solid angle covered by the plane wave  $i$ ). Note that,  $\sum_{i=1}^I w_i = 1$ . For equally probable directions  $w_i = 1/I$  where  $I$  is the number of plane waves.

The plane waves used to represent the diffuse field can be specified through any of the two methods: **User Options** or **Table**.

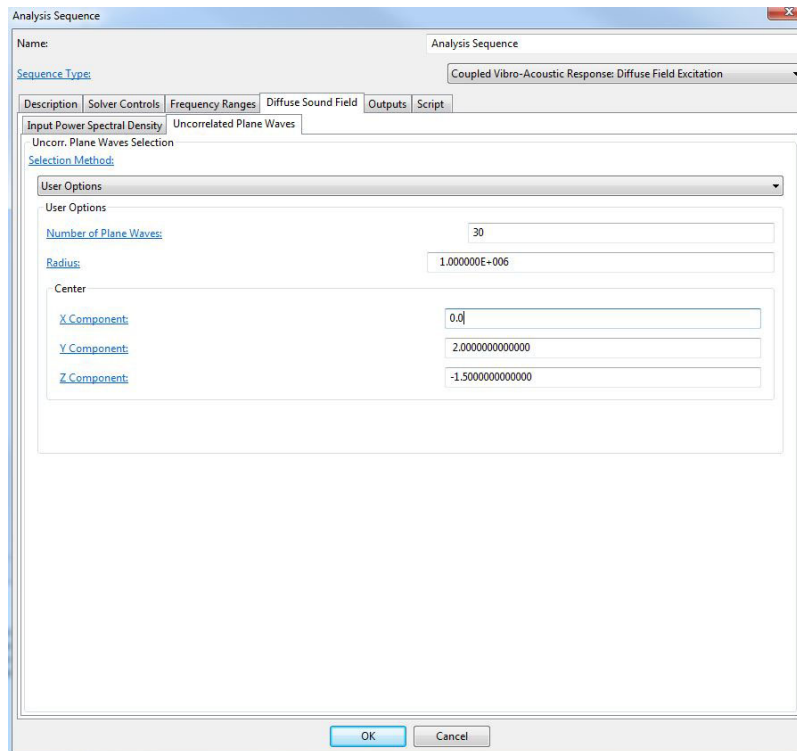
**User Options** Select this method to automatically identify the location, direction and weights of all the plane waves used to represent a diffuse field. The options that are required are (Figure 7.28): **Number of Plane Waves** positioned on a spherical surface with center defined by **X Component**, **Y Component**, and **Z Component**, and radius **Radius**.

Follow the guidelines listed below for a good representation of diffuse field by plane waves:

- Select as many number of plane waves as possible. Note that the amount of time to run the analysis increases linearly with the number of plane waves specified. Need to balance the accuracy requirements with feasible solution run times. Suggested number of plane waves based on the method the plane waves are created internally are, 4, 6, 12, 24, 30, 32, 50, and 72.
- Select the center of the sphere to be the center of your structure.
- Select the radius of the sphere to be a large value when compared to the size of the structure.



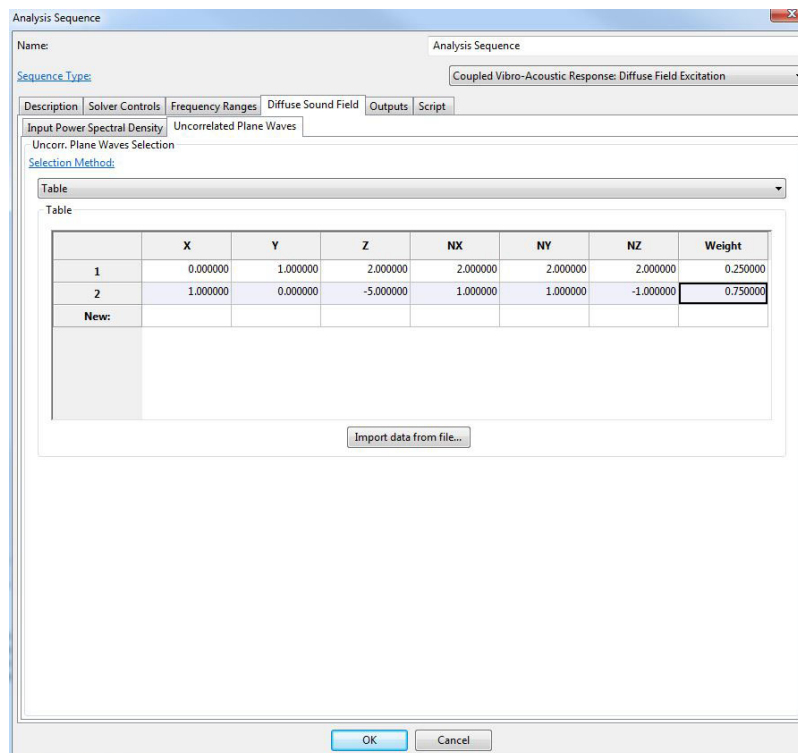
**Figure 7.27** Diffuse sound field modeled as a superposition of uncorrelated plane waves.



**Figure 7.28** Uncorrelated plane waves selection through User Options.

**Table** Select this method to manually enter the positions, directions and weights for all the plane waves used to represent the diffuse field (Figure 7.29). Typically we select the plane waves to be located on a spherical surface with large radius from the center of the structure, with directions pointed radially inwards. Enter the (X,Y,Z) position of the plane wave, (NX,NY,NZ) direction of the plane wave, and their weights. Note that the sum of weights of all entries in the table shall be equal to 1.0.

In addition to the manual entry to the table, it can also be modified by loading data from an ascii file using the button **Import data from file**. The file must contain rows corresponding to the number of plane waves with data in columns for (X,Y,Z) position, (NX,NY,NZ) direction and weight respectively. Make sure the data is in the same units as model units.



**Figure 7.29** Uncorrelated plane waves selection through Table.

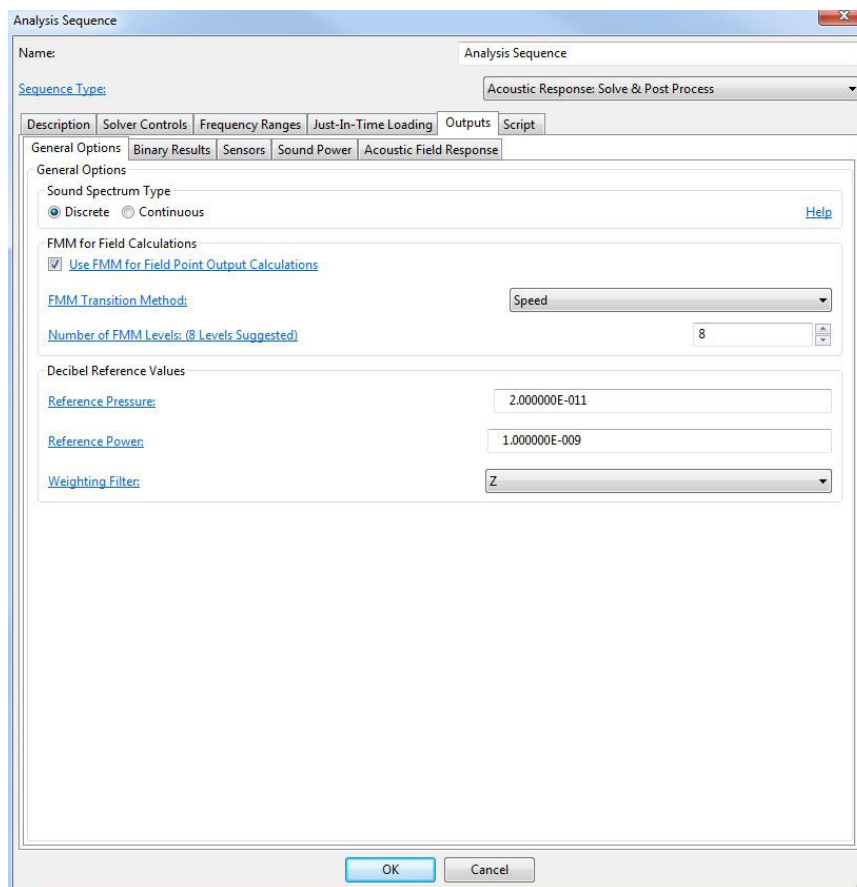
## 7.3. Outputs

The **Outputs** tabs change dependant on the current analysis sequence. It consists of options to save the results to a binary file, save pressure and velocity at sensor locations to an *ASCII* file, save sound power, create an iGlass file for post-processing visualization, create structure frequency response files and more.

### 7.3.1. General Options

The **General Options** tab is shown in Figure 7.30. It allows users to set some common attributes that dictate how various outputs are computed. Set the following in this tabbed window: **Sound Spectrum Type**, whether or not to use **FMM for Field Calculations** or set the **Decibel Reference Values**.

**7.3.1.1. Sound Spectrum Type** Choose between two types of Sound Spectrum; **Discrete** or **Continuous**. Note that the selection of sound spectrum type chiefly effects the computation of overall (sound pressure



**Figure 7.30** General Options window.

or sound power) levels or octave band (sound pressure or sound power) levels. For example, you may get different overall sensor sound pressure level based on your frequency resolution and the type of sound spectrum specified. For a frequency resolution 1 Hz, however, both spectrum types result in the same overall or octave band levels.

**Discrete** When this option is selected the signal is assumed to have a discrete spectrum and the overall levels or octave band levels are computed by discrete dB sum of the response at the discrete analysis frequencies.

$$dB_{band,i} = 10\log_{10} \left( \sum_j \left( \frac{p_{rms}^2}{p_{ref}^2} \right)_{f_j} \right) \quad (7.5)$$

**Continuous** When this option is selected the signal is assumed to have a continuous spectrum but the analysis is assumed to be only run at discrete frequency steps. The total response between any two consecutive frequencies is obtained by integrating responses at these two frequencies using simple trapezoidal rule. Hence the overall or octave band level response for a continuous spectra is calculated as follows:

$$dB_{band,i} = 10\log_{10} \left( \sum_j \left\{ \left( \frac{p_{rms}^2}{p_{ref}^2} \right)_{avg,j} \cdot \Delta f_j \right\} \right) \quad (7.6)$$

$$\left( \frac{p_{rms}^2}{p_{ref}^2} \right)_{avg,j} \approx \frac{1}{2} \left[ \left( \frac{p_{rms}^2}{p_{ref}^2} \right)_{f_j} + \left( \frac{p_{rms}^2}{p_{ref}^2} \right)_{f_{j+1}} \right] \quad (7.7)$$

where  $\Delta f_j = f_{j+1} - f_j$  is the difference in Hz between two consecutive analysis frequencies. Note that when  $\Delta f_j = 1\text{ Hz}$  both the discrete and continuous spectra converge to same results.

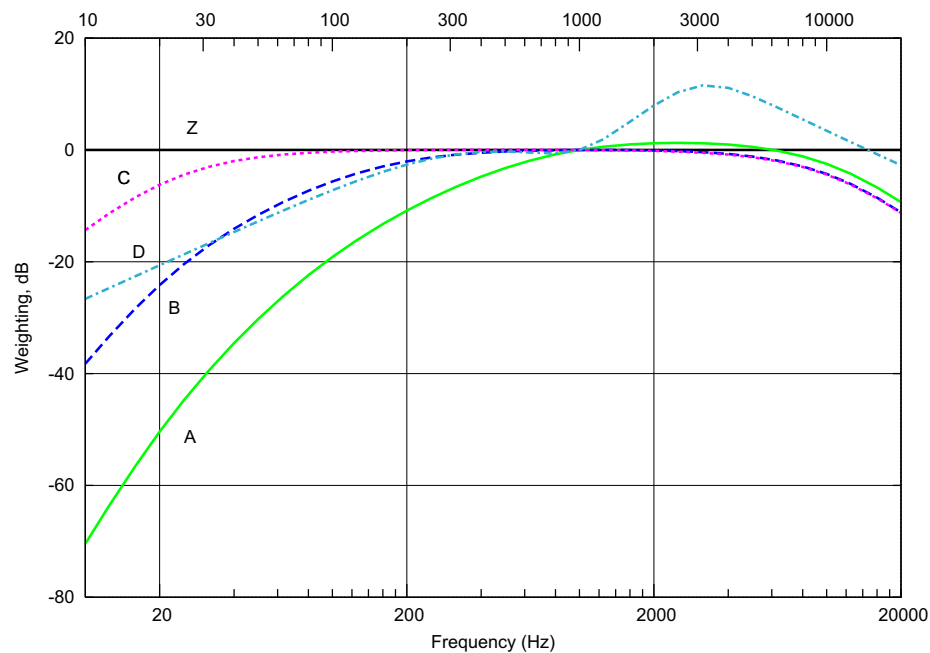
**7.3.1.2. FMM for Field Point Calculations** This option enables the use of FMM to perform field point output calculations. In contrast, the FMM options provided in **Solver Controls** enable only to employ FMM for solving surface solution on the BE mesh. With this option the user can take advantage of FMM even for computing field point outputs. Enable this feature whenever you have large number of sensors, acoustic field point grids (for visualization) or large field mesh used to evaluate sound power.

Refer to section on **FMM** for more details on what each of the options **FMM Transition Method** and **Number of FMM Levels** mean. Keep in mind that the root cell (Level 0) in output calculations not only encloses the BE mesh but also the field points. Choose the **Number of FMM Levels**. The default number of FMM levels suggested will be quite appropriate for most cases, however, note that this number is simply estimated to be one above the suggested number of levels in Solver Controls, as Coustyx does not have any prior knowledge of the field points spread.

**7.3.1.3. Reference Pressure** Choose appropriate reference pressure to be used for decibel calculations. Note that the units of the reference pressure should be consistent with the model units. For example, when model units are in **meter - kilogram - second** (m – kg – s) use the reference pressure value of  $p_{ref} = 2 \times 10^{-5}$ ; where as when the model units are in **millimeter - newton - second** (mm – N – s) use  $p_{ref} = 2 \times 10^{-11}$ .

**7.3.1.4. Reference Power** Choose appropriate reference power to be used for decibel calculations. Note that the units of the reference power should be consistent with the model units. For example, when model units are in **meter - kilogram - second** (m – kg – s) use the reference power value of  $W_{ref} = 10^{-12}$ ; where as when the model units are in **millimeter - newton - second** (mm – N – s) use  $W_{ref} = 10^{-9}$ .

**7.3.1.5. 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.

## 7.3.2. Binary Results

**7.3.2.1. 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 7.30.

**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

### 7.3.3. Sensors

Panel contributions at specified sensor locations could be evaluated using options in this window. Sensors are the field point locations at which pressure and particle velocities are computed. They are specified in the Sensor coordinates table. The coordinates of a sensor along with the domain in which it is present could be entered manually into the table or imported from an *ASCII* file. Figure 7.32. Note that the domain information is needed only for sensors defined in a *Multidomain* model and is not necessary for sensors defined in an *Indirect* model.

**7.3.3.1. Create an Ascii Sensors File** This option is selected to compute panel contributions to the field point pressure and velocities at sensor locations in *ASCII* output files.

**7.3.3.2. Update Domains** This button appears only for sensors in *Multidomain* models. The domains associated with sensors can be changed separately by selecting a domain name from the drop down menu under the column **Domain**. You can also change the domains associated with a group of sensors (filtered through their indices) through **Update Domains** button. Update Sensors Domain window opens up with the following options when you click on the button. Figure 7.33.

**Sensor Selection by Index** Select the sensor indices whose domains need to be modified using this filter. Enter the lower and upper limits of the sensor indices in the text boxes: **Lower Limit** and **Upper Limit** respectively.

**Select Domain** Choose the domain to be associated with sensors that have indices that lie between the lower and upper limits entered above.

**7.3.3.3. 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 (first) three columns representing X, Y and Z coordinates of a sensor (Figure 7.34). The fourth column that is optional takes a valid string (use double quotes if there are spaces within a name) representing the domain name to which the sensor belongs to. Note that domain names are not read for sensors in *Indirect* models. 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 7.34)

**Scale factor** All the values in the *ASCII* file are multiplied by the Scale factor before being read into the table (Figure 7.34). 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 values in the file from m to mm. A Scale factor of 1 imports the values as they are.

**7.3.3.4. Panel Selection** A panel is a collection of elements that is identified by the name of the set in a specified mesh. A default selection of "\*" for a mesh name or a set name implies that all meshes or all sets in a given mesh would be considered as a single panel. The output data for each panel mentioned in the Panel Selection table is written to a file specified in the FileName box. You can modify the output file name by typing in a new name or by selecting an existing file through the *Browse* button. Multiple panels could be specified in the Panel Selection table. Use buttons **Add** or **Delete** to add a new panel or delete existing one from the table respectively.



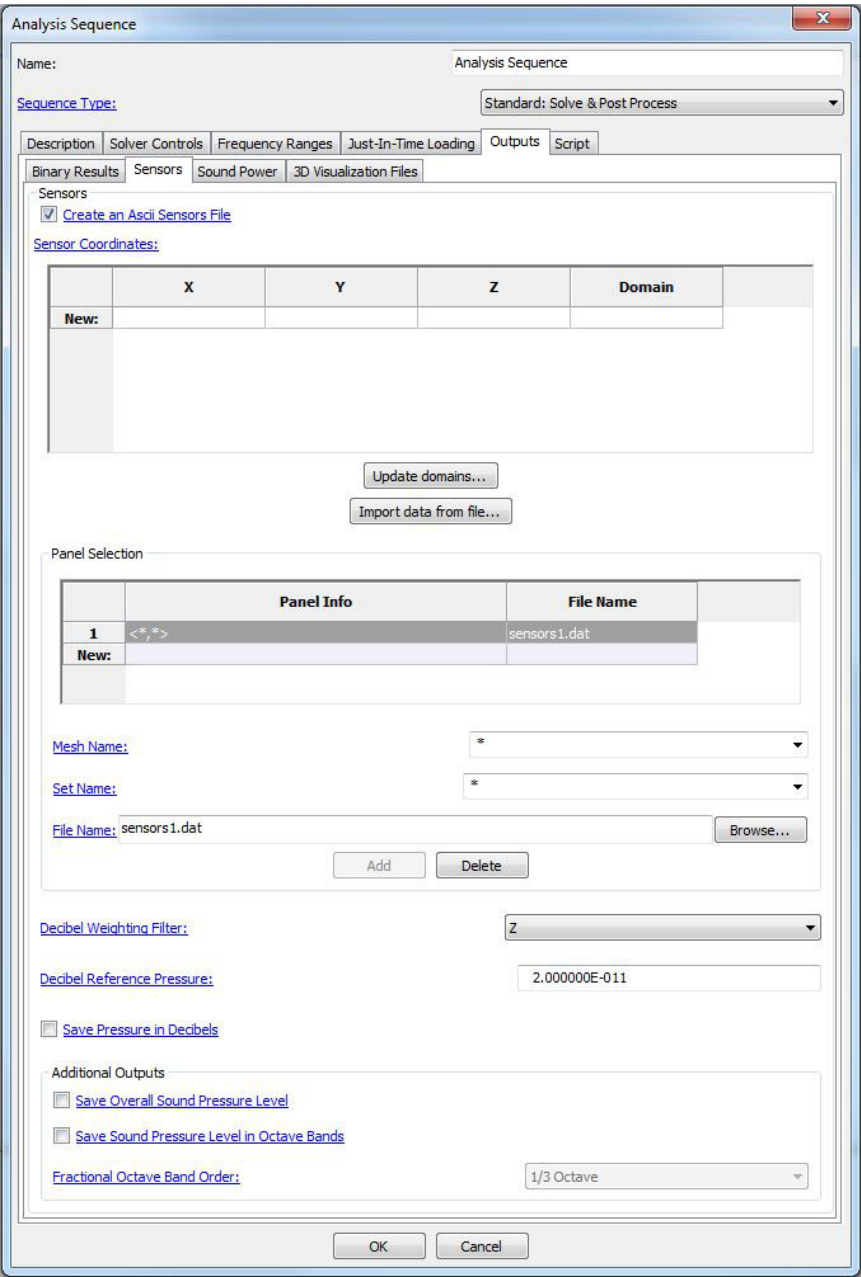


Figure 7.32 Sensor output window.

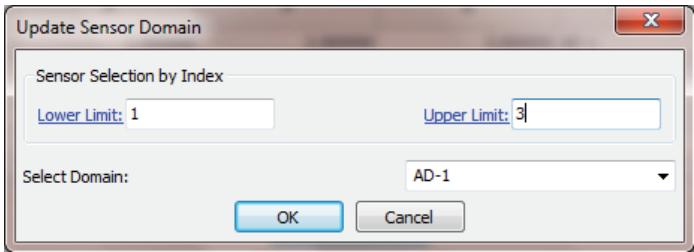
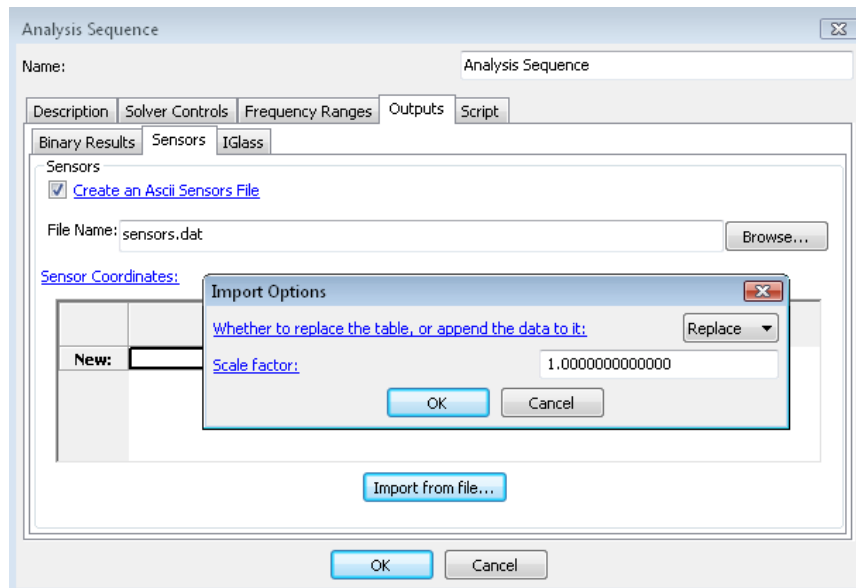


Figure 7.33 Sensor Update Domain window.



**Figure 7.34** Import Sensor Coordinates from a File.

**Mesh Name** Select the name of the mesh that contains the panel. A selection of “\*” for the mesh name forces Coustyx to consider the specified set name from all boundary element meshes.

**Set Name** Select the name of the set that represents the panel. Make sure the panel (set) with desired elements is already created. Specify “\*” to consider all elements in the BE mesh specified by MeshName.

**File Name** Provide the full path of the output file name. Use the **Browse** button to browse folders for the file name.

**7.3.3.5. Save Pressure in Decibels** Select this option to save the field pressure at the sensor location in decibels. The weighting filter and the reference pressure set in **General Options** tabbed window are used for the decibel calculations.

**7.3.3.6. Save Overall Sound Pressure Level** Select this option to save the overall sound pressure level at the sensor location. Note that the sound spectrum type, the reference pressure and the weighting filter set in **General Options** tabbed window effect the calculations. Refer to the formulae described in Equation 7.5 and Equation 7.6.

**7.3.3.7. Save Sound Pressure Level in Octave Band** Select this option to save the sound pressure level in octave bands. Note that the sound spectrum type, the reference pressure and the weighting filter set in **General Options** tabbed window effect the calculations. Refer to the formulae described in Equation 7.5 and Equation 7.6. Choose appropriate fractional octave band order.

**Fractional Octave Band Order** Select the fractional order of the octave band. This dictates the width and center frequency of the band. Refer to **Octave-Band Center Frequencies** for more details.

### 7.3.4. Sound Power

Sound power radiated by the structure could be evaluated using options in this window. Three different evaluation methods are made available to provide greater flexibility to users. They are, **Default**, **Field Mesh**,

and **ISO Standards**. More details on each method is provided in the next sections. Note that domain information is needed to evaluate sound power for *Multidomain* models but is not needed for *Indirect* models.

The output file contains acoustic sound power values computed at each analysis frequency. Based on the options selected in this window, additional outputs are also written. Each column displays the output name for easy reading. Typical outputs include radiated (active) sound power, reactive sound power, input power, and radiation efficiency. Note that the units of power would be consistent with the units chosen for material properties - sound speed, and ambient density. Also, note that some of the outputs mentioned above such as input power or radiation efficiency may not be available for certain combination of options selected in this window.

**7.3.4.1. Create a Sound Power File** This option is selected to compute sound power and save it to an *ASCII* output file.

**7.3.4.2. Domain Name** Select the domain name from the drop down for which you would like to compute the sound power. The domain must have been previously defined to appear in the drop down menu.

**7.3.4.3. Save Power in Decibels** Select this option to save the sound power in decibels. Choose appropriate reference power for the decibel calculations. Note that the sound power level is automatically saved for all weighting filters.

**7.3.4.4. Save Overall Sound Power Level** Select this option to save the overall sound power level. Note that the sound spectrum type, and the reference power set in **General Options** tabbed window effect the calculations. Also, note that the sound power level is automatically saved for all weighting filters. Refer to the formulae in Equation 7.5 and Equation 7.6 for definitions used for discrete and continuous spectrum types.

**7.3.4.5. Save Sound Power Level in Octave Band** Select this option to save the sound power level in octave bands. Note that the sound spectrum type, and the reference power set in **General Options** tabbed window effect the calculations. Also note that the sound power level is automatically saved for all weighting filters. Refer to the formulae in Equation 7.5 and Equation 7.6 for definitions used for discrete and continuous spectrum types. Choose appropriate fractional octave band order.

**Fractional Octave Band Order** Select the fractional order of the octave band. This dictates the width and center frequency of the band. Refer to **Octave-Band Center Frequencies** for more details.

**7.3.4.6. Default** In this method, sound power is computed by integrating normal acoustic intensity over the *boundary element mesh* surface. The surface solution from the acoustic analysis is directly used in above computations and hence no additional post-processing computations are required. This is in contrast with the other two sound power evaluation methods, where acoustic field at some field point locations need to be computed first before evaluating sound power.

The sound power  $W$  through an area  $S$  is given by,

$$W = \int \mathbf{I} \cdot d\mathbf{S} \quad (7.8)$$

where  $\mathbf{I} = P\mathbf{V}$  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 **Complex Intensity**.

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 7.18) over a surface  $S$ , that is,

$$W_a = \int I_a dS \quad (7.9)$$

**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 7.19) over a surface area  $S$ , that is,

$$W_r = \int I_r dS \quad (7.10)$$

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,

$$W = \frac{1}{2} \rho_o c \int v_n^2 dS \quad (7.11)$$

where  $\rho_o$  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,

$$\sigma = \frac{W_a}{W_i} \quad (7.12)$$

### 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

$$P(x, t) = \text{Re} [p(x) \exp(i\phi_p) \exp(-i\omega t)] = p(x) \cos(\phi_p - \omega t) = p_r(x) \cos \omega t + p_i(x) \sin \omega t \quad (7.13)$$

$$V(x, t) = \text{Re} [v(x) \exp(i\phi_v) \exp(-i\omega t)] = v(x) \cos(\phi_v - \omega t) = v_r(x) \cos \omega t + v_i(x) \sin \omega t \quad (7.14)$$

The instantaneous sound intensity is the product of sound pressure and acoustic particle velocity at a point and is given as follows:

$$\begin{aligned} I(x, t) &= P(x, t) \cdot V(x, t) \\ &= p(x)v(x) \cos(\phi_p - \omega t) \cos(\phi_v - \omega t) \\ &= \frac{p(x)v(x)}{2} 2 \cos(\phi_p - \omega t) \cos(\phi_v - \omega t) \\ &= \frac{p(x)v(x)}{2} [\cos(\phi_p - \omega t + \phi_v - \omega t) + \cos(\phi_p - \phi_v)] \\ &= \frac{p(x)v(x)}{2} [\cos(2\phi_p - 2\omega t - \phi_r) + \cos \phi_r] \\ &= \frac{p(x)v(x)}{2} [\cos(2(\phi_p - \omega t)) \cos \phi_r + \sin(2(\phi_p - \omega t)) \sin \phi_r + \cos \phi_r] \\ &= \frac{p(x)v(x)}{2} \cos \phi_r [1 + \cos(2(\phi_p - \omega t))] + \frac{p(x)v(x) \sin \phi_r}{2} \sin(2(\phi_p - \omega t)) \\ &= \frac{p_r(x)v_r(x) + p_i(x)v_i(x)}{2} [1 + \cos(2(\phi_p - \omega t))] + \frac{p_i(x)v_r(x) - p_r(x)v_i(x)}{2} \sin(2(\phi_p - \omega t)) \end{aligned} \quad (7.15)$$

The complex exponential representation of sound intensity is given as

$$\begin{aligned}
 I(x, t) &= \operatorname{Re} \left\{ \tilde{C}(x) [1 + \exp(-2i(-\omega t + \phi_p))] \right\} \\
 &= \operatorname{Re} \left\{ (C_r(x) + iC_i(x)) [1 + \cos(2(\phi_p - \omega t)) - i \sin(2(\phi_p - \omega t))] \right\} \\
 &= C_r(x) [1 + \cos(2(\phi_p - \omega t))] + C_i(x) \sin(2(\phi_p - \omega t))
 \end{aligned} \tag{7.16}$$

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.

$$\begin{aligned}
 \tilde{C}(x) &= C_r(x) + iC_i(x) \\
 &= I_a(x) + iI_r(x)
 \end{aligned} \tag{7.17}$$

Comparing Equation 7.15 with Equation 7.16, we get the expressions for mean active intensity  $I_a(x)$  and amplitude of reactive intensity as follows:

$$I_a(x) = C_r(x) = \frac{p_r(x)v_r(x) + p_i(x)v_i(x)}{2} = \frac{1}{2} \operatorname{Re}(\tilde{p}(x)\tilde{v}^*(x)) \tag{7.18}$$

$$I_r(x) = C_i(x) = \frac{p_i(x)v_r(x) - p_r(x)v_i(x)}{2} = \frac{1}{2} \operatorname{Im}(\tilde{p}(x)\tilde{v}^*(x)) \tag{7.19}$$

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

$$\tilde{C}(x) = \frac{1}{2} \tilde{p}(x)\tilde{v}^*(x) \tag{7.20}$$

**7.3.4.7. Field Mesh** In this method, sound power is computed by integrating normal acoustic intensity over a field mesh. Note that the sound power definitions follow Section 7.3.4.6. The field mesh should already be present in **Structures** model tree member. It should completely enclose the radiating structure and preferably be a surface mesh. Refer Section 5 on how to import a mesh into *Coustyx* model. Choose the structure from the list of structure names provided in the drop-down menu.

#### Structure Name

The name of the structure mesh in **Structures** to be used as the field mesh. This list would be populated only if there are structure meshes in the main model tree member **Model** → **Structures**.

**7.3.4.8. ISO Standards** In this method, sound power is computed from **ISO Standards**. The following standard methods are implemented: **ISO 3744**, **ISO 3745** and **ISO 9614-1**. Figure 7.35 shows the Sound Power Standards outputs window.

#### 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 7.35.

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**, **Parallellepipiped**, **Parallellepipiped against wall**, **Parallellepipiped against corner**.

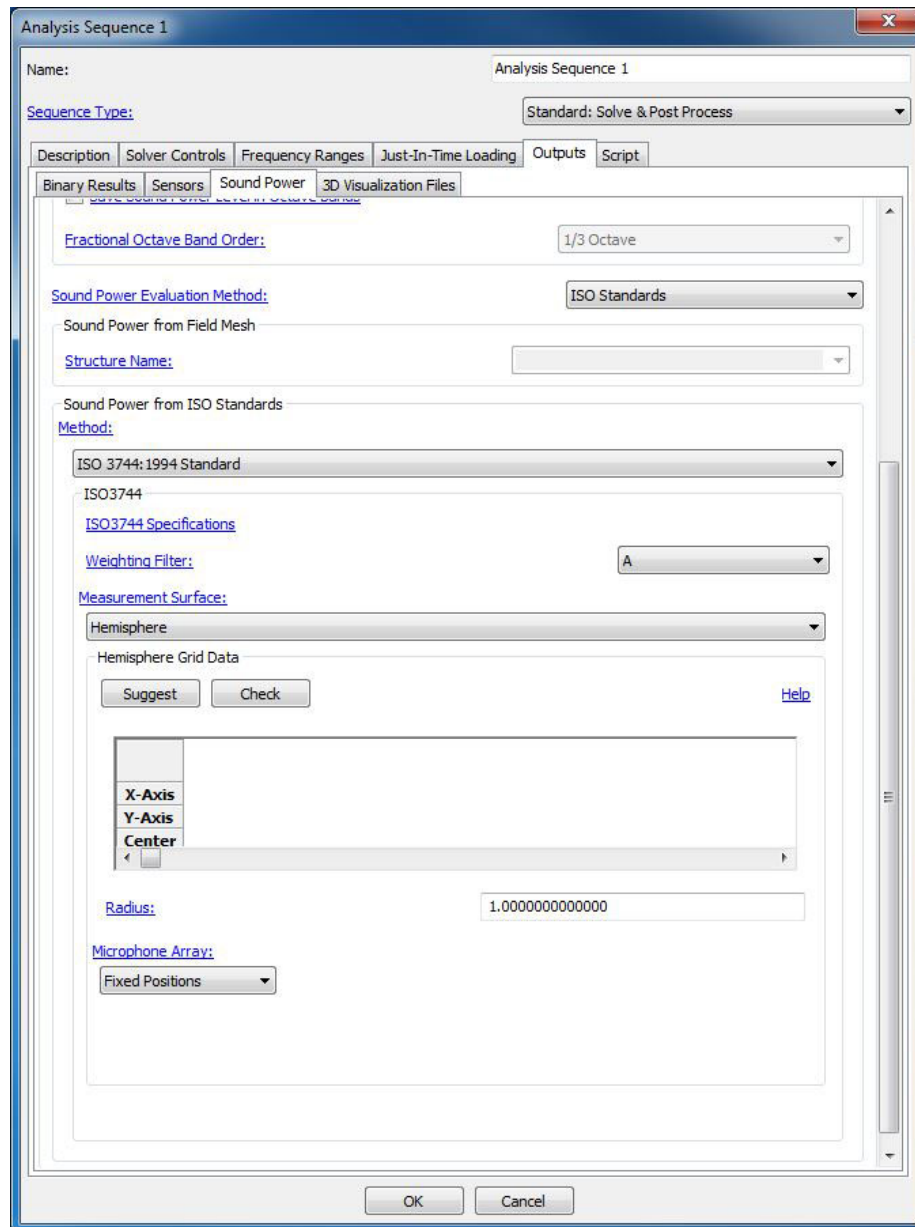


Figure 7.35 ISO3744 sound power standard window.

- **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 7.36 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.

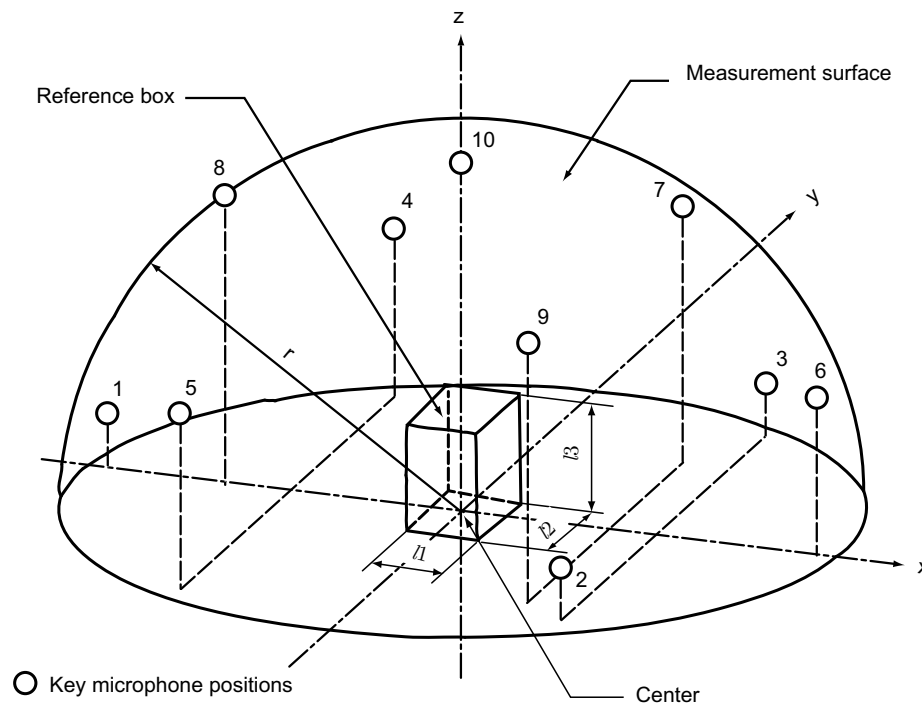
**X-Axis** Specify the orientation of the X-axis here. See Figure 7.36 for definition.

**Y-Axis** Specify the orientation of the Y-axis here. See Figure 7.36 for definition.

**Center** Specify the center of the 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.

**Microphone Array** Choose any of the available types of microphone arrays from the drop-down menu. Figure 7.36 shows a microphone array on the hemisphere.



**Figure 7.36** ISO3744 - Microphone array on the hemisphere.

**Fixed Positions** The microphone positions for this option are listed in Table 7.6 (Figure 7.36). 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<sup>th</sup> position when 19 microphones are selected. The

**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

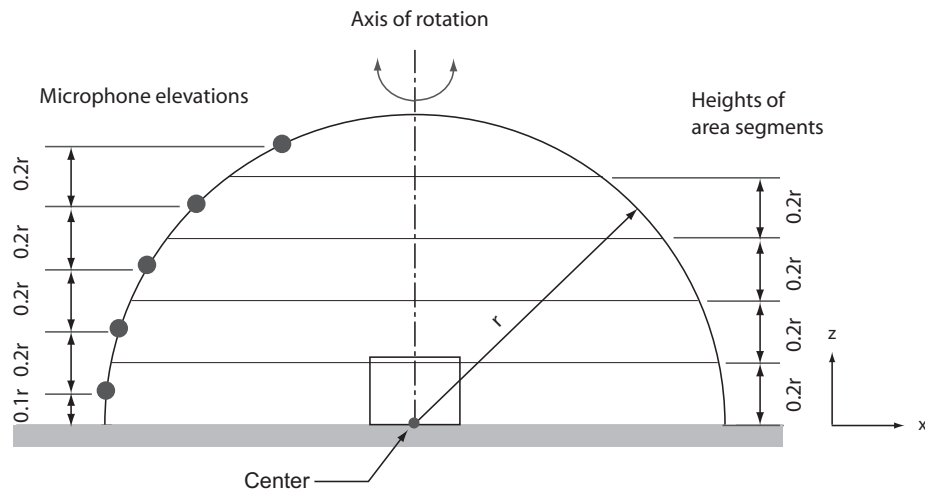
**Table 7.7** ISO3744 - Coordinates of microphone positions for sources emitting discrete tones.

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

partial area associated with the 10<sup>th</sup> position will be twice that of other microphones to account for the omission of 20<sup>th</sup> 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.





**Figure 7.37** Coaxial circular paths in parallel planes for microphone traverses over a reflecting plane.

**Coaxial Circular Paths** Figure 7.37 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. The least number is five.

**Quadrant** Use quadrant measurement surface when the source under test is in front of a wall. Figure 7.38 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 7.38 for definition.

**Y-Axis** Specify the orientation of the Y-axis here. See Figure 7.38 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.

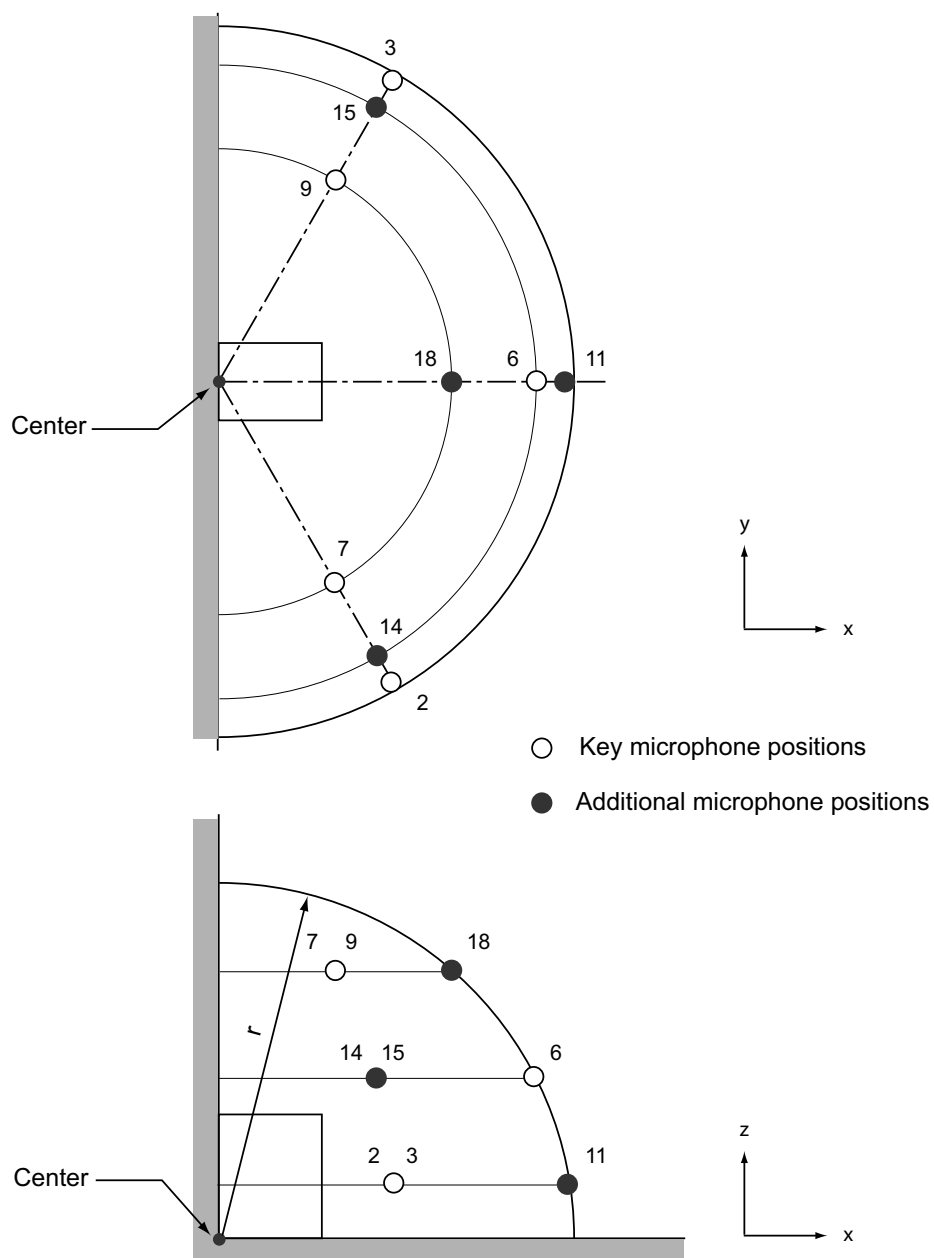
**Microphone Array** Figure 7.38 shows a microphone array on the quadrant surface. The microphone positions are listed in Table 7.6.

**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 7.39 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 7.39 for definition.

**Y-Axis** Specify the orientation of the Y-axis here. See Figure 7.39 for definition.

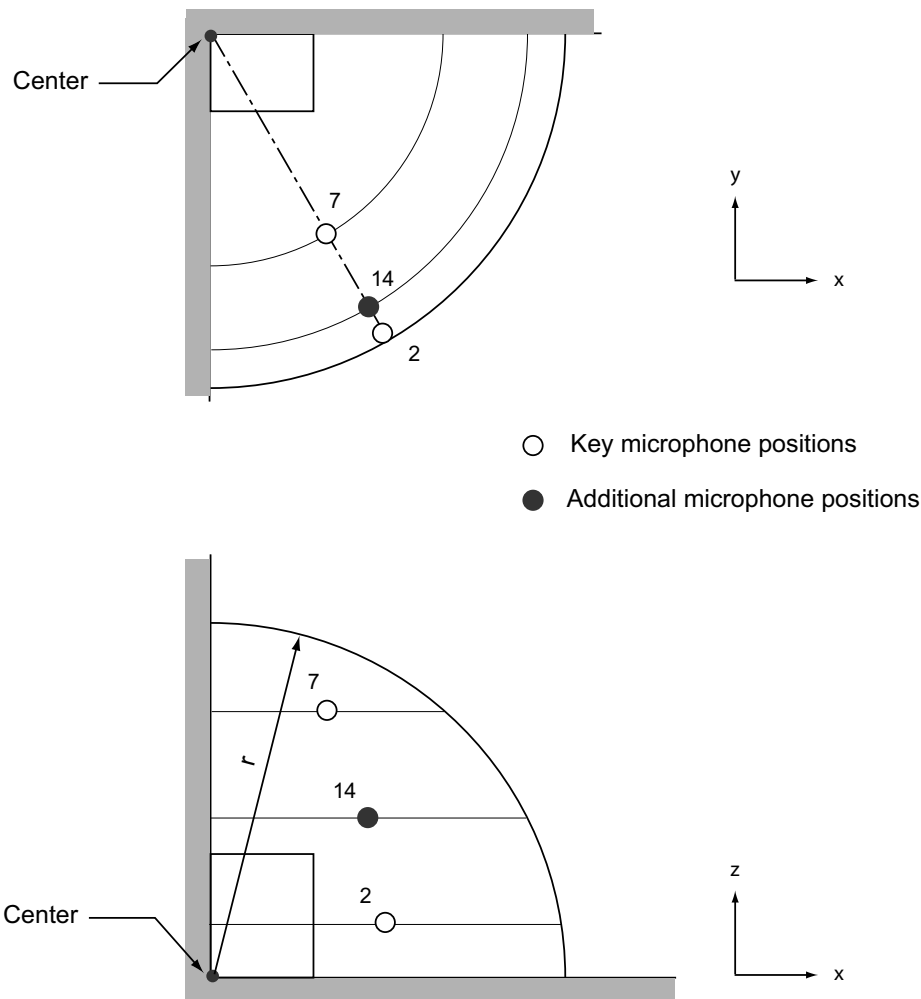


**Figure 7.38** ISO3744 - Microphone array on the quadrant.

**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 7.39 shows a microphone array on the octant surface. The microphone positions are listed in Table 7.6.

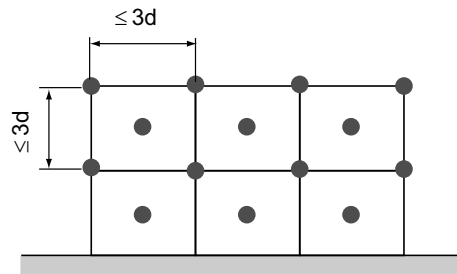


**Figure 7.39** ISO3744 - Microphone array on the octant.

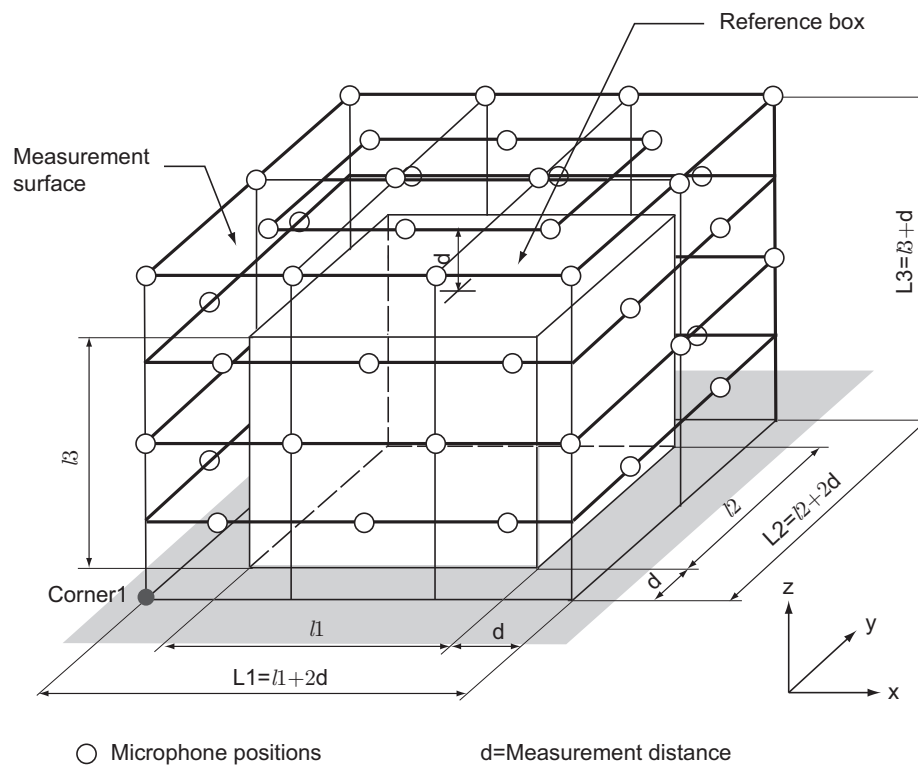
**Number of Probe Positions** Specify the number of microphones associated with equal areas spread over the octant surface. Choose the number of microphones to be 2 or 3 from the drop-down menu.

**Parallelepiped** Use parallelepiped measurement surface when there is only one reflecting plane. Figure 7.41 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 7.41 for definition.



**Figure 7.40** ISO3744 - Procedure for fixing microphone positions on a parallelepiped measurement surface.



**Figure 7.41** ISO3744 - Parallelepiped measurement surface with microphone positions.

**Y-Axis** Specify the orientation of the Y-axis here. See Figure 7.41 for definition.

**Corner1** Specify Corner1 of the parallelepiped surface here. See Figure 7.41 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 7.41). 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  $d = 1\text{ m}$ .

**L2** Length of the parallelepiped in Y-direction (Figure 7.41). 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  $d = 1\text{ m}$ .

**L3** Length of the parallelepiped in Z-direction (Figure 7.41). 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  $d = 1\text{ m}$ .

**N1** Number of subdivisions in X-direction (Figure 7.41). 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 7.40. 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 7.41). 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 7.40. 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 7.41). 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 7.40. 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 7.42 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 7.42 for definition.

**Y-Axis** Specify the orientation of the Y-axis here. See Figure 7.42 for definition.

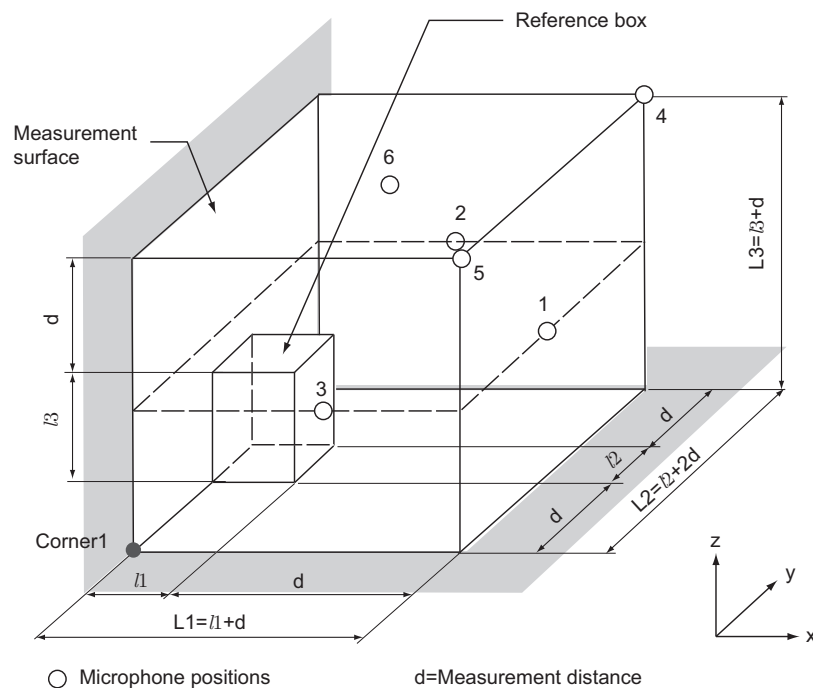
**Corner1** Specify Corner1 of the parallelepiped surface here. See Figure 7.42 for definition. The parallelepiped is constructed using Corner1 as the starting point and  $L1$ ,  $L2$ ,  $L3$  as its dimensions along X, Y, Z-axis respectively.

**L1** Length of the parallelepiped in X-direction (Figure 7.42). 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  $d = 1\text{ m}$ .

**L2** Length of the parallelepiped in Y-direction (Figure 7.42). 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  $d = 1\text{ m}$ .

**L3** Length of the parallelepiped in Z-direction (Figure 7.42). 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  $d = 1\text{ m}$ .

- N1** Number of subdivisions in X-direction (Figure 7.42). 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 7.40. 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 7.42). 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 7.40. 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 7.42). 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 7.40. 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).

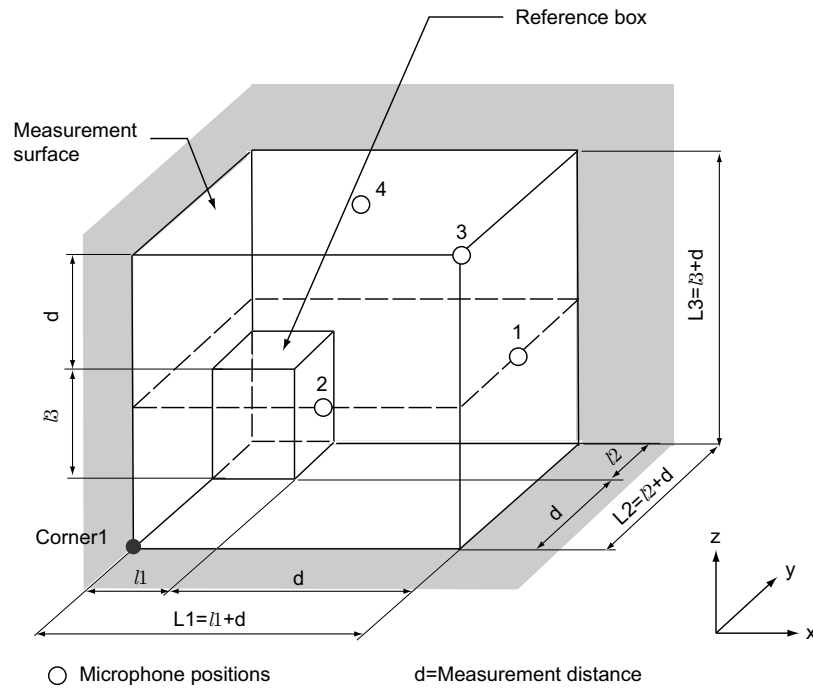


**Figure 7.42** ISO3744 - Parallelepiped measurement surface with microphone positions for a source placed on the floor against a wall.

**Parallelepiped against corner** Use this measurement surface when the source under test is placed on a floor against two walls. Figure 7.43 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 7.43 for definition.

**Y-Axis** Specify the orientation of the Y-axis here. See Figure 7.43 for definition.



**Figure 7.43** ISO3744 - Parallelepiped measurement surface with microphone positions for a source placed on the floor against two walls.

**Corner1** Specify Corner1 of the parallelepiped surface here. See Figure 7.43 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 7.43). 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  $d = 1\text{ m}$ .

**L2** Length of the parallelepiped in Y-direction (Figure 7.43). 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  $d = 1\text{ m}$ .

**L3** Length of the parallelepiped in Z-direction (Figure 7.43). 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  $d = 1\text{ m}$ .

**N1** Number of subdivisions in X-direction (Figure 7.43). 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 7.40. 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 7.43). 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 7.40. 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 7.43). 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 7.40.

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  $p_0 = 2 \times 10^{-5} \text{ Pa}$ .

## 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 7.44.

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 7.36. 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 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.



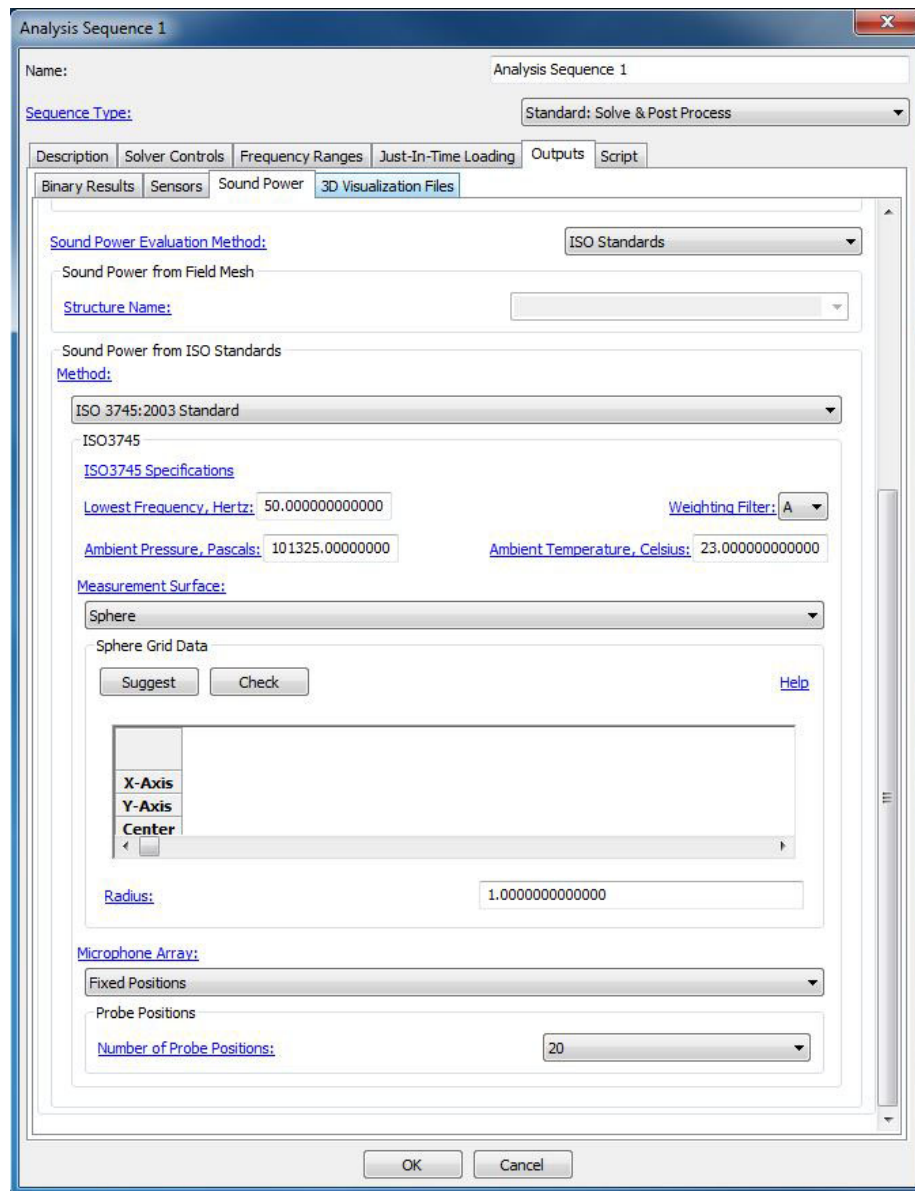


Figure 7.44 ISO3745 sound power standard window.

**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

**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.

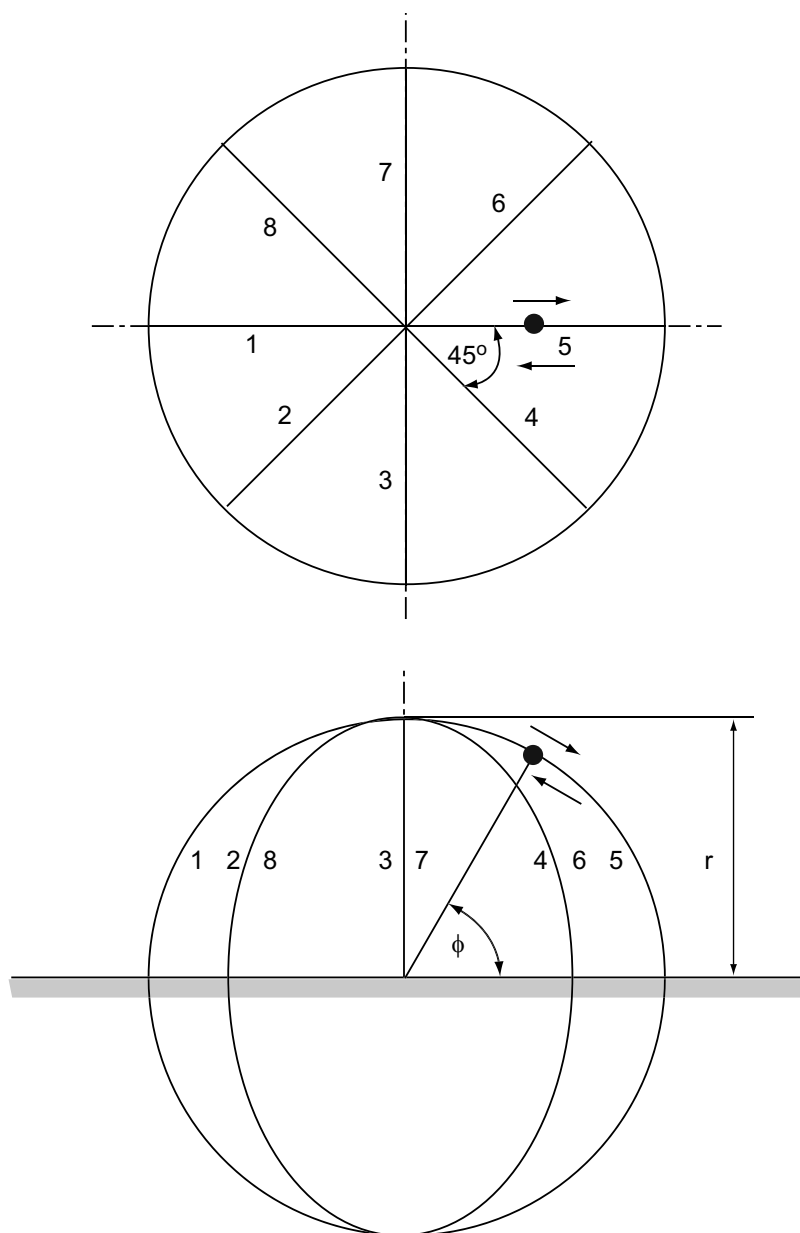
**Fixed Positions** Choose this option to specify an array of fixed microphone positions associated with equal partial areas distributed over the sphere surface.

**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° about the Z-axis.

**Coaxial Circular Paths** Figure 7.37 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.

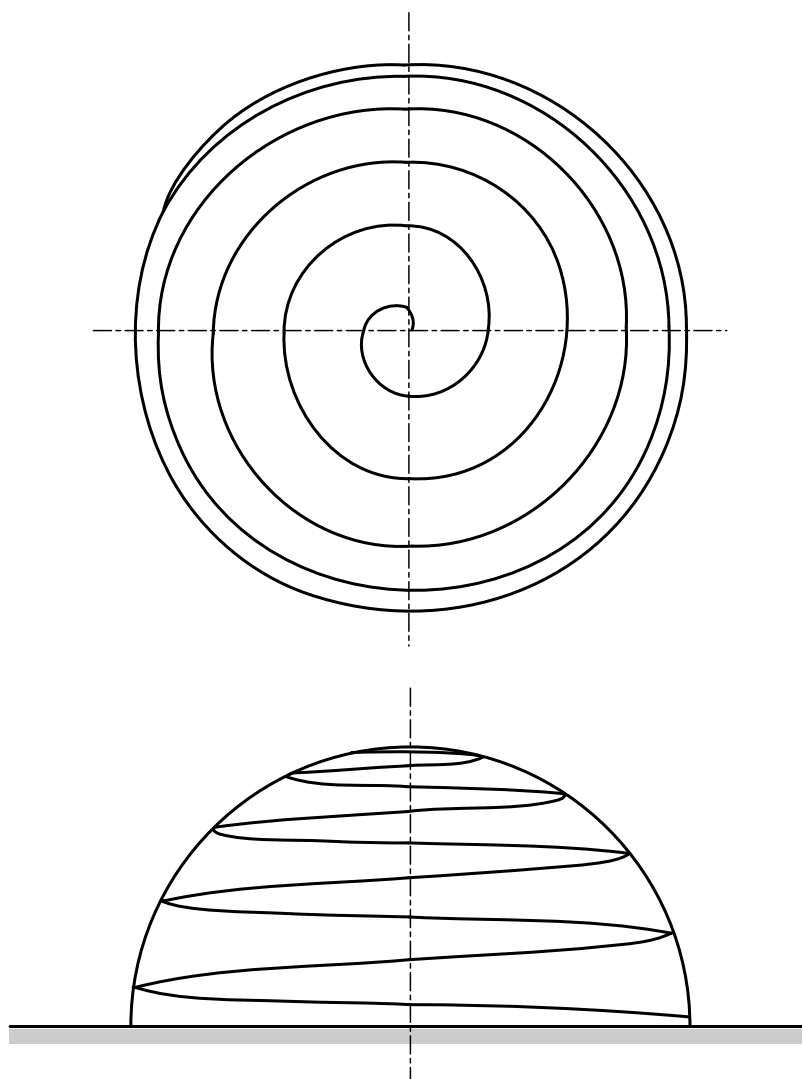
**Number of Circular Paths Traversed by a Probe** Enter the number of circular paths traversed by a microphone on each half of the sphere.

**Meridional Arc Traverses** Figure 7.45 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 7.45** 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 7.45). The least number of microphone traverses is eight.



**Figure 7.46** 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 7.46 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 7.46). 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 7.36. 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

**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

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 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° about the Z-axis.

**Coaxial Circular Paths** Figure 7.37 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 7.45 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.

**Number of Probe Traverses** Enter the number of microphone traverses at equal increments of azimuth angle around the source (Figure 7.45). 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 7.46 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 7.46). 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,  $1.01325 \times 10^5 \text{ Pa}$ ;  $\theta$  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  $p_0 = 2 \times 10^{-5} \text{ Pa}$ .

## 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 7.47.

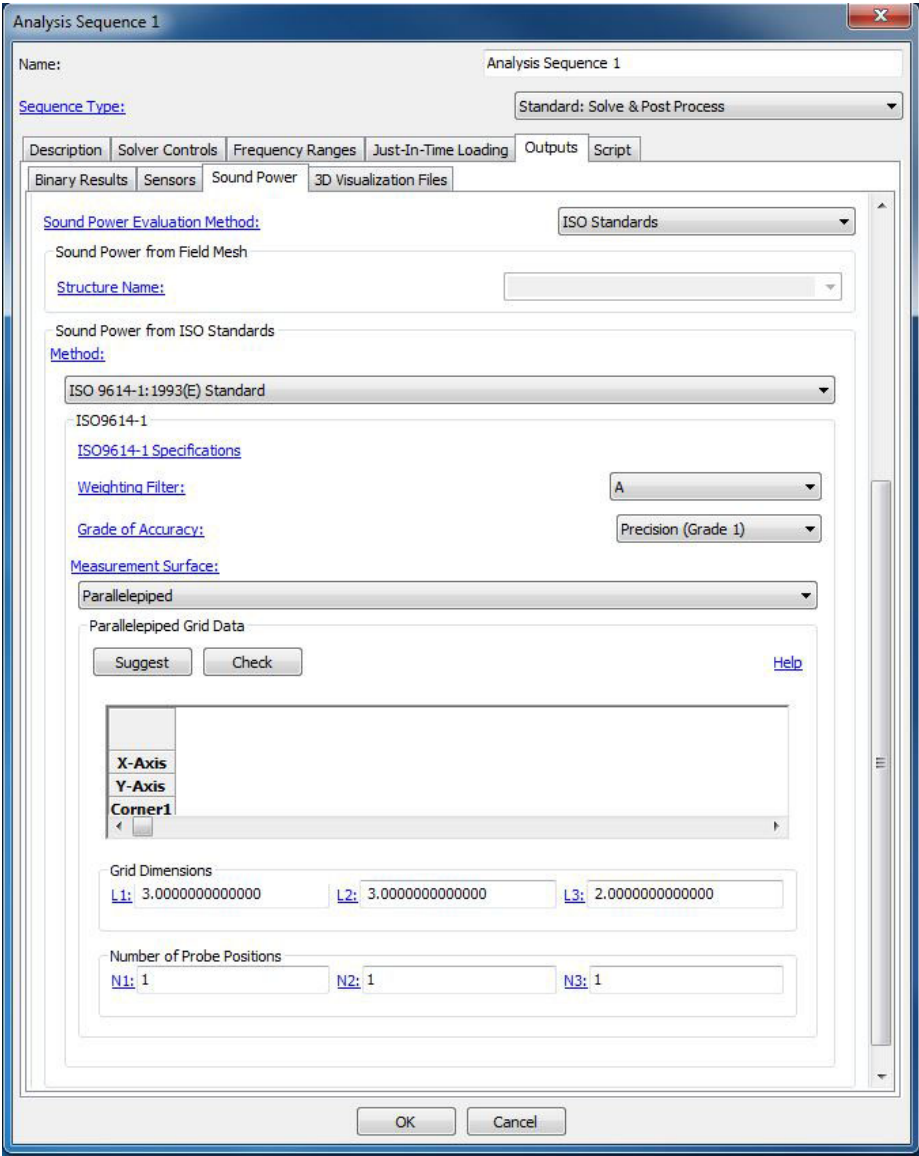


Figure 7.47 ISO9614-1 sound power standard window.

Definitions of some commonly used terms in the standard:

- **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 7.36 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 7.36 for definition.

**Y-Axis** Specify the orientation of the Y-axis here. See Figure 7.36 for definition.

**Center** Specify the center of the 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  $\theta$  direction, and  $2n$  divisions along  $\phi$  direction. Note:  $\theta$  is the angle made by a point on the hemisphere with the Z-axis, and  $\phi$  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 7.48 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.

**X-Axis** Specify the orientation of the X-axis here. See Figure 7.48 for definition.

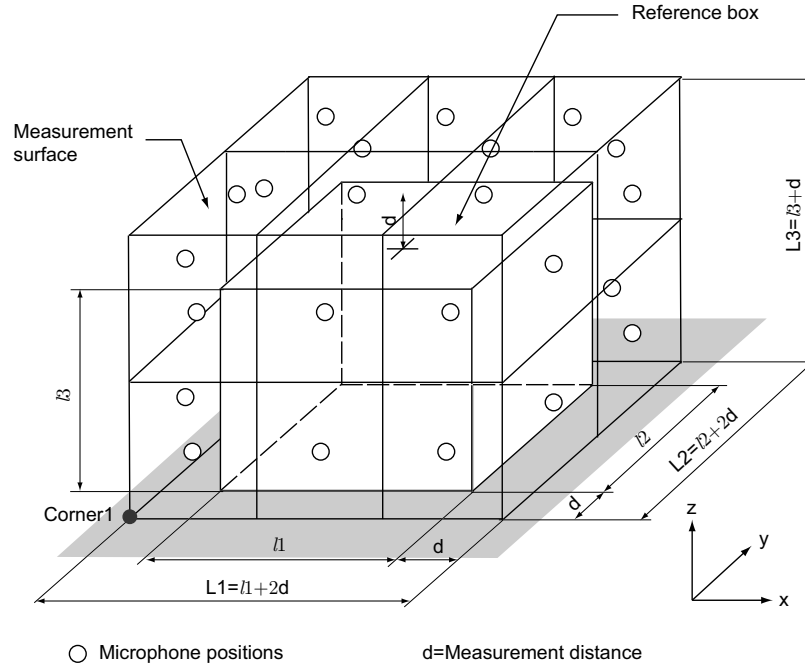
**Y-Axis** Specify the orientation of the Y-axis here. See Figure 7.48 for definition.

**Corner1** Specify Corner1 of the parallelepiped surface here. See Figure 7.48 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 7.48). 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 7.48). 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.





**Figure 7.48** ISO9614-1 - Parallelepiped measurement surface with probe positions.

- L3** Length of the parallelepiped in Z-direction (Figure 7.48). 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  $d = 1\text{ m}$ .
- N1** Number of subdivisions in X-direction (Figure 7.48). 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 7.48). 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 7.48). 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:

$$L_w = 10 \log_{10} \left( \sum_{i=1}^N \frac{P_i}{P_0} \right) \text{ dB} + W$$

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  $i$ .

### 7.3.5. Muffler Transmission Loss

This window enables users to evaluate transmission loss of a muffler or an intake manifold using the *three-point* method. Note that the user still has to setup the rest of the model appropriately to perform this analysis correctly. For example, the inlet should be applied with a unit uniform normal velocity and the outlet should be applied with an anechoic termination while the rest of the mesh is applied with a zero velocity boundary condition.

In a three-point method, the transmission loss is evaluated from the field pressures measured at three microphone locations inside the muffler. Among the three microphones, two of them (microphone 1, and 2) are located in the inlet pipe and one (microphone 3) in the outlet pipe (refer to Figure 7.49). The two microphones in the inlet pipe are used to extract the incoming wave pressure ( $p_i$ ). The field point pressure at microphone 3 is the same as the transmitted wave pressure ( $p_t$ ) in the outlet pipe, that is,  $p_3 = p_t$ . This is due to the specification of anechoic termination at the outlet, which by definition doesn't reflect waves back into the outlet pipe.

The incoming wave  $p_i$  extracted from the two inlet pipe measurements is given by

$$p_i = -\frac{1}{2i \sin k(x_2 - x_1)} [p_1 e^{-ikx_2} - p_2 e^{-ikx_1}] \quad (7.21)$$

where  $\sin k(x_2 - x_1) \neq 0$  or  $k(x_2 - x_1) \neq n\pi$ ,  $n = 0, 1, 2, \dots$ . Note that the spacing between the microphones 1 and 2 should be carefully chosen to satisfy this condition at all frequencies.

The transmission loss is evaluated from the incoming ( $p_i$ ) and the transmitted ( $p_t = p_3$ ) wave pressures as,

$$TL = 20 \log_{10} \left\{ \frac{|p_i|}{|p_3|} \right\} + 10 \log_{10} \left( \frac{S_i}{S_o} \right) \quad (7.22)$$

where  $S_i$ , and  $S_o$  are the inlet and outlet tube areas respectively.

Provide the following inputs to evaluate transmission loss.

**7.3.5.1. File Name** Provide the full path for the transmission loss output file name. Use the **Browse** button to browse folders for the file name.

#### 7.3.5.2. Inlet Pipe

**Mic 1** Define the position (**X,Y,Z**) of the microphone 1 placed in the inlet pipe. Also, enter the distance from the inlet to the microphone 1 position (**X1**).

**Mic 2** Define the position (**X,Y,Z**) of the microphone 2 placed in the inlet pipe. Also, enter the distance from the inlet to the microphone 2 position (**X2**).

**Domain Name** Select the domain name in which the inlet microphones are placed from the drop down menu. Note that domains must have been already defined to appear in the drop down menu.

#### 7.3.5.3. Outlet Pipe

**Mic 3** Define the position (**X,Y,Z**) of the microphone 3 placed in the outlet pipe.

**Domain Name** Select the domain name in which the outlet microphone is placed from the drop down menu. Note that domains must have been already defined to appear in the drop down menu.

**7.3.5.4. Si/So** Enter the ratio of inlet to the outlet pipe cross-section areas to be used in the transmission loss calculations.

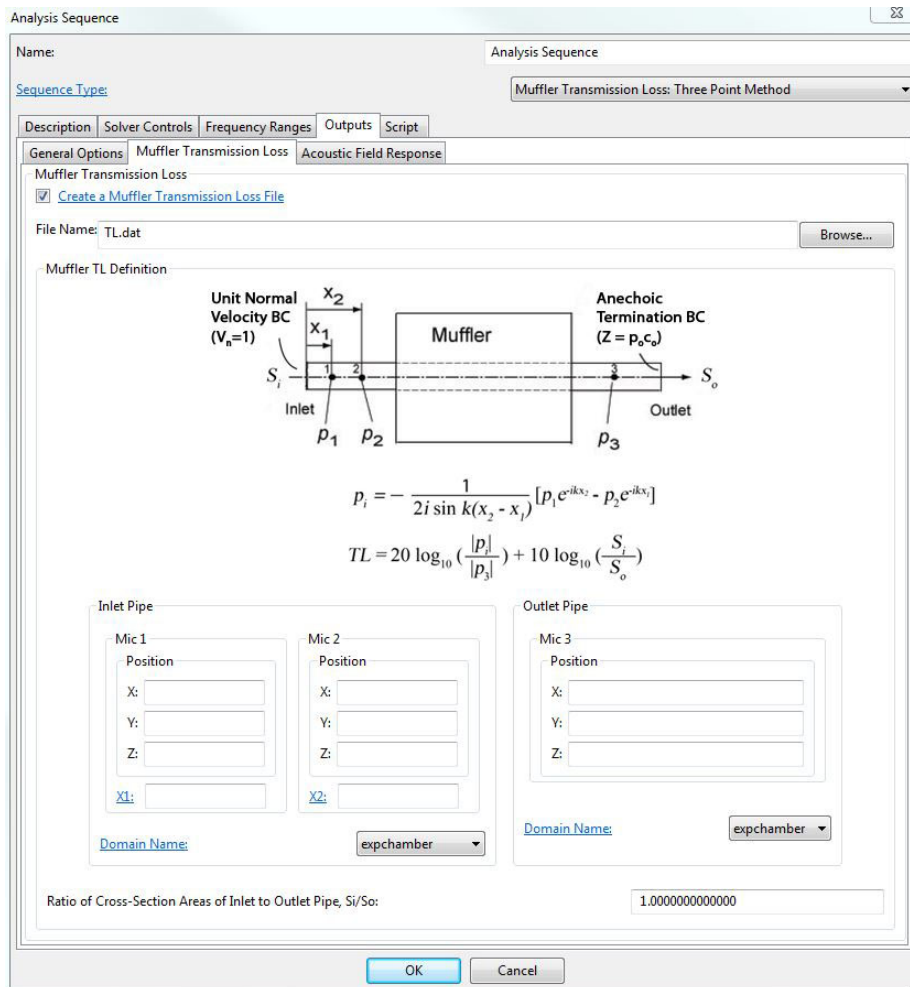


Figure 7.49 Muffler Transmission Loss window.

### 7.3.6. Acoustic Field Response

This window provides options to save acoustic field results for 3-d post-processing visualization through the two tabbed windows: **Response Files** and **Field Point Grids**. Figure 7.50. The output file format options that are currently offered are: (a) IGlass file, (b) Abaqus Odb file, and (c) Hyperworks H3D file. IGlass files are data files that can be read by *Coustyx* IGlass Viewer. Abaqus Odb files can be read by Abaqus CAE/Odb Viewer, where as Hyperworks H3D files can be read by Hyperview. Make sure the corresponding API dlls for Abaqus Odb and Hyperworks H3D are installed on your system when you choose to save *Coustyx* results to these file formats.

In addition to the generation of acoustic output data for the boundary element mesh, the user has the option to create different field point grids through the tabbed window: **Field Point Grids**, and add them to the output file to visualize acoustic variables at the field points.

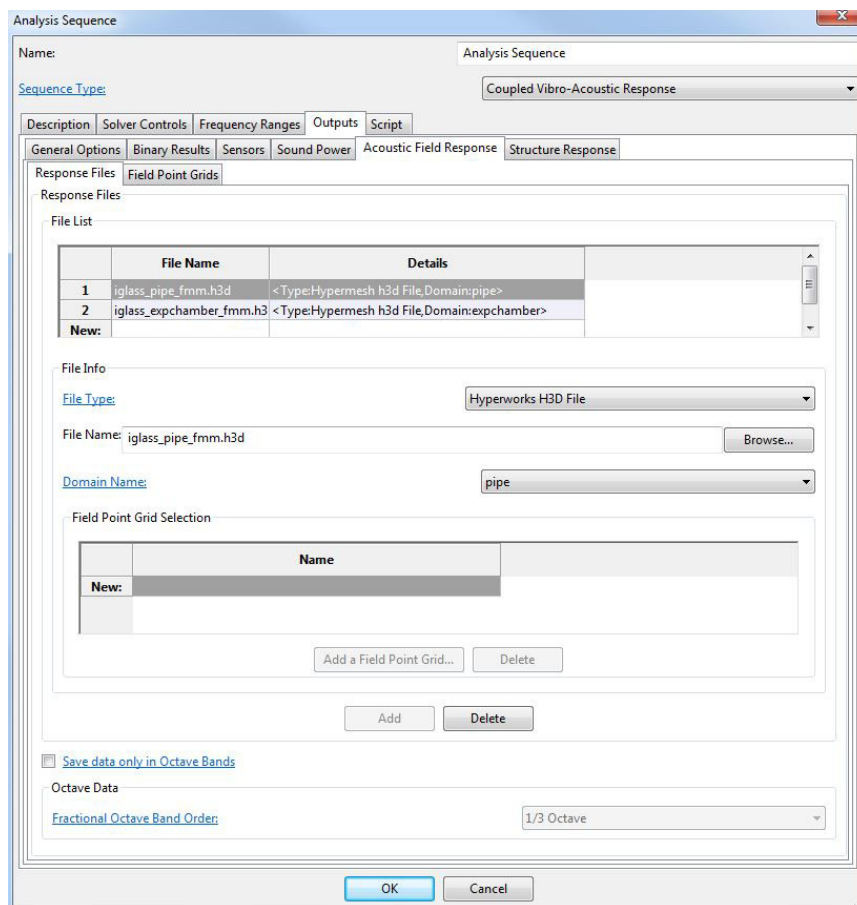


Figure 7.50 Acoustic Field Response window.

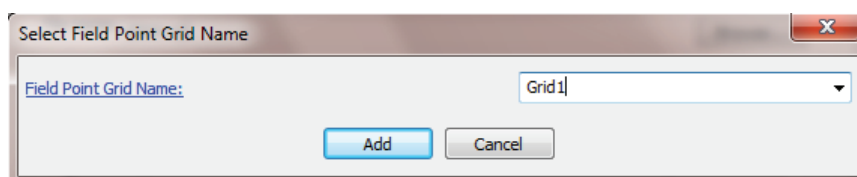


Figure 7.51 Select field point grids window.

### 7.3.6.1. Response Files

**File List** Multiple visualization output files could be saved by adding file details to the list in the table. Use the button **Add** or **Delete** to update the File List table. For each file the user can choose different options described in **File Info**.

**File Type** Choose the desired file type to write the visualization data into. The output file formats that are currently supported are: (1) IGlass file, (2) Abaqus Odb file, and (3) Hyperworks H3D file. Appropriate API dlls have to be installed on your machine before selecting the file types: Abaqus Odb file or Hyperworks H3D file. Follow the instructions in the error message in case Coustyx fails to find these dlls.

**File Name** Enter the file name here. Use the button **Browse...** to enter an existing file name or path. It is recommended to add the default extensions for each file types in the file name. Example: \*.cgl for IGlass files, \*.odb for Abaqus Odb files, and \*.h3d for Hyperworks H3D files.

**Domain Name** This option is applicable only for *Multidomain* models. Select the domain of interest from the drop down menu. Only data related to this domain is saved to the visualization file. For models with multiple domains, visualization files need to be saved for each domain separately. Also note that the field point grids that are selected in the table: [Field Point Grid Selection](#) must physically belong to this domain as well.

**Field Point Grid Selection** Add field points grids defined in the tabbed window: [Field Point Grids](#) to the current file. In addition to the data for the boundary element mesh, visualization data for the selected field point grids are also computed and added to this file. Click on the button **Add a Field Point Grid...** that opens up the dialog window **Select Field Point Grid Name** (Figure 7.51). Only the names of the field point grids already defined appear in the drop-down menu: **Field Point Grid Name**. Choose the desired field point grid and click OK to add. Use **Delete** button to remove the current selection from the Field Point Grid Selection table.

**Save data only in Octave Bands** Select this option to save the output data only at octave band center frequencies. The energy in a band is summed up (assuming discrete spectra or continuous spectra from the options selected in [General Options](#) tabbed window) to obtain a single representative value for the entire band at its center frequency.

For example, for a discrete spectrum signal the octave band root mean square (RMS) pressure ( $p_{b,rms}$ ) is evaluated from the RMS pressures computed at the narrow band analysis frequencies ( $p_{i,rms}$ ) within that band, that is,

$$p_{b,rms}^2 = \sum_i p_{i,rms}^2$$

**Fractional Octave Band Order** Select the fractional order of the octave band. This dictates the width and center frequency of the band. Refer to [Octave-Band Center Frequencies](#) for more details.

**7.3.6.2. Field Point Grids** *Coustyx* offers a variety of field point grid options to visualize acoustic field generated by the boundary element solution. Figure 7.52.

**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 pressure, particle velocity, time-averaged intensity are some of the acoustic variables computed at these points. Figure 7.53 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 7.53).

Analysis Sequence

Name: Analysis Sequence

Sequence Type: Acoustic Response: Solve & Post Process

Description Solver Controls Frequency Ranges Just-In-Time Loading Outputs Script

General Options Binary Results Sensors Sound Power Acoustic Field Response

Response Files Field Point Grids

Field Point Grids

	Name	Type
New:		

Field Mesh Name: NewFieldPointGrid

Type: Quadrilateral

Quadrilateral Grid Data

	X	Y	Z
Corner 1	0.000000	0.000000	0.000000
Corner 2	0.000000	0.000000	0.000000
Corner 3	0.000000	0.000000	0.000000

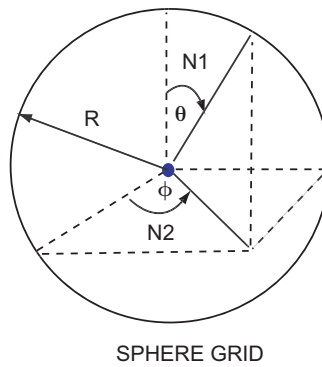
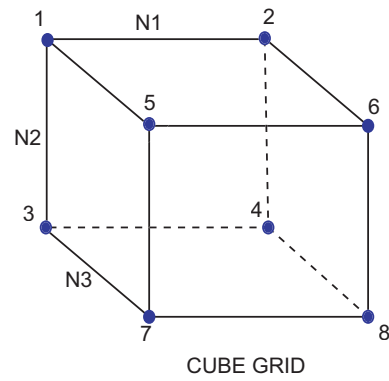
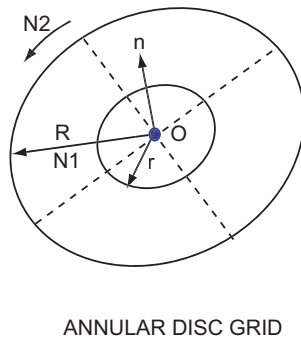
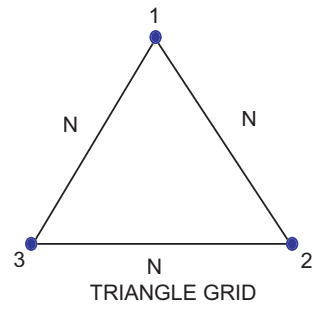
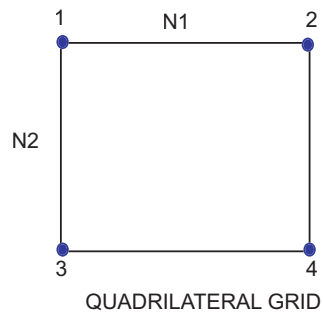
No. of divisions N1: 1

No. of divisions N2: 1

Add Delete

OK Cancel

Figure 7.52 Field Point Grids tab.



**Figure 7.53** IGlass field point grid types - Quadrilateral, Triangle, Annular Disc, Box, and Sphere.

**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 7.53 for quadrilateral grid.

**No. of divisions N1** The number of divisions in the direction connecting **Corner 1** to **Corner 2** in a quadrilateral grid (refer to Figure 7.53). The grid defined by the four corners is divided into  $N1 \times N2$  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 7.53). The grid defined by the four corners is divided into  $N1 \times N2$  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.

**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.

**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.

**Center** The coordinates of the center of the annular disc are set using this option.

**Normal** The normal to the plane of the annular disc is set using this option.

**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.

**Outer Radius** This option sets the outer radius of the annular disc. Any value greater than the inner radius is valid.

**No. of divisions in radial direction** The number of divisions in the radial direction of the annular disc grid (refer to Figure 7.53).

**No. of divisions in circular direction** The number of divisions in the circular direction of the annular disc grid (refer to Figure 7.53).

**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 7.53 for the box grid.

**No. of divisions N1** The number of divisions in the direction connecting **Corner 1** to **Corner 2** (refer to Figure 7.53). The box grid surface is divided into  $N1 \times N2 \times N3$  divisions.

**No. of divisions N2** The number of divisions in the direction connecting **Corner 1** to **Corner 3** (refer to Figure 7.53). The box grid surface is divided into  $N1 \times N2 \times N3$  divisions.

**No. of divisions N3** The number of divisions in the direction connecting **Corner 1** to **Corner 5** (refer to Figure 7.53). The box grid surface is divided into  $N1 \times N2 \times N3$  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  $\theta$  and  $\phi$  directions.

**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  $\theta$  direction (refer to Figure 7.53).

**No. of divisions in azimuth angle direction** The number of divisions in  $\phi$  direction (refer to Figure 7.53).

**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 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.

**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**.



### 7.3.6.3. Output Variables

- Run acoustic analysis and create output visualization file in either of the three allowed file formats: IGlass, Abaqus Odb file, or Hyperworks H3D file.
  - For IGlass file: double-click on the IGlass file to open it using IGlass Viewer (Figure 7.54).  
For Abaqus Odb file: open the odb file using Abaqus CAE/Odb Viewer.  
For Hyperworks H3D file: open the h3d file using Hyperview.
  - For IGlass file: 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.  
For Abaqus Odb file: open the odb file in Abaqus CAE/Odb Viewer to visualize the field outputs or the history outputs. In addition to field variables the Odb file consists of history outputs that include radiated sound power, input sound power, radiation efficiency and panel contributions at all sensors.  
For Hyperworks H3D file: open the h3d file in Hyperview to visualize the field outputs.
- Discussed below are the list of attributes that are written to the acoustic response files.

**Multi-Domain Model** For a Multi-Domain model, the attributes include the following:

**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 6.5 for definition).

Surface normals for different types of field point grids provided in Coustyx (Figure 7.53) 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 7.53. 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 7.53. 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.

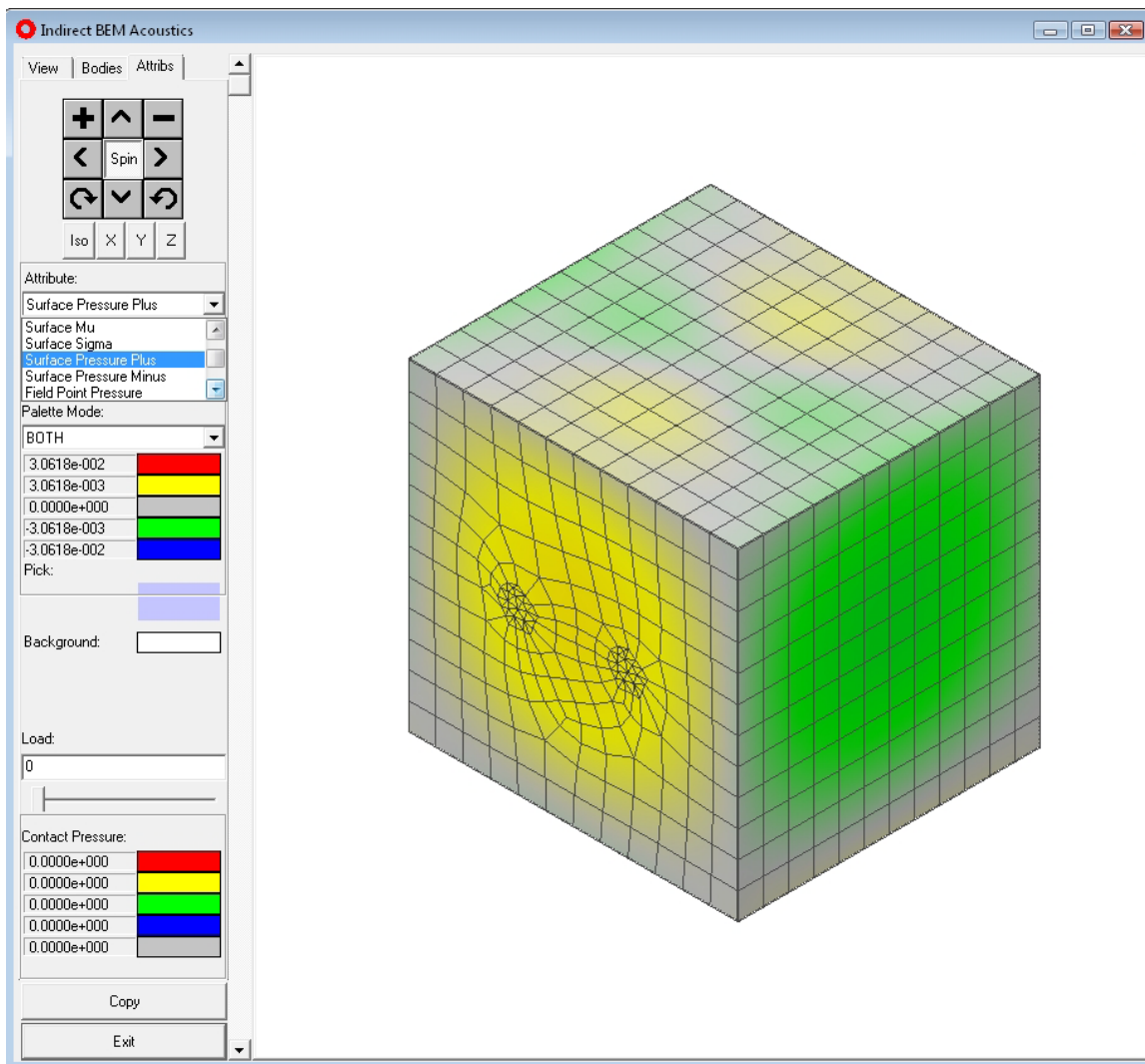
**Sound Intensity** This attribute displays the time averaged sound intensity vector on the boundary element mesh and over field point grids.

**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 attributes include the following:

**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**



**Figure 7.54** IGLASS viewer showing sound pressure distribution on the exterior surface of a housing.

**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 7.53) 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 7.53. 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 7.53. 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 6.23). 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 6.23). 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.

### 7.3.7. Structure Response

This window provides options to save the coupled structure frequency response data to an output file. Refer to Figure 7.55. Note that amplitudes of the response are written out for the sequence type: **Coupled Vibro-Acoustic Response**, whereas auto power spectral densities are written to the output files for the sequence type **Coupled Vibro-Acoustic Response: Diffuse Field Excitation**. The output file formats currently supported are: (a) Abaqus Odb file, (b) Hyperworks H3D file, and (c) Nastran Punch file. Make sure the corresponding API dlls for Abaqus Odb and Hyperworks H3D are installed on your system when you choose to save structure response to these file formats. Multiple output files could be generated by specifying multiple structures and their output files in the **Structure List** table. Use buttons **Add** or **Delete** to add a new structure or delete existing one from the table respectively.

**Structure Name** Select the name of the coupled structure for which you want to write the response output file from the drop-down menu. This list would be populated only if there are structure meshes in the main model tree member **Model** → **Structures**.

**Set Name** Select the name of a set from the chosen structure from the drop-down menu. Only the response at nodes of this set are written to the output file. Make sure the set with desired elements or nodes is already created in the corresponding structure. Specify “\*” to write results at all nodes of the structure.

**Data Type** Select the data type that is output to the file. Choose any of the following: **Nodal Displacement**, **Nodal Velocity**, **Nodal Acceleration**, and **Nodal Force**. Note that the nodal force response written to the file consists of only the acoustic loads and not any external forces applied to the structure. For the sequence type: **Coupled Vibro-Acoustic Response** structure frequency response amplitudes are written out, whereas for the sequence type: **Coupled Vibro-Acoustic Response: Diffuse Field Excitation** auto power spectral densities of nodal responses are written out.

**File Type** Choose the desired file format to write the structure response from the drop-down menu. The output file formats that are currently supported are: (1) Abaqus Odb file, (2) Hyperworks H3D file, and (3) Nastran Punch file. Appropriate API dlls have to be installed on your machine before selecting the file types: Abaqus Odb file or Hyperworks H3D file. Follow the instructions in the error message in case Coustyx fails to find these dlls.

**File Name** Provide the full path of the output file name. Use the **Browse** button to browse folders for the file name.

## 7.4. Script

The window **Script** (Figure 7.56) shows the analysis script that was generated automatically based on the options selected in **Solver Controls**, **Frequency Ranges**, and **Outputs** windows. Coustyx analysis script is a collection of commands that are used to run an analysis.

Advanced users can directly modify the script to suit their needs. To edit an analysis script, select **Model** → **Analysis Sequences** → *<Analysis Sequence>* → **Script**. Open the script in an editor by right-clicking

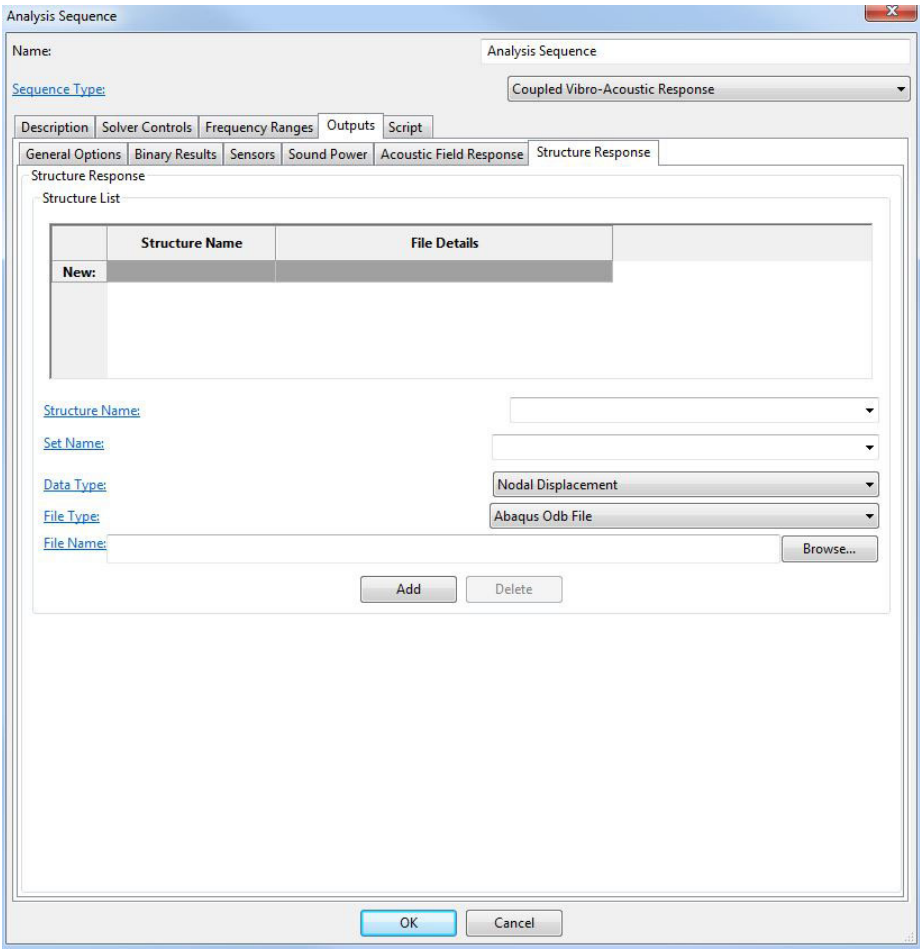


Figure 7.55 Structure Response window.



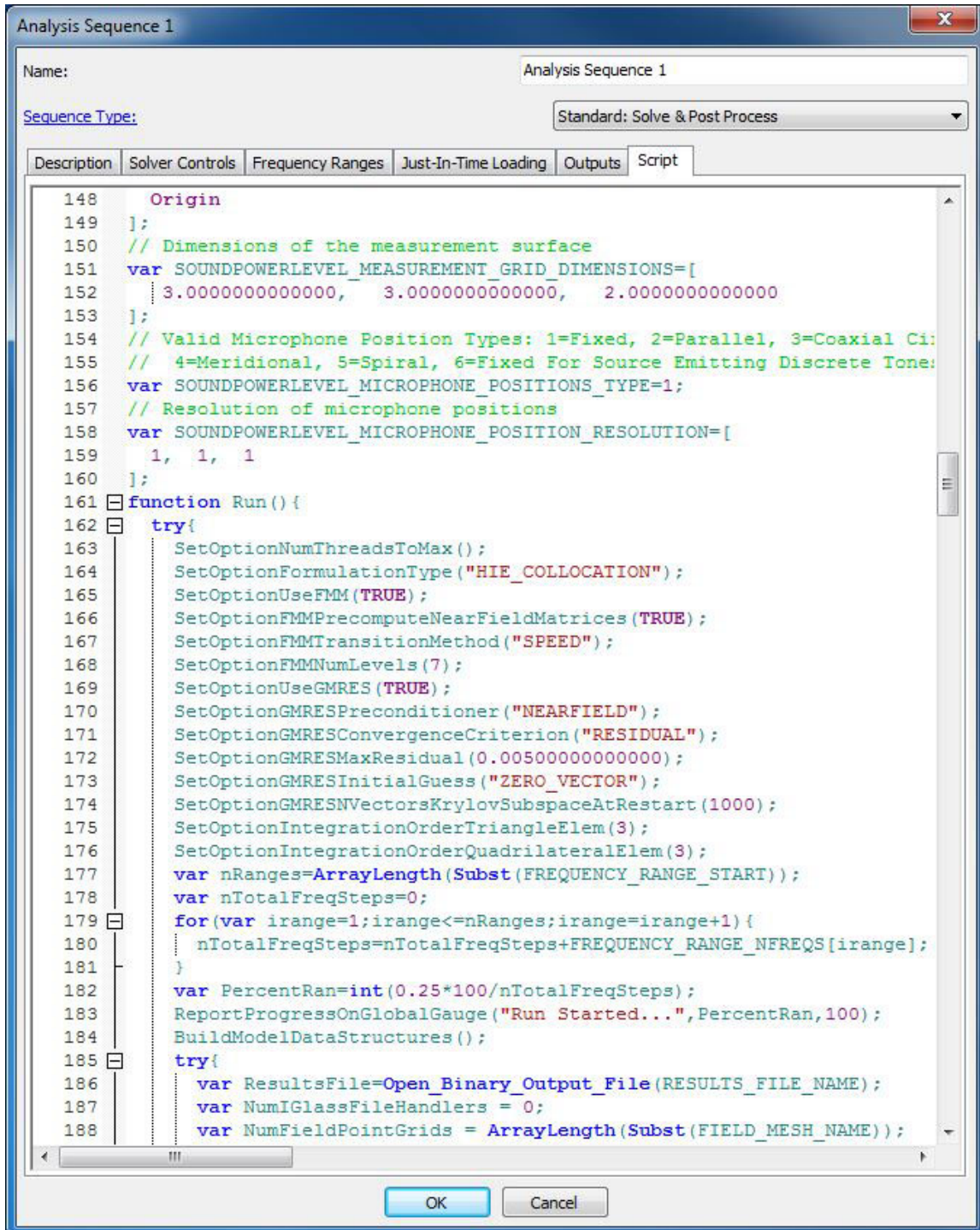


Figure 7.56 Analysis script.

on the **Script** and selecting **Open**. The user can now make desired changes to the script before closing the window. Use the syntax presented in Chapter 8 to modify the script. A simple analysis script is shown below:

```

1  function Run() {
2      try{
3          // Setup solver options:
4          SetOptionNumThreadsToMax();
5          SetOptionFormulationType("VARIATIONAL");
6          SetOptionUseFMM(TRUE);
7          SetOptionFMMPrecomputeNearFieldMatrices(TRUE);
8          SetOptionFMMTransitionMethod("SPEED");
9          SetOptionFMMNumLevels(6);
10         SetOptionUseGMRES(TRUE);
11         SetOptionGMRESPreconditioner("EBE");
12         SetOptionGMRESConvergenceCriterion("RESIDUAL");
13         SetOptionGMRESMaxResidual(0.005000000000000);
14         SetOptionGMRESInitialGuess("EXCITATION_VECTOR");
15         SetOptionGMRESNVectorsKrylovSubspaceAtRestart(1000);
16         SetOptionIntegrationOrderTriangleElem(3);
17         SetOptionIntegrationOrderQuadrilateralElem(3);
18         // Build model data structures:
19         BuildModelDataStructures();
20         try{
21             // Open binary results file:
22             var ResultsFile=Open_Binary_Output_File("results.dat");
23             for(var irange=1;irange<=1;irange=irange+1) {
24                 for(var ifreq=1;ifreq<=1;ifreq=ifreq+1) {
25                     var Freq=50;
26                     // Set analysis frequency:
27                     SetFrequency(Freq);
28                     // Build RHS vector using FMM:
29                     BuildRHSFMM("");
30                     // Solve system of equations by GMRES:
31                     var Residual=SolveUsing_GMRES_FMM("");
32                     // Write solution state at this frequency to results file:
33                     WriteAnalysisResults(ResultsFile);
34                 }
35             }
36             // Close results file opened for output:
37             Close_File(ResultsFile);
38         } catch (msg) {
39             // Clear model data structures:
40             ClearModelDataStructures();
41             throw Eval(msg);
42         }
43         // Clear model data structures:
44         ClearModelDataStructures();
45         return TRUE;
46     } catch (msg) {
47         Out(Eval("Error:"+msg));
48         return FALSE;
49     }
50 };

```

The above script can be divided into the following broad groups of commands: (1) Model setup commands (lines 19, 27, 29, 31, 40, 44), (2) Solver option setup commands (lines 4-17), (3) Post-processing commands (lines 22,33,37).

Some of the important commands and their usage syntax are described in detail below.

#### 7.4.1. Model Setup Commands

Some of the most basic commands required to build the model matrices and solve the system of equations are described below.

**BuildModelDataStructures** Assigns memory and builds various internal data structures used later in solve. Set analysis solver options, such as SetOptionUseFMM, SetOptionIntegrationOrderQuadrilateralElem, etc., before calling this function.

Usage:

```
BuildModelDataStructures();
```

**ClearModelDataStructures** Clears all memory assigned for various internal data structures during the call to BuildModelDataStructures(). The usage of this function should always be preceded by the usage of BuildModelDataStructures().

Usage:

```
ClearModelDataStructures();
```

**LoadModelFromFile** Loads the model from Coustyx (.cyx) model file. Use BuildModelDataStructures() in conjunction with this function to build internal data structures after loading a model. This function can be called only from a command script (.csr) and not from the analysis script that is within a .cyx file.

Usage:

```
var FileName = "pulsating_sphere.cyx";
var ReturnCode = LoadModelFromFile(FileName);
```

**UnloadModel** Unloads the model by clearing all the memory assigned for various internal data structures. The usage of this function should always be preceded by the usage of the functions LoadModelFromFile() and BuildModelDataStructures().

Usage:

```
UnloadModel();
```

**SetAngularFreq** Sets the analysis frequency.

Usage:

```
var Freq = 100;
SetAngularFreq(Freq*2*PI);
```

**BuildDomainMatrices** Builds the BE coefficient matrix and RHS vector. This function is used only when UseFMM is NOT turned on (for Non-FMM cases only).

Usage:

```
var BuildLHS = TRUE;
var BuildRHS = TRUE;
var DomainName="AD-1";
```



```
BuildDomainMatrices (DomainName, BuildLHS, BuildRHS);
```

**SolveDomainEquations** Solves the BE system of equations. This function is used only when UseFMM is NOT turned on (for Non-FMM cases only). Call to this function should only be made after building BE coefficient matrix and RHS vector by calling the function BuildDomainMatrices().

Usage:

```
var DomainName="AD-1";
var Residual=SolveDomainEquations (DomainName);
```

**BuildRHSFMM** Builds the RHS vector using Fast Multipole Method. This function is used only when UseFMM is turned on (FMM cases only).

Usage:

```
var DomainName="AD-1";
BuildRHSFMM(DomainName);
```

**SolveUsing\_GMRES\_FMM** Computes the matrix-vector product (Ax) using Fast Multipole Method and solves the system of equations using the Generalized Minimum Residual Method (GMRES). This function is used only when UseFMM is turned on (FMM cases). Call to this function should only be made after building the RHS vector by BuildRHSFMM().

Usage:

```
var DomainName="AD-1";
var Residual=SolveUsing_GMRES_FMM(DomainName);
```

**ImportStructureMeshFromNastranBDFFile** Imports a structure mesh in Nastran BDF format into a Coustyx model.

Usage:

```
var result = ImportStructureMeshFromNastranBDFFile( "Mesh1.BDF" );
var result = ImportStructureMeshFromNastranBDFFile( "C:\Users\john\Mesh1.BDF" );
Output
result[1] = boolean. TRUE/FALSE. TRUE if the import was successful
result[2] = String containing the name of the mesh
```

**ImportStructureMeshFromRadiossFEMFile** Imports a structure mesh in Radioss FEM file format into a Coustyx model.

Usage:

```
var result = ImportStructureMeshFromRadiossFEMFile("Mesh1.FEM");
var result = ImportStructureMeshFromRadiossFEMFile("C:\Users\john\Mesh1.FEM");
Output
result[1] = boolean. TRUE/FALSE. TRUE if the import was successful
result[2] = String containing the name of the mesh
```

**ImportStructureMeshFromAbaqusINPFile** Imports a structure mesh in Abaqus INP file format into a Coustyx model.

Usage:

```
var result = ImportStructureMeshFromAbaqusINPFile("Mesh1.inp");
var result = ImportStructureMeshFromAbaqusINPFile("C:\Users\john\Mesh1.inp");
Output
```

```
result[1] = boolean. TRUE/FALSE. TRUE if the import was successful
result[2] = String containing the name of the mesh
```

**ImportStructureMeshFromAbaqusODBFile** Imports a structure mesh in Abaqus ODB file format into a Coustyx model.

Usage:

```
var result = ImportStructureMeshFromAbaqusODBFile( <Filename>, PartIndex=1 );
Parameters:
```

Filename = String. Filename or full path of the odb file

PartIndex = int. Optional. Index of the part to be imported. Default=1

```
var result = ImportStructureMeshFromAbaqusODBFile("Mesh1.odb",1);
var result = ImportStructureMeshFromAbaqusODBFile("Mesh1.odb");
var result = ImportStructureMeshFromAbaqusODBFile("C:\Users\john\Mesh1.odb",1);
var result = ImportStructureMeshFromAbaqusODBFile("C:\Users\john\Mesh1.odb",2);
```

Output

```
result[1] = boolean. TRUE/FALSE. TRUE if the import was successful
result[2] = String containing the name of the mesh
```

**ImportStructureMeshFromAnsysRSTFile** Imports a structure mesh in Ansys RST file format into a Coustyx model.

Usage:

```
var result = ImportStructureMeshFromAnsysRSTFile("Mesh1.rst");
var result = ImportStructureMeshFromAnsysRSTFile("C:\Users\john\Mesh1.rst");
```

Output

```
result[1] = boolean. TRUE/FALSE. TRUE if the import was successful
result[2] = String containing the name of the mesh
```

**ImportStructureMeshFromUniversalFile** Imports a structure mesh in IDEAS universal file format into a Coustyx model.

Usage:

```
var result = ImportStructureMeshFromUniversalFile("Mesh1.unv");
var result = ImportStructureMeshFromUniversalFile("C:\Users\john\Mesh1.unv");
```

Output

```
result[1] = boolean. TRUE/FALSE. TRUE if the import was successful
result[2] = String containing the name of the mesh
```

**ImportStructureMeshFromPLYFile** Imports a structure mesh in PLY file format into a Coustyx model.

Usage:

```
var result = ImportStructureMeshFromPLYFile("Mesh1.ply");
var result = ImportStructureMeshFromPLYFile("C:\Users\john\Mesh1.ply");
```

Output

```
result[1] = boolean. TRUE/FALSE. TRUE if the import was successful
result[2] = String containing the name of the mesh
```

**CreateBEMeshFromStructureMesh** Skin the structure mesh to create the acoustic BE Mesh.

Usage:

```
var resultOfSkin = CreateBEMeshFromStructureMesh( structureMeshName,
```

```

    [VarConnectivityOrder], [useDefaultVarConnectivity],
    [WrapSkinAroundFreeEdge], [isDomainOnPositiveSideOfMesh] );
Output
    resultOfSkin = boolean. TRUE/FALSE. TRUE if the skinning was successful.
Parameters:
    structureMeshName = String. Name of the mesh imported in step 2
    VarConnectivityOrder = String. Optional. "SAME", "LINEAR",
        "QUADRATIC", "CUBIC", "CONSTANT". Default is "LINEAR"
    useDefaultVarConnectivity = boolean. Optional. TRUE if structure
        node ids are to be retained. FALSE otherwise. Default Value = FALSE
    WrapSkinAroundFreeEdge = boolean. Optional. TRUE if skin should
        wrap around free edges. Default Value = TRUE
    isDomainOnPositiveSideOfMesh = boolean. Optional. TRUE if domain is
        on the positive side of the mesh. FALSE otherwise. Default value = TRUE
Example:
var structureMeshName = "Mesh1";
var VarConnectivityOrder = "LINEAR";
var useDefaultVarConnectivity = FALSE;
var WrapSkinAroundFreeEdge = TRUE;
var isDomainOnPositiveSideOfMesh = FALSE;
var resultOfSkin = CreateBEMeshFromStructureMesh( structureMeshName,
    VarConnectivityOrder, useDefaultVarConnectivity, WrapSkinAroundFreeEdge,
    isDomainOnPositiveSideOfMesh );

```

**ApplyBoundaryCondition** Applies a boundary condition to a set of elements.

Usage:

```

    var result = ApplyBoundaryCondition( domainName, meshNumberInDomain,
        elementSetName, boundaryConditionName );
Parameters:
    domainName = String. Name of the domain. Usually "" for the default domain
    meshNumberInDomain = int. Usually 1 for the first mesh in the domain.
    elementSetName = String. Name of the element set where the BC should be applied.
        Sets inherited from structure mesh.
    boundaryConditionName = String. Name of the BC that should be applied.
Output
    result = boolean. TRUE/FALSE. TRUE if the operation was successful.
Example:
var myDomainName1 = "";
var meshNumberInDomain1 = 1;
var elementSetLeft = "Inlet";
var bcNameLeft = "UnitVz";
var resultApplyBCLeft = ApplyBoundaryCondition( domainName, meshNumberInDomain,
    elementSetName, boundaryConditionName );

```

#### 7.4.2. Solver Options Setup Commands

**SetOptionNumThreadsToMax** This function sets the number of threads (CPU-Cores) used for running the analysis to the maximum possible threads available to the system or the maximum allowed by the user's license, which ever is smaller.

Usage:

```

SetOptionNumThreadsToMax();

```

**SetOptionNumThreads** This function sets the number of threads (CPU-Cores) to be available for the analysis run. If the number of threads chosen is higher than the maximum allowed then Coustyx automatically chooses the latter.

Usage:

```
SetOptionNumThreads (2);
```

**SetOptionFormulationType** This function sets the type of formulation to be used to perform acoustic analysis. The function takes in any of the following strings as an input: 1) VARIATIONAL, 2) HIE\_COLLOCATION, and 3) BURTON\_MILLER\_GALERKIN. Note that HIE\_COLLOCATION and BURTON\_MILLER\_GALERKIN are not valid for Indirect BE models. VARIATIONAL is the only formulation allowed for Indirect BE models.

Usage:

```
SetOptionFormulationType ("VARIATIONAL");
SetOptionFormulationType ("HIE_COLLOCATION");
SetOptionFormulationType ("BURTON_MILLER_GALERKIN");
```

**SetOptionUseFMM** This function takes a boolean argument specifying whether or not the FMM option needs to be used.

Usage:

```
SetOptionUseFMM (TRUE);
SetOptionUseFMM (FALSE);
```

**SetOptionFMMPrecomputeNearFieldMatrices** Argument to this function is a boolean value indicating whether or not the near field matrices are computed. This is memory vs. speed trade off issue. If the near field matrices are computed, the memory usage is increased, while the time per GMRES iteration will be reduced. So if you are concerned about the RAM, and can wait a bit longer for the solution, set this option to FALSE, else set it to TRUE.

Usage:

```
SetOptionFMMPrecomputeNearFieldMatrices (TRUE);
SetOptionFMMPrecomputeNearFieldMatrices (FALSE);
```

**SetOptionFMMTransitionMethod** This function sets in the option that determines how internal FMM coefficients are represented during an analysis. Use any of the following strings as the input argument: 1) SPEED, 2) ACCURACY, and 3) BESTBLEND. Set SPEED option for faster run times, ACCURACY option for precise error control at all frequencies (but slower than SPEED), BESTBLEND option that combines the best features of SPEED and ACCURACY to yield precise error control without sacrificing the speed.

Usage:

```
SetOptionFMMTransitionMethod ("SPEED");
SetOptionFMMTransitionMethod ("ACCURACY");
SetOptionFMMTransitionMethod ("BESTBLEND");
```

**SetOptionFMMNumLevels** This function sets up the number of levels used in the cell hierarchy constructed during FMM. As a general thumb rule, the number of levels must be specified such that the smallest cells have a edge length similar to the size of the elements in the mesh. Different number of FMM levels can be used for different domains in the model (refer to Usage 1). To set the same number of FMM levels to all the domains in a model use the function call in Usage 2.

Usage 1:

```
var DomainName="AD-1";
SetOptionFMMNumLevels(Eval(DomainName), 6);
```

Usage 2:

```
SetOptionFMMNumLevels(6);
```

**SetOptionFMMNumLevelsToDefault** This function sets up the number of levels used in the cell hierarchy constructed during FMM to the recommended values for each domain in the model automatically (refer to Usage 1). To set the recommended number of FMM levels for a particular domain use the function call in Usage 2.

Usage 1:

```
SetOptionFMMNumLevelsToDefault();
```

Usage 2:

```
var DomainName="AD-1";
SetOptionFMMNumLevelsToDefault(Eval(DomainName));
```

**SetOptionUseGMRES** This function takes a boolean argument specifying whether or not to use the Generalized Minimum Residual Method (GMRES) for solving the system of equations. Note that this option should be always be set to true when FMM is being used.

Usage:

```
SetOptionUseGMRES(TRUE);
```

**SetOptionGMRESPreconditioner** A preconditioner is used to accelerate the convergence of GMRES solver (fewer iterations needed for solution). They might however, increase the RAM usage and increase the time per iteration. We have following different options for the preconditioner matrix: 1) NONE, 2) EBE, 3) NEARFIELD, and 4) DIAGONAL. NEARFIELD Preconditioner results in the fastest convergence, DIAGONAL Preconditioner is slowest, and EBE (Element-By-Element) Preconditioner is in between the two.

Usage:

```
SetOptionGMRESPreconditioner("NONE");
SetOptionGMRESPreconditioner("EBE");
SetOptionGMRESPreconditioner("NEARFIELD");
SetOptionGMRESPreconditioner("DIAGONAL");
```

**SetOptionGMRESDiagonalPreconditionerNBands** This function sets the number of bands from the diagonal of the element self influence matrices that are assembled to create the DIAGONAL Preconditioner. Note that a call to this function should always be preceded by the call: SetOptionGMRESPreconditioner(DIAGONAL) to set the Preconditioner to DIAGONAL type.

Usage:

```
SetOptionGMRESDiagonalPreconditionerNBands(1);
```

**SetOptionGMRESConvergenceCriterion** GMRES solver is used for solving the BEM system of equations. As this is an iterative solver, one could specify the criteria for convergence as follows: 1) RESIDUAL, or 2) SOUND\_POWER. RESIDUAL option uses the residual from the solution vector, residual

$R = |A * x - b|/|b|$ , as the convergence criterion. SOUND\_POWER option uses the difference between the computed sound power from two consecutive iterations.

Usage:

```
SetOptionGMRESConvergenceCriterion("RESIDUAL");
SetOptionGMRESConvergenceCriterion("SOUND_POWER");
```

**SetOptionGMRESMaxResidual** The GMRES solver terminates when the residual  $|A * x - b|/|b|$  is less than the MaxResidual specified by this function. Call to this function should always be preceded by a call to SetOptionGMRESConvergenceCriterion("RESIDUAL").

Usage:

```
SetOptionMaxResidual(0.005);
```

**SetOptionGMRESMaxSoundPowerDiff** The GMRES solver terminates when the difference between the computed sound power from two consecutive iterations is than the MaxSoundPowerDiff specified by this function. Call to this function should always be preceded by a call to SetOptionGMRESConvergenceCriterion("SOUND\_POWER").

Usage:

```
SetOptionGMRESMaxSoundPowerDiff(0.005);
```

**SetOptionGMRESInitialGuess** The GMRES solver is an iterative solver that requires an initial guess. This function sets the initial guess from the following choices: 1) PREVIOUS\_FREQ, 2) EXCITATION\_VECTOR, or 3) ZERO\_VECTOR. PREVIOUS\_FREQ option sets the solution from the previous frequency, while running a frequency sweep, as the initial guess. EXCITATION\_VECTOR option sets the excitation vector (b in  $Ax=b$ ) as the initial guess. ZERO\_VECTOR sets a zero vector as the initial guess. The selection of appropriate initial guess is important for faster convergence.

Usage:

```
SetOptionGMRESInitialGuess("PREVIOUS_FREQ");
SetOptionGMRESInitialGuess("EXCITATION_VECTOR");
SetOptionGMRESInitialGuess("ZERO_VECTOR");
```

**SetOptionGMRESNVectorsKrylovSubspaceAtRestart** 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.

Usage:

```
SetOptionGMRESNVectorsKrylovSubspaceAtRestart(500);
```

**SetOptionUseConstNumIntegrationPoints** In Coustyx, regular Gauss quadrature method is used to numerically evaluate element matrices. This function takes a boolean argument that specifies whether or not to set the number of integration points used for the Gauss quadrature method to be constant.

Usage:

```
SetOptionUseConstNumIntegrationPoints(TRUE);
SetOptionUseConstNumIntegrationPoints(FALSE);
```

**SetOptionIntegrationOrderTriangleElem**

**SetOptionIntegrationOrderQuadrilateralElem** The number of quadrature points that are used in a regular Gauss quadrature method are represented by the integration order. An integration order corresponds to the highest degree of the polynomial that can be integrated over an element with zero error. `SetOptionIntegrationOrderTriangleElem()` sets the integration order for triangle elements. `SetOptionIntegrationOrderQuadrilateralElem()` sets the integration order for quadrilateral elements.

Usage:

```
SetOptionIntegrationOrderTriangleElem(3);
SetOptionIntegrationOrderQuadrilateralElem(3);
```

**SetOptionIntegrationScheme** The element integrals computed for acoustic analysis contain  $1/r$  factor due to the presence of Green's function in the formulation. This effects the accuracy of the Gauss quadrature procedure when two elements are very close to each other. More integration points are required to accurately evaluate such integrals. Simple thumb rules are established as schemes that determine the number of integration points based on the distance between the elements and the element size. This function is used to set the integration scheme to be any of the following: 1) MEDIUM, 2) FINE, and 3) FINEST. MEDIUM accuracy level is good enough for most of the applications. Set the integration scheme to FINE or FINEST for higher accuracies.

Usage:

```
SetOptionIntegrationScheme(1); //MEDIUM
SetOptionIntegrationScheme(2); //FINE
SetOptionIntegrationScheme(3); //FINEST
```

### 7.4.3. Pre-Processing Commands

Some of the pre-processing commands are listed below.

**EnumerateFrequenciesFromFrequencyResponseDataFile** This function extracts the list of frequencies from inside a frequency response data file.

Inputs:

The input arguments to the function are File name, File type, Part name and Step name. File name is the name of the data file. File type could be any of the following strings: "AbaqusODB", "NastranOP2", "NastranPunch", "AnsysRst", "AnsysRfrq", and "IDEASUniversal". Part name and Step name are strings representing the part name or the step name from which to load the data. Note that this is applicable only for the AbaqusODB file type. For other file types use an empty string: "".

Outputs:

This function returns an array of three items. The first item is a boolean specifying the success of the function execution. The second item lists the total number of frequencies read. The third item lists all the extracted frequencies.

Usage:

```
var FileName="freqresp.op2";
var RetVal=EnumerateFrequenciesFromStructFrequencyResponseDataFile(
    Eval(FileName),"NastranOP2","", ""
);
var SuccessFlag=RetVal[1];
if(SuccessFlag){
    var nFrequencies=RetVal[2];
    var FreqList=RetVal[3];
    for(var ifreq=1; ifreq<=nFrequencies; ++ifreq){
        var Freq=FreqList[ifreq];
```

```

        Out ("Frequency = ", Eval (Freq));
    }
}

```

**LoadStructureFrequencyResponseFromFile** This function loads the frequency response data from a file into the model.

**Inputs:**

The input arguments to the function are Structure name, File name, File type, Part name, Step name, (optional) Frequencies of interest and (optional) Output variable type to load. Structure name is a string representing the structure in the model. File name is the name of the data file. File type could be any of the following strings: "AbaqusODB", "NastranOP2", "NastranPunch", "AnsysRst", "AnsysRfrq" and "IDEASUniversal". Part name and Step name are strings representing the part name or the step name from which to load the data. Note that this is applicable only for the AbaqusODB file type. For other file types use an empty string: "". Frequencies of interest is an optional argument that could be used to filter the data for only the specified frequencies. Output variable type to load is another optional argument that is applicable only for AbaqusODB file type. It is a string that could take any of the following values: "U", "UT", "UR", "TU", "V", "VT", "VR", "TV", "A", "AT", "AR", "TA". Only the data related to the specified output variable type would be loaded.

**Outputs:**

This function returns an array of three items. The first item is a boolean specifying the success of the function execution. The second item lists the total number of frequencies read. The third item lists all the extracted frequencies.

**Usage:**

```

var StructName="Struct_0";
var FileName="freqresp.op2";
var InterestedFreqList=[100,150,[300,500],1000];
var PartName="PART-1-1";
var StepName="STEP-1";
var OutputVarType="V";
var RetVal=LoadStructureFrequencyResponseFromFile (
    Eval (StructName), Eval (FileName), "AbaqusODB", Eval (PartName), Eval (StepName),
    Eval (InterestedFreqList), Eval (OutputVarType)
);
var SuccessFlag=RetVal[1];
if (SuccessFlag) {
    var nFrequencies=RetVal[2];
    var FreqList=RetVal[3];
    for (var ifreq=1; ifreq<=nFrequencies; ++ifreq) {
        var Freq=FreqList[ifreq];
    }
}

```

**ClearStructureFrequencyResponse** This function clears the frequency response data for a specified structure from the model.

**Inputs:**

A valid structure mesh name as a string.

**Usage:**

```

var StructName="Struct_0";
ClearStructureFrequencyResponse (Eval (StructName));

```



**LoadStructureInVacuoModesFromFile** This function loads the natural mode data from a file into the model.

**Inputs:**

The input arguments to the function are Structure name, File name, File type, Part name, Step name, (optional) Frequencies of interest and (optional) Output variable type to load. Structure name is a string representing the structure in the model. File name is the name of the data file. File type could be any of the following strings: "AbaqusODB", "NastranOP2", "NastranPunch", "AnsysRst", "AnsysRfrq" and "IDEASUniversal". Part name and Step name are strings representing the part name or the step name from which to load the data. Note that this is applicable only for the AbaqusODB file type. For other file types use an empty string: "". Frequencies of interest is an optional argument that could be used to filter the data for only the specified natural frequencies. Output variable type to load is another optional argument that is applicable only for AbaqusODB file type. It is a string that could take any of the following values: "U", "UT", "UR", "TU", "V", "VT", "VR", "TV", "A", "AT", "AR", "TA". Only the data related to the specified output variable type would be loaded.

**Outputs:**

This function returns an array of three items. The first item is a boolean specifying the success of the function execution. The second item lists the total number of natural modes read. The third item lists all the extracted natural frequencies.

**Usage:**

```
var StructName="Struct_0";
var FileName="modalresp.op2";
var InterestedFreqList=[100,150,[300,500],1000];
var RetVal=LoadStructureInVacuoModesFromFile(
    Eval(StructName),Eval(FileName),"NastranOP2","", "",Eval(InterestedFreqList)
);
var SuccessFlag=RetVal[1];
if(SuccessFlag){
    var nFrequencies=RetVal[2];
    var FreqList=RetVal[3];
    for(var ifreq=1; ifreq<=nFrequencies; ++ifreq){
        var Freq=FreqList[ifreq];
        Out("Natural Frequencies = ", Eval(Freq));
    }
}
```

**ClearStructureInVacuoModes** This function clears the natural mode data for a specified structure from the model.

**Inputs:**

A valid structure mesh name as a string.

**Usage:**

```
var StructName="Struct_0";
ClearStructureInVacuoModes(Eval(StructName));
```

**GetNStructureNaturalModes** Returns the number of modes stored in the specified structure of the model.

**Usage:**

```
var StructName="Struct_0";
var nModes=GetNStructureNaturalModes(Eval(StructName));
```

**CopyNaturalModeToFreqResponseData** This function is used to copy the modal response from the natural mode data of a structure to its frequency response data. It is especially useful when the user wants to perform acoustic analysis over a frequency range with the current mode as the structural velocity excitation. Coustyx uses the Freq Response Data of a specified structure as the excitation. One can only apply natural modes as structural velocity excitation by copying the mode data to the frequency response data using this function. Beware that this action clears any existing frequency response data. 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.

Inputs:

The input arguments to the function are Structure name, and the ModeIndex (index of the mode).

Usage:

```
var StructName="Struct_0";
var nModes=GetNStructureNaturalModes(Eval(StructName));
for(var Index=1; Index<=nModes; Index=Index+1) {
    CopyNaturalModeToFreqResponseData(StructName,Index);
    // Perform analysis
    // Save output
}
```

**LoadStructureModalCoordinatesFromFile** This function loads the modal coordinates data from a file into the model.

Inputs:

The input arguments to the function are Structure name, File name, and File type. Structure name is a string representing the structure in the model. File name is the name of the data file. Ansys modal coordinates file type, "AnsysMcf", is the only valid file type supported for now.

Outputs:

This function returns an array of three items. The first item is a boolean specifying the success of the function execution. The second item lists the total number of analysis frequencies read. The third item lists all extracted frequencies.

Usage:

```
var StructName="Struct_0";
var FileName="modalcoords.MCF";
var RetVal=LoadStructureModalCoordinatesFromFile(
    Eval(StructName),Eval(FileName),"AnsysMcf"
);
var SuccessFlag=RetVal[1];
if(SuccessFlag){
    var nFrequencies=RetVal[2];
    var FreqList=RetVal[3];
    for(var ifreq=1; ifreq<=nFrequencies; ++ifreq){
        var Freq=FreqList[ifreq];
        Out("Analysis Frequencies Loaded = ", Eval(Freq));
    }
}
```

**ClearStructureModalCoordinates** This function clears the modal coordinates data for a specified structure from the model.

**Inputs:**

A valid structure mesh name as a string.

**Usage:**

```
var StructName="Struct_0";
ClearStructureModalCoordinates(Eval(StructName));
```

**ComputeFrequencyResponseFromModalCoordinatesAndModes** This function is used to compute the frequency response from modal superposition method using modal coordinates and modes of the structure. Load modal coordinates and modes first before calling this function. Use the script functions: LoadStructureModalCoordinatesFromFile() and LoadStructureInVacuoModesFromFile() to load modal coordinates and modes respectively.

**Inputs:**

The input arguments to the function are Analysis frequency, Structure name, LoadModes flag, (optional, if LoadModes=TRUE) File name, (optional, if LoadModes=TRUE) File type, and (optional, if LoadModes=TRUE) LoadIndividually. Analysis frequency is the frequency at which frequency response is computed. Structure name is a string representing the structure in the model. LoadModes is a flag (TRUE or FALSE) that determines whether to load invacuo mode data from a file or use the data already loaded into the model. (Optional, if LoadModes=TRUE) File name is the name of the data file containing InVacuoModes. (Optional, if LoadModes=TRUE) File type could be any of the following strings: "AbaqusODB", "NastranOP2", "NastranPunch", "AnsysRst", "AnsysRfrq" and "IDEASUniversal". LoadIndividually is a flag (TRUE or FALSE) that determines how InVacuoModes are loaded from the file. If TRUE, InVacuoModes are loaded one at a time during the frequency response computation. If FALSE all InVacuoModes are loaded in one shot.

**NOTE:**

If there is no corresponding data in the modal coordinates file at the requested analysis frequency, Coustyx computes frequency response at nearby frequencies and uses linear interpolation to get response at the analysis frequency.

**Outputs:**

This function returns an array of three items. The first item is a boolean specifying the success of the function execution. The second item lists the frequencies at which frequency response is computed. The third item lists all the extracted frequencies.

**Usage:**

```
var Freq=1000;//Hertz
var StructName="Struct_0";
var MCFFileName="modalcoords.MCF";
var RstFileName="modes.rst";
// Load modal coordinates from file:
var MCFRetVal=LoadStructureModalCoordinatesFromFile(
    Eval(StructName),Eval(MCFFileName),"AnsysMcf"
);
// Load invacuo modes from file:
var RSTRetVal=LoadStructureInVacuoModesFromFile(
    Eval(StructName),Eval(RstFileName),"AnsysRst"
);
// Compute frequency response at Freq:
var RespRetVal=ComputeFrequencyResponseFromModalCoordinatesAndModes(
    Eval(Freq),Eval(StructName),FALSE
);
var SuccessFlag=RespRetVal[1];
```

**LoadCondensedStructureXfmMatrixFromFile** This function loads the transformation matrix (component modes) used for dynamic condensation of structure using Component Mode Synthesis (CMS) methods.

**Inputs:**

The input arguments to the function are Structure name, File type, File name, (optional) DOF map file. Structure name is a string representing the structure in the model. For the File type, currently we support only out4 file generated by Radioss (Optistruct), hence the argument should be: "RadiossOUT4". File name is the name of the data file that contains transformation matrix (component modes). DOF map file is an optional argument that refers to the file with a map between the indices in transformation matrix to the node ids and dofs [Note that this file is generated when you generate the transformation matrix in the external FE software].

**Outputs:**

This function returns an array of three items. The first item is a boolean specifying the success of the function execution. The second and third items list the total number of rows and columns in the matrix respectively.

**Usage:**

```
var StructName="HOUSING";
var FileName="HOUSING_X20A.OUT4";
var DOFMapFileName="HOUSING.DOFT";
var RetVal=LoadCondensedStructureXfmMatrixFromFile(
    Eval(StructName),"RadiossOUT4",Eval(FileName),
    Eval(DOFMapFileName)
);
var SuccessFlag=RetVal[1];
if(SuccessFlag){
    var nRows=RetVal[2];
    var nCols=RetVal[3];
}
```

**ClearCondensedStructureXfmMatrixFromFile** This function clears the transformation matrix (component modes) loaded into a structure.

**Inputs:**

A valid structure mesh name as a string.

**Usage:**

```
var StructName="HOUSING";
ClearCondensedStructureXfmMatrixFromFile(Eval(StructName));
```

**LoadCondensedStructureGeneralizedCoordinatesFromFile** This function loads the generalized coordinates of the Housing from the dynamic analysis of the gearbox in Calyx (or Transmission3D). It also performs Fast Fourier Transform (FFT) to convert the time-domain results to frequency domain suited for acoustic analysis.

**Inputs:**

The input arguments to the function are Structure name, File type, File name, (optional) Start time step for FFT analysis, (optional) End time step for FFT analysis. Structure name is a string representing the structure in the model. For the File type, currently we support only Calyx time-domain results, hence the argument should be: "CalyxTimeDomain". File name is the name of the data file generated by Calyx gearbox dynamic analysis typically of the name dofmap.dat. Note you need to make sure all the files referenced in the dofmap.dat are available in the same folder. Start and End time steps are optional

arguments to determine the time window to perform FFT. The default values would be the first and the last time steps found in the data file.

Outputs:

This function returns an array of three items. The first item is a boolean specifying the success of the function execution. The second item lists the total number of frequencies generated from the FFT on the time-domain results loaded. The third item lists all the extracted frequencies.

Usage:

```
var StructName="HOUSING";
var FileName="dofmap.dat";
var StartTimeStep=600;
var RetVal=LoadCondensedStructureGeneralizedCoordinatesFromFile(
    Eval(StructName),"CalyxTimeDomain",Eval(FileName),
    Eval(StartTimeStep)
);
var SuccessFlag=RetVal[1];
if(SuccessFlag){
    var nFrequencies=RetVal[2];
    var FreqList=RetVal[3];
    for(var ifreq=1; ifreq<=nFrequencies; ++ifreq){
        var Freq=FreqList[ifreq];
    }
}
```

**ClearCondensedStructureGeneralizedCoordinates** This function clears the dynamic response (or generalized coordinates) loaded for a specified structure in the model.

Inputs:

A valid structure mesh name as a string.

Usage:

```
var StructName="HOUSING";
ClearCondensedStructureGeneralizedCoordinates(Eval(StructName));
```

**ComputeFrequencyResponseFromCondensedStructXfmMatrixAndGeneralizedCoords** This function computes the frequency response from the transformation matrix (component modes loaded from the CMS analysis) and the generalized coordinates (loaded from Calyx gearbox dynamic analysis). Use the script functions: LoadCondensedStructureXfmMatrixFromFile() and LoadCondensedStructureGeneralizedCoordinatesFromFile() to load component modes and generalized coordinates first before calling this function.

Inputs:

The input arguments to the function are Frequency of analysis, Structure name. Structure name is a string representing the structure in the model.

Outputs:

This function returns an array of three items. The first item is a boolean specifying the success of the function execution. The second item lists the total number of frequencies generated from the FFT on the time-domain results loaded. The third item lists all the extracted frequencies.

Usage:

```
var Freq=1000;
var StructName="HOUSING";
// Load transformation matrix (component modes) from file:
```

```

var XfmMatFileName="HOUSING_X2OA.OUT4";
var XfmMatDOFMapFileName="HOUSING.DOFT";
var RetVal=LoadCondensedStructureXfmMatrixFromFile(
    Eval (StructName), "RadiossOUT4", Eval (XfmMatFileName),
    Eval (XfmMatDOFMapFileName)
);
// Load generalized coordinates from Calyx dynamic analysis results file:
var GenCoordsFileName="dofmap.dat";
var StartTimeStep=600;
var RetVal=LoadCondensedStructureGeneralizedCoordinatesFromFile(
    Eval (StructName), "CalyxTimeDomain", Eval (GenCoordsFileName),
    Eval (StartTimeStep)
);
// Compute frequency response at Freq:
var RetVal=
    ComputeFrequencyResponseFromCondensedStructXfmMatrixAndGeneralizedCoords(
        Eval (Freq), Eval (StructName)
    );
var SuccessFlag=RetVal[1];

```

**PerformFreqInterpolationForDataInResultsFile** This function is used to compute nodal responses at slave frequency by performing frequency interpolation of the data at any two master frequencies (between which the slave frequency lies). The data for the master frequency is read from a results file.

**Inputs:**

The input arguments to the function are Domain name, Frequency, Results file name, (optional) Interpolation type, (optional) DoFactorExponential, and (optional, if DoFactorExponential=TRUE) FieldPointLocation. Domain name is a string representing the domain of interest. Domains are defined only for MultiDomain models. For Indirect models use an empty string: "". Frequency is the analysis frequency at which the data needs to be interpolated. Results file name is a string providing the full path to the binary results file created by Coustyx. Use the function WriteAnalysisResults() to save results to a file. Linear interpolation, represented by string "LINEAR", is the only interpolation type that is supported for now. DoFactorExponential is a flag (TRUE or FALSE) that determines whether the function factors out the exponential ( $\exp(ikr)$ ,  $k$ =wavenumber,  $r$ =distance) while performing the interpolation and factors the exponential back in at the end. Setting the flag DoFactorExponential to TRUE is recommended for accurate interpolation. Note that you need to provide the field point location that should be used to compute the distance ( $r$ ) in the exponential factor ( $\exp(ikr)$ ) when DoFactorExponential is set to TRUE.

**Outputs:**

This function returns an array of two items. The first item is a boolean specifying the success of the function execution. The second item lists the nodal responses (results) after the frequency interpolation.

**Usage:**

```

var Freq=1000;//Hertz
var DomainName="Domain1";
var ResultsFileName="results.dat";
var FieldPointLocation=Origin+1*e1+1*e2+1*e3;
var FreqInterpRetVal=PerformFreqInterpolationForDataInResultsFile(
    Eval (DomainName), Eval (Freq), Eval (ResultsFileName), "LINEAR",
    TRUE, Eval (FieldPointLocation)
);
var SuccessFlag=FreqInterpRetVal[1];
var FreqInterpResults=FreqInterpRetVal[2];

```

**GenerateConsistentWeightageFunction** This function is used to generate a consistent weightage function. Some of the applications of this function are: it could be used to compute consistent nodal forces

from nodal pressures, or it could be used to compute consistent nodal normal velocities. Note that the consistent weightage function does not depend on the analysis frequency. It is solely dependent on the mesh geometry. The only input argument for this function is Domain name, which is a string representing the domain of interest. Domains are defined only for MultiDomain models. Use an empty string: "", for Indirect models. Based on the success of the function execution a success flag is returned.

Usage:

```
var DomainName="Domain1";
var SuccessFlag=GenerateConsistentWeightageFunction(Eval(DomainName));
```

**GetConsistentWeightedNormalVelocity** This function is used to compute consistent weighted normal velocities. Note that you need to generate consistent weightage function (call function GenerateConsistentWeightageFunction()), and also specify a structure velocity boundary condition prior to calling this function. The consistent weightage function is multiplied with the structure velocity data to compute the consistent weighted normal velocities. The only input argument for this function is Domain name, which is a string representing the domain of interest. Domains are defined only for MultiDomain models. Use an empty string "" for Indirect models. The function returns an array of two items. The first item is a boolean specifying the success of the function execution. The second item is a list of nodal normal velocities weighted appropriately using the consistent weightage function.

Usage:

```
var DomainName="Domain1";
var VelRetData=GetConsistentWeightedNormalVelocity(Eval(DomainName));
var SuccessFlag=VelRetData[1];
var StructVelData=VelRetData[2];
```

**GetFilterWeight** This function returns the weight of a filter at a specified frequency.

Inputs:

The input arguments to the function are the weighting filter type and the frequency. The weighting filter type is a string that could take any of the following values: "A", "B", "C", "D", and "Z". The second argument is the frequency value in Hertz for which the weight is to be computed.

Usage:

```
var Freq=1000;//Hertz
var FilterWeight=GetFilterWeight("A",Eval(Freq));
```

**AddAcousticSources** Use this function to add a new acoustic source to the model from the script.

Inputs:

The input arguments to the function are Domain name, and Acoustic source list. Domain name is a string representing the domain of interest. For Indirect model, use an empty string: "". Acoustic source list contains an array of source data. The array size would be equal to the number of new sources you would like to add. Each source data is again an array of size six. The first item of the array should be the Source id, a number. The second item should be a string representing the acoustic source type. For a Monopole source use: "MONOPOLESOURCE". The third item should be the source position. The fourth item should be the source strength type. Use the string "VOLUMEVELOCITY" to define volume velocity for a Monopole source. The fifth item should be the frequency dependency type of the value. Use "CONSTANT" for defining a constant complex number for the value of the volume velocity. The sixth item should be the complex value of the volume velocity.

Outputs:

This function returns a success flag upon successful execution of the function.

Usage:

```

var DomainName="Domain1";
var SrcId=1;
var SourceLocation=Origin+10*e1+2*e2+3*e3;
var VolumeVel=1;
var SuccessFlag=AddAcousticSources (
    Eval (DomainName), [[Eval (SrcId), "MONOPOLESOURCE", Eval (SourceLocation),
    "VOLUMEVELOCITY", "CONSTANT", Eval (VolumeVel)]]
);

```

**DeleteAllAcousticSources** This function deletes all acoustic sources in the model. The only argument to the function is the Domain name, which is a string of the name of the domain of interest from which to delete all acoustic sources. For Indirect models, use empty string: "". The function returns a success flag upon successful execution.

Usage:

```

var DomainName="Domain1";
var SuccessFlag=DeleteAllAcousticSources (Eval (DomainName));

```

**GetSoundSpeed** This function returns the speed of sound for the desired fluid medium (material) at the current analysis frequency.

Inputs:

The input argument to the function is the material name.

Usage:

```

var c=GetSoundSpeed("Air");

```

**GetAmbientDensity** This function returns the ambient density for the desired fluid medium (material) at the current analysis frequency.

Inputs:

The input argument to the function is the material name.

Usage:

```

var rho=GetAmbientDensity("Air");

```

#### 7.4.4. Post-Processing Commands

Some of the important post-processing commands are listed below.

**WriteAnalysisResults** This function writes the solution state at the end of each analysis frequency to a binary file. The binary file could later be loaded using the Coustyx function: ReadAnalysisResults(), to recreate the solution state without repeating the analysis.

Usage:

```

var ResultsFile=Open_Binary_Output_File("results.dat");
WriteAnalysisResults(ResultsFile);
Close_File(ResultsFile);

```

**ReadAnalysisResults** This function loads the solution state saved in the results file created by Coustyx function: WriteAnalysisResults().

Usage:

```

var LoadResultsFile=Open_Binary_Input_File("results.dat");

```



```
ReadAnalysisResults (LoadResultsFile);
Close_File (LoadResultsFile);
```

**ComputeFieldPointPressureVelocity** Computes the pressure and velocity at a field point not on the mesh.

Usage:

```
var xp = 1;
var yp = 1;
var zp = 1;
var PosnVec = Origin+xp*e1+yp*e2+zp*e3;
var ComputePressure = TRUE;
var ComputeVelocity = TRUE;
var DomainName="AD-1";
var PV = ComputeFieldPointPressureVelocity(
    DomainName,PosnVec,ComputePressure,ComputeVelocity
);
var Pressure = PV[1];
var Velocity = PV[2];
var vx = Velocity[1]; var vy = Velocity[2]; var vz = Velocity[3]
```

**GenerateIGlassHeader** Writes the geometry information of the boundary element mesh and all the field point grids to the IGlass file.

Usage:

```
var DomainName = "AD-1";
var fid = Open_Binary_Output_File("IGlass.cgl");
var GridList = []; //List of field point grids
var DecibelWeightingFilter = "Z_FILTER"; //Use A_FILTER, B_FILTER,
//C_FILTER and D_FILTER for others
var DecibelRefPressure = 2e-5;
GenerateIGlassHeader(Eval(DomainName),fid,GridList,
    Eval(DecibelWeightingFilter),DecibelRefPressure
);
```

**GenerateIGlassData** Writes the results data for each frequency to the IGlass file.

Usage:

```
var DomainName = "AD-1";
var fid = Open_Binary_Output_File("IGlass.cgl");
var GridList = []; //List of field point grids
var DecibelWeightingFilter = "Z_FILTER"; //Use A_FILTER, B_FILTER,
//C_FILTER and D_FILTER for others
var DecibelRefPressure = 2e-5;
GenerateIGlassHeader(Eval(DomainName),fid,GridList,
    Eval(DecibelWeightingFilter),DecibelRefPressure
);
var NumFreqSteps=1;
Out_Binary_Integer(fid, Eval(NumFreqSteps));
GenerateIGlassData(Eval(DomainName),fid,GridList,
    Eval(DecibelWeightingFilter),DecibelRefPressure
);
```

**ComputeSoundPower** Computes the Radiated Sound Power, Reactive Sound Power, Input Sound Power and Radiation Efficiency.

Usage:

```
var DomainName="AD-1";
var RetVal=ComputeSoundPower(DomainName);
var RadiatedSoundPower = RetVal[1];
var ReactiveSoundPower = RetVal[2];
var InputPower = RetVal[3];
var RadiationEfficiency = RetVal[4];
```

**ComputeSoundPowerFromSet** Computes the Radiated Sound Power, Reactive Sound Power, and Input Sound Power contributed by a set of elements.

Usage:

```
var DomainName="AD-1";
var SetName="Cover";//This set should already be defined in the bemesh
var RetVal=ComputeSoundPowerFromSet(Eval(DomainName),Eval(SetName));
var RadiatedSoundPower = RetVal[1];
var ReactiveSoundPower = RetVal[2];
var InputPower = RetVal[3];
```

**ComputeSoundPowerTransmittedThruFieldMesh** Computes the Radiated Sound Power, and Reactive Sound Power passing through the field mesh by integrating the time-averaged intensity through the mesh. The field mesh should be a surface mesh imported into Structures in Coustyx model file.

Usage:

```
var DomainName="AD-1";
var FieldMeshName="Structmesh_0";//This field mesh should already be present
//in the model tree member: Structures
var RetVal=ComputeSoundPowerTransmittedThruFieldMesh(
    Eval(DomainName),Eval(FieldMeshName)
);
var RadiatedSoundPower = RetVal[1];
var ReactiveSoundPower = RetVal[2];
```

**ComputeSoundPowerFromStandards** Computes the Radiated Sound Power from ISO Standards.

Usage:

```
var DomainName="AD-1";
var ISOStandard=[ "ISO3745STANDARD",
    [ "SPHEREMEASUREMENTGRID", [1.0*e1,1.0*e2,Origin], [24]],
    [ "FIXEDMICPOSITION", [20]],
    "A_FILTER",
    [101325,23,50],
];
var SoundPowerLevel=ComputeSoundPowerFromStandards(
    Eval(DomainName),Eval(ISOStandard)
);
```

**ComputeVolumeVelocityFromSet** Computes the Volume Velocity contributed by a set of Element Faces.

Usage:

```
var DomainName="AD-1";
var SetName="Cover_Side1";//This set should already be defined in the bemesh
```

```

//This set should contain element faces
var VolumeVel=ComputeVolumeVelocityFromSet (
    Eval (DomainName), Eval (SetName)
);

```

**ComputeSurfaceArea** Computes the surface area of all elements in the BE mesh.

Usage:

```

var DomainName="AD-1";
var A=ComputeSurfaceArea (Eval (DomainName));

```

**ComputeSurfaceAreaFromSet** Computes the surface area of elements in a set in the BE mesh.

Usage:

```

var DomainName="AD-1";
var SetName="Cover";//This set should already be defined in the bemesh
var A=ComputeSurfaceAreaFromSet (Eval (DomainName), Eval (SetName));

```

**EnumerateFrequenciesFromResultsFile** This function extracts list of frequencies from inside a Coustyx binary results file. The results file is created using the function: WriteAnalysisResults().

Inputs:

The function takes in the file handle to the Coustyx results file opened for input.

Outputs:

This function returns an array of three items. The first item is a boolean specifying the success of the function execution. The second item lists the total number of frequencies read. The third item lists all the extracted frequencies.

Usage:

```

var LoadResultsFile=Open_Binary_Input_File("results.dat");
var RetVal=EnumerateFrequenciesFromResultsFile(LoadResultsFile);
var SuccessFlag=RetVal[1];
if (SuccessFlag) {
    var nFrequencies=RetVal[2];
    var FreqList=RetVal[3];
    for (var ifreq=1; ifreq<=nFrequencies; ++ifreq) {
        var Freq=FreqList[ifreq];
        Out ("Frequency = ", Eval (Freq));
    }
}

```

**WriteNodalPressures** This function writes the sound pressures at all pressure nodes in the domain to an output file. Note: This command is valid for *MultiDomain* models only. The first column corresponds to the pressure node index, the next three correspond to the X, Y, Z coordinates of the node, and the last two columns correspond to the real and imaginary values of the pressure at the node.

Usage:

```

var fid = Open_Output_File("nodal_pressures.txt");
WriteNodalPressures (fid, DomainName);
Close_File (fid);

```

**WriteNodalNormalVelocities** This function writes the normal velocities at all Pn nodes in the domain to an output file. Note: This command is valid for *MultiDomain* models only. The first column corresponds

to the Pn node index, the next three correspond to the X, Y, Z coordinates of the node, and the last two columns correspond to the real and imaginary values of the normal velocity at the node.

Usage:

```
var fid = WriteNodalNormalVelocities("nodal_vn.txt");
WriteNodalNormalVelocities(fid, DomainName);
Close_File(fid);
```

**GetNumMeshes** Returns the number of meshes in the domain. Note: This command is valid for *MultiDomain* models only.

Usage:

```
var num_of_meshes=GetNumMeshes(DomainName);
```

### GetNumPresNodes

**GetNumPnNodes** Returns the number of Pressure nodes (or Pn nodes) in a given mesh of the domain. Note: This command is valid for *MultiDomain* models only.

Usage:

```
var num_of_pressure_nodes=GetNumPresNodes(DomainName, imesh);
```

### GetPresNodeID

**GetPnNodeID** Given the index, returns the Pressure node id (Pn node ID). Note: This command is valid for *MultiDomain* models only.

Usage:

```
var nodeID = GetPresNodeID(DomainName, imesh, inode);
```

### GetPresNodeValue

**GetPnNodeValue** Returns the Pressure value (Pn value) at a node. Note: This command is valid for *MultiDomain* models only.

Usage:

```
var NodalPressure=GetPresNodeValue(DomainName, imesh, nodeID);
```

### GetPresNodePosAndNormal

**GetPnNodePosAndNormal** Returns the position vector and normal vector at a given pressure node. Note: This command is valid for *MultiDomain* models only.

Usage:

```
var NodalPosNormal = GetPresNodeValue(DomainName, imesh, nodeID);
var NodalPositionVector = PosNormal[1];
var PosX = NodalPositionVector[1];
var PosY = NodalPositionVector[2];
var PosZ = NodalPositionVector[3];
var NodalNormalVector = PosNormal[2];
var NormalX = NodalNormalVector[1];
var NormalY = NodalNormalVector[2];
var NormalZ = NodalNormalVector[3];
```

### ListPresNodeIDsFromElemSet

**ListPnNodeIDsFromElemSet** Given the MeshName and Element Set Name, this function returns an array of 2 items. The first item is always equal to 1 when a valid MeshName is provided. When MeshName is not provided, that is MeshName="", then it represents the total number of DirectBEMeshes that contain the ElementSetName. The second item lists all the PresNode IDs in each of the DirectBEMeshes. Each item is an array of size 3 with the first one representing the MeshName, the second one representing the size of the PresNodeIDs list and the third the list of PresNode IDs. Note: This command is valid for *MultiDomain* models only.

Usage 1:

```
var MeshName="NewMeshCreatedFromSkin";//This should be an existing BE mesh
var SetName="Cover";//This set should already be defined in the bemesh
var RetVal=ListPresNodeIDsFromElemSet (Eval (MeshName), Eval (SetName));
var numMeshes=RetVal[1];
for (var imesh=1; imesh<=numMeshes; imesh=imesh+1) {
    var List_i=RetVal[2][imesh];
    var NameOfMesh=List_i[1];
    var numPresNodes=List_i[2];
    for (var inode=1; inode<=numPresNodes; inode=inode+1) {
        var PresNodeID=List_i[3][inode];
        Out ("PresNode ID = ", Eval (PresNodeID));
    }
}
```

Usage 2:

```
//When MeshName is not provided. All DirectBEMeshes are searched for the Set
var SetName="Cover";//This set should already be defined in the bemesh
var RetVal=ListPresNodeIDsFromElemSet ("", Eval (SetName)); //MeshName not provided
var numMeshes=RetVal[1];
for (var imesh=1; imesh<=numMeshes; imesh=imesh+1) {
    var List_i=RetVal[2][imesh];
    var NameOfMesh=List_i[1];
    var numPresNodes=List_i[2];
    for (var inode=1; inode<=numPresNodes; inode=inode+1) {
        var PresNodeID=List_i[3][inode];
        Out ("PresNode ID = ", Eval (PresNodeID));
    }
}
```

**ComputeSurfacePressures** This function computes pressure on either side of the surface mesh. It does not take any arguments. Call to this function shall be made only when the analysis solution is available. The surface pressures computed could be extracted using the function: GetPresAtMuNode(), or, they could be visualized through IGLASS visualization files. Note: This command is valid for *Indirect* models only.

**ComputeSurfacePointPressure** At a given a point on the surface, this function computes the pressure (on either side of the surface mesh). This function has to be used for points that lie on the surface ONLY. If this function is used with a point that does not lie on the surface mesh, the pressure and normal velocity values are computed at the closest point (projection) that lies on the surface. For a general spatial point, use the function ComputeFieldPointPressureVelocity. Note: This command is valid for *Indirect* models only.

Usage:

```

var xp=0;
var yp=4.86945;
var zp=9.12999;
var xpos=Origin+xp*e1+yp*e2+zp*e3;
var RetVal=ComputeSurfacePointPressure(DummyDomainName,xpos);
var pinlet_plus=RetVal[1];
var pinlet_minus=RetVal[2];

```

### GetNumSigmaNodes

**GetNumMuNodes** Returns the number of Sigma (Mu) nodes in the indirect be mesh. Note: This command is valid for *Indirect* models only.

Usage:

```
var num_of_sigma_nodes=GetNumSigmaNodes();
```

### GetSigmaNodeID

**GetMuNodeID** Returns the id of the Sigma (Mu) node, given its index. Note: This command is valid for *Indirect* models only.

Usage:

```
var nodeID = GetSigmaNodeID(inode);
```

### GetSigmaNodePosition

**GetMuNodePosition** Given an id of a Sigma (Mu) node, this function returns its position vector. Note: This command is valid for *Indirect* models only.

Usage:

```
var position=GetSigmaNodePosition(nodeID);
```

### GetSigmaNodeValue

**GetMuNodeValue** Given an id of a Sigma (Mu) node, this function returns the nodal sigma value. Note: This command is valid for *Indirect* models only.

Usage:

```
var SigmaValue=GetSigmaNodeValue(nodeID);
```

**GetNormalVelocityAtSigmaNode** This function takes in the following arguments: (1) GetNormalVelocityAtSigmaNode(SigmaNodeID,ElemID), or (2) GetNormalVelocityAtSigmaNode(SigmaNodeID,UnitVector). The function returns an array of two values for normal velocity plus and normal velocity minus. The plus and minus sides for a SigmaNode are determined by the orientation of the (1) element normal (of ElemID), or (2) UnitVector. The vector is assumed to point from the minus side to plus side. Note: This command is valid for *Indirect* models only.

Usage 1:

```

var SigmaNodeID=50;
var ElemID=60;
var RetVal=GetNormalVelocityAtSigmaNode(SigmaNodeID,ElemID);
var vn_plus=RetVal[1];
var vn_minus=RetVal[2];

```

Usage 2:

```
var SigmaNodeID=50;
var UnitVector=1*e1;
var RetVal=GetNormalVelocityAtSigmaNode(SigmaNodeID,UnitVector);
var vn_plus=RetVal[1];
var vn_minus=RetVal[2];
```

**GetPresAtMuNode** This function takes in the following arguments: (1) GetPresAtMuNode(MuNodeID,ElemID), or (2) GetPresAtMuNode(MuNodeID,UnitVector). The function returns an array of two values for pressure plus and pressure minus. The plus and minus sides for a MuNode are determined by the orientation of the (1) element normal (of ElemID), or (2) UnitVector. The vector is assumed to point from the minus side to plus side. Note: This command is valid for *Indirect* models only.

Usage 1:

```
var MuNodeID=50;
var ElemID=60;
var RetVal=GetPresAtMuNode(MuNodeID,ElemID);
var pres_plus=RetVal[1];
var pres_minus=RetVal[2];
```

Usage 2:

```
var MuNodeID=50;
var UnitVector=1*e1;
var RetVal=GetPresAtMuNode(MuNodeID,UnitVector);
var pres_plus=RetVal[1];
var pres_minus=RetVal[2];
```

### ListSigmaNodeIDsFromElemSet

**ListMuNodeIDsFromElemSet** Given the Element Set Name, this function returns an array of 2 items. The first item returns the size of the SigmaNodeList (MuNodeList). The second item lists all the SigmaNode IDs (MuNode IDs). Note: This command is valid for *Indirect* models only.

Usage:

```
var SetName="Cover";//This set should already be defined in the bemesh
var RetVal=ListMuNodeIDsFromElemSet(Eval(SetName));
var numList=RetVal[1];
for(var inode=1;inode<=numList;inode=inode+1){
    var MuNodeID=RetVal[2][inode];
    Out("MuNode ID = ",Eval(MuNodeID));
}
```





## REFERENCES

---

- [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 Standardization (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 Standardization (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  $3 + 4 * i$ .

**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 **Origin** + 2 \* **e1** + 3 \* **e2** + 5 \* **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)`.

## 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_Viewpport(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: Arithmetic 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 substitute 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;i2=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 an 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. IO Functions Syntax

Some of the Input and Output functions are listed below. These functions could be directly used in *Coustyx* scripts.

**Close\_File** Closes a file that is currently opened. Usage:

```
Close_File(ResultsFile);
```

**Open\_Binary\_Output\_File** Opens a binary file for writing. Usage:

```
var ResultsFile=Open_Binary_Output_File("results.dat");
```

**Open\_Binary\_Output\_File\_Append** Opens a binary file for writing. Output will be appended to the end of an existing file. Usage:

```
var ResultsFile=Open_Binary_Output_File_Append("results.dat");
```

**Table 8.1** Coustyx Language Grammar

Non-Terminal Symbol	Expansion
$\epsilon \rightarrow$	(empty)
input $\rightarrow$	$\epsilon$ stmt_list.toplevel
stmt_list.toplevel $\rightarrow$	$\epsilon$
	stmt_list.toplevel statement
compound_stmt $\rightarrow$	{ stmt_list }
stmt_list $\rightarrow$	$\epsilon$
	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 $\rightarrow$	$\epsilon$
	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.)

Non-Terminal Symbol	Expansion
identdecl $\rightarrow$	IDENTIFIER
	IDENTIFIER = exp
	IDENTIFIER := exp
assign_exp $\rightarrow$	$\epsilon$
	addr = exp
	addr := exp
	exp
addr $\rightarrow$	addr [ exp ]
	IDENTIFIER
exp $\rightarrow$	or_list
or_list $\rightarrow$	or_list   and_list
	and_list
and_list $\rightarrow$	and_list & binary
	binary
binary $\rightarrow$	binary relop binary
	binary + binary
	binary - binary
	binary * binary
	binary / binary
	binary ^ binary
	unary
relop $\rightarrow$	<
	<=
	>
	>=
	==
	!=
unary $\rightarrow$	addr
	aoterm
	[ list ]
	+ unary
	- unary
	! unary
	floating_point_constant
	string_constant
	TRUE
	FALSE
aoterm $\rightarrow$	aoterm [ exp ]
	IDENTIFIER ( list )
	( exp )
list $\rightarrow$	$\epsilon$
	list_1
list_1 $\rightarrow$	list_1 ; exp
	exp

**Open\_Output\_File** Opens an ascii text file for writing. Usage:

```
var ResultsFile= Open_Output_File("results.dat");
```

**Open\_Output\_File\_Append** Opens an existing ascii text file for writing. New output will be appended to the end of the existing file. Usage:

```
var ResultsFile= Open_Output_File_Append("results.dat");
```

**Out** Prints a variable or an expression to the terminal. Usage:

```
var x=10;
Out(Eval(x));
```

**Out\_Binary\_Complex\_Double\_Precision** Prints a complex number to a file in double precision. Usage:

```
var ResultsFile=Open_Binary_Output_File("results.dat");
var x=1+2*i;
Out_Binary_Complex_Double_Precision(ResultsFile,Eval(x));
```

**Out\_Binary\_Complex\_Single\_Precision** Prints a complex number to a file in single precision. Usage:

```
var ResultsFile=Open_Binary_Output_File("results.dat");
var x=1+2*i;
Out_Binary_Complex_Single_Precision(ResultsFile,Eval(x));
```

**Out\_Binary\_Double\_Precision** Prints a real number to a file in double precision. Usage:

```
var ResultsFile=Open_Binary_Output_File("results.dat");
var x=1;
Out_Binary_Double_Precision(ResultsFile,Eval(x));
```

**Out\_Binary\_Integer** Prints an integer to a file in binary format. Usage:

```
var ResultsFile=Open_Binary_Output_File("results.dat");
var x=1;
Out_Binary_Integer(ResultsFile,Eval(x));
```

**Out\_Binary\_Single\_Precision** Prints a real number to a file in single precision in binary format. Usage:

```
var ResultsFile=Open_Binary_Output_File("results.dat");
var x=1;
Out_Binary_Single_Precision(ResultsFile,Eval(x));
```

**Out\_To\_File** Outputs a variable to the file. Usage:

```
var ResultsFile= Open_Output_File("results.dat");
var nSteps=10;
Out_To_File(ResultsFile,Eval(nSteps));
```

## 8.10. 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:  $\text{Sin}(\pi/2)=1$ .

**Cos or cos** Obtain cosine of an angle. The angle is assumed to be in radians. Example:  $\text{Cos}(\pi)=-1$ .

**Tan or tan** Obtain tangent of an angle. The angle is assumed to be in radians. Example:  $\text{Tan}(\pi/4)=1$ .

**Asin or asin** Obtain the arc sine (inverse function of sine) of a number between -1 and +1. The output is in radians. Example:  $\text{Asin}(0.707107)=0.785398 (= \pi/4)$ .

**Acos or acos** Obtain the arc cosine (inverse function of cosine) of a number between -1 and +1. The output is in radians. Example:  $\text{Acos}(0.707107)=0.785398 (= \pi/4)$ .

**Atan or atan** Obtain the (2-quadrant) arc tangent (inverse function of tangent) of a number. The output is in radians. Example:  $\text{Atan}(1)=0.785398 (= \pi/4)$ .

**atan2(y,x)** Obtain the 4-quadrant arc tangent of the real arguments  $x$  and  $y$ .  $-\pi \leq \text{atan2}(y, x) \leq \pi$ .

**Exp or exp** Obtain the exponential of a number. Example:  $\text{Exp}(1)=2.71828 (=e^1)$ .

**Ln or ln** Obtain logarithm to the base  $e$  (natural logarithms). Example:  $\text{Ln}(10)=2.30259$ ,  $\text{Ln}(\text{exp}(1))=1$ .

**int** Obtain the integer value of a number. Example:  $\text{int}(4.926)=4$ ,  $\text{int}(-4.926)=-4$ .

**abs** Obtain the absolute value of a real number or the magnitude of a complex number. Example:  $\text{abs}(-10.958)=10.958$ ,  $\text{abs}(-3+4*i)=5$ .

**mag2(x,y)** Obtains the magnitude  $\sqrt{x^2 + y^2}$  for real arguments  $x$  and  $y$ . Example:  $\text{mag2}(3,-4)=5$ .

**real, imag** Obtain real and imaginary values of a complex number respectively. Example:  $\text{real}(3+4*i)=3$ ,  $\text{imag}(3+4*i)=4$ .

## 8.11. 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.

### 8.11.1. 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  $\text{Plm}(z)$  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  $[m1,m2]$

**8.11.2. 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  $P_n(z)$  for degrees  $n$  in  $[0,maxorder]$

**8.11.3. 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  $Y_{lm}(\theta,\phi)$  for degree  $l$ , and all orders  $[m1,m2]$

**8.11.4. 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  $h_{1n}(z)$

### 8.11.5. Spherical Hankel Function of Second Kind

#### **$h_{2n}(n, nterms, z, result)$**

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  $h_{2n}(z)$

### 8.11.6. Cylindrical Hankel Function of First Kind

#### **$H_{1n}(n, nterms, z, result)$**

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  $H_{1n}(z)$

#### **$H_{1nu}(n, nterms, z, result)$**

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  $H_{1nu}(z)$

### 8.11.7. Cylindrical Hankel Function of Second Kind

#### **$H_{2n}(n, nterms, z, result)$**

**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  $[n, n+1, \dots, n+nterms-1]$  will be computed.

complex:  $z$  - argument to the Cylindrical Hankel function

**Output:**

complex array result contains  $H_2n(z)$

**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  $[n, n+1, \dots, n+nterms-1]$  will be computed.

complex:  $z$  - argument to the Cylindrical Hankel function

**Output:**

complex array result contains  $H_{2nu}(z)$

**8.11.8. 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  $[n, n+1, \dots, n+nterms-1]$  will be computed.

complex:  $z$  - argument to the Spherical Bessel function

**Output:**

complex array: result contains  $j_n(z)$

**8.11.9. 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  $[n, n+1, \dots, n+nterms-1]$  will be computed.

complex:  $z$  - argument to the Cylindrical Bessel function

Output:

complex array: result contains  $J_n(z)$

**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  $[n, n+1, \dots, n+nterms-1]$  will be computed.

complex:  $z$  - argument to the Cylindrical Bessel function

Output:

complex array: result contains  $J_{nu}(z)$

# APPENDIX A

## 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’.

### A.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.

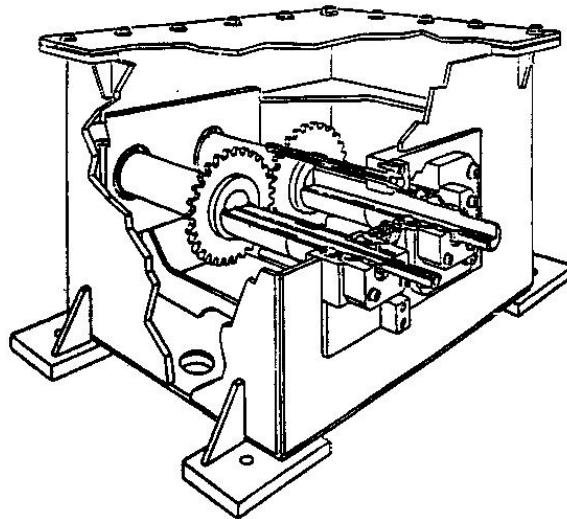
3. Compute the housing structural response at each frequency of analysis using a FEA software. The gearbox housing frequency response is computed by modal superposition.

**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.

## A.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 A.1. The structure is a rectangular box



**Figure A.1** Detail of the NASA Lewis gearbox.

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.

## A.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.

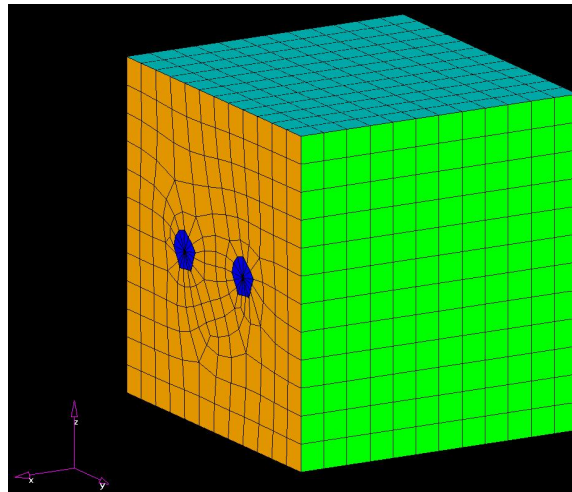
- The structural response due to forces on the housing at each frequency in the frequency domain is loaded into the Coustyx model. These values are used as the input velocity boundary condition.

Coustyx can read FEA data from NASTRAN, ABAQUS, and ANSYS softwares.

### A.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 A.2 has 974 nodes, 943 quadrilateral plate elements, 1 concentrated mass, and 4 rigid elements.



**Figure A.2** FE model of the NASA Lewis gearbox.

The physical properties of the materials used in FEA analysis are given in Table A.1.

**Table A.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

### A.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 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 A.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

**Table A.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

gearbox undergo deformation as well. Figure A.3 shows the first five free vibration modes of the gearbox housing.

### A.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 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.

## A.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.

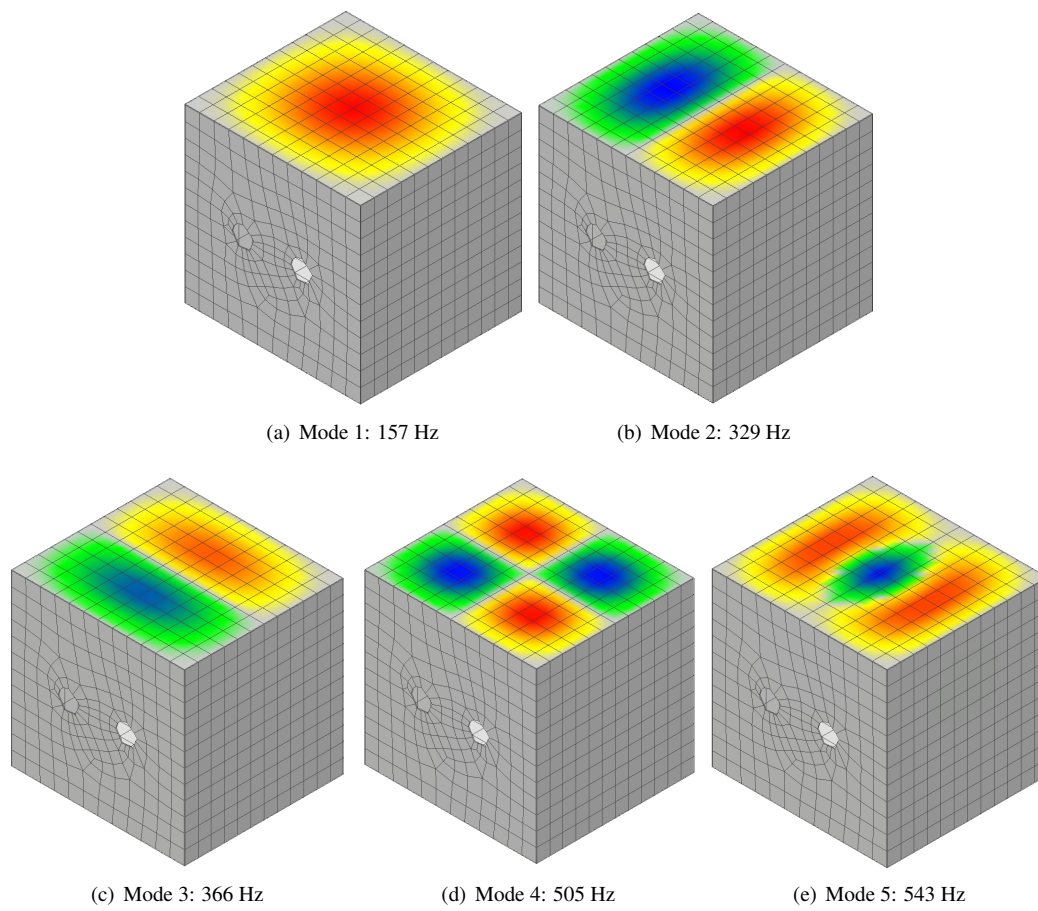
### A.4.1. Problem Setup

Follow the steps to setup Coustyx model and perform acoustic analysis on the gearbox housing. Open Coustyx from the start menu of your computer.

#### A.4.1.1. Create a New Model

- In the main menu select: **File** → **New Model**. The window in Figure A.4 will then appear.





**Figure A.3** Surface normal velocity distribution for the first five free vibration modes.

- 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.

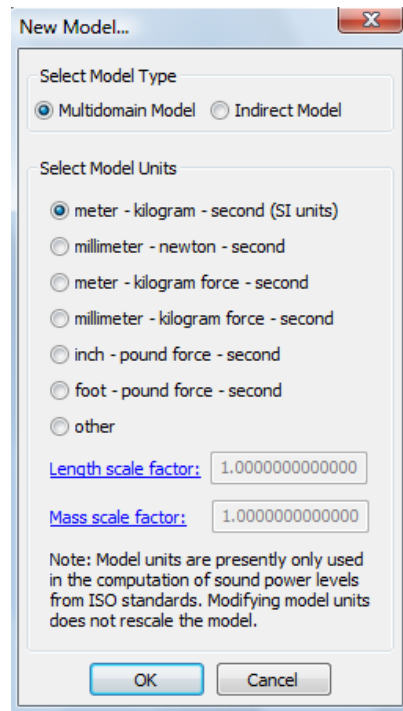


Figure A.4 New Model Selection Window.

#### A.4.1.2. Import FE Structure Mesh

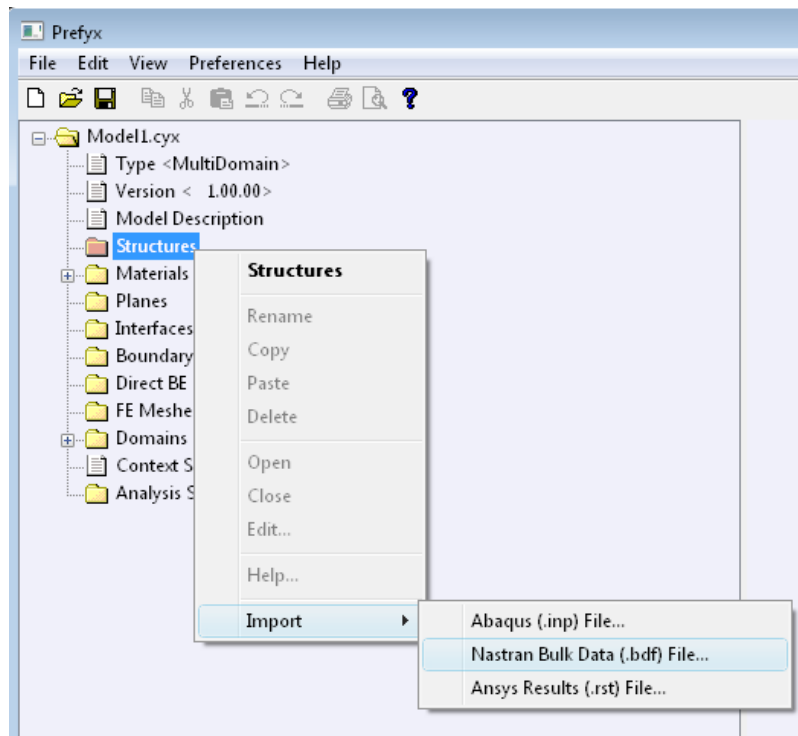
- In Coustyx model main menu select: **Model** → **Structures**.
- Right-click on **Structures** and select: **Import** → **Nastran Bulk Data (.bdf) File...** (as shown in Figure A.5) to import mesh from Nastran bulk data format. The FE meshes from Abaqus and Ansys data formats can be imported by selecting **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’.

#### A.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 A.6). The other valid data formats from which frequency response data can be loaded into Coustyx are **Nastran OP2 File** and **Ansys rst File**.
- Select the appropriate frequency response data file to be loaded from the browser and click ‘Open’.

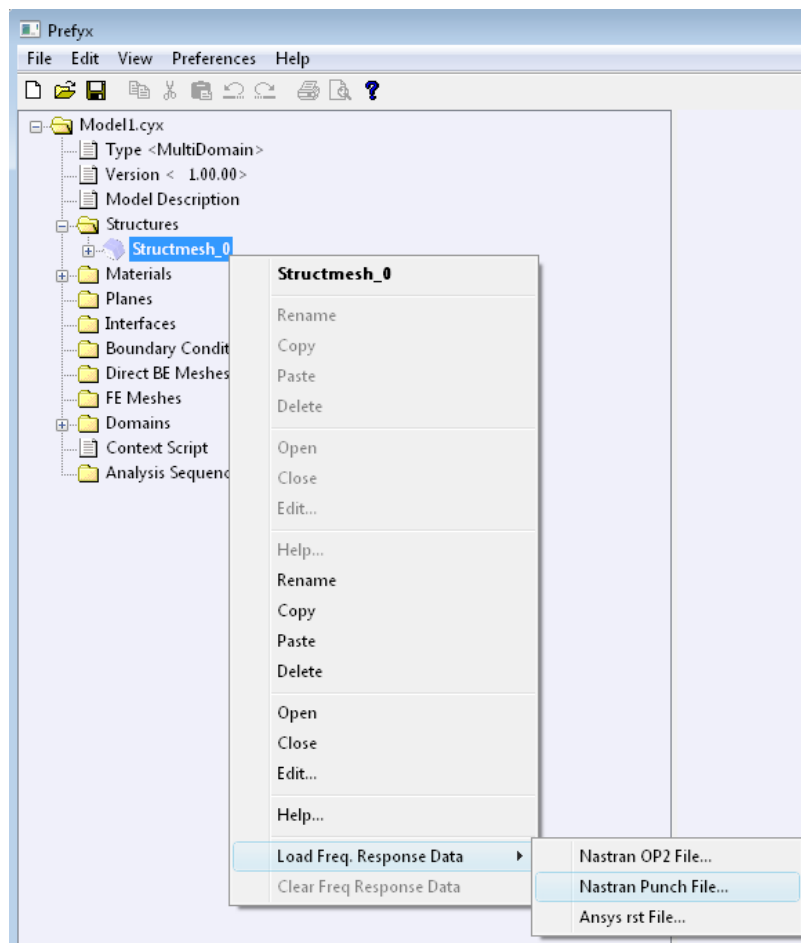
#### A.4.1.4. Generate BE Mesh

- 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 A.7.



**Figure A.5** Import a finite element structure mesh.

- 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 A.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.
- 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 A.9 and select: **Select All Displayed Elements**.
  - 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 A.10.
  - Pick the option **Show Edges** and click 'OK'.
  - 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.



**Figure A.6** Load frequency response data into Coustyx.

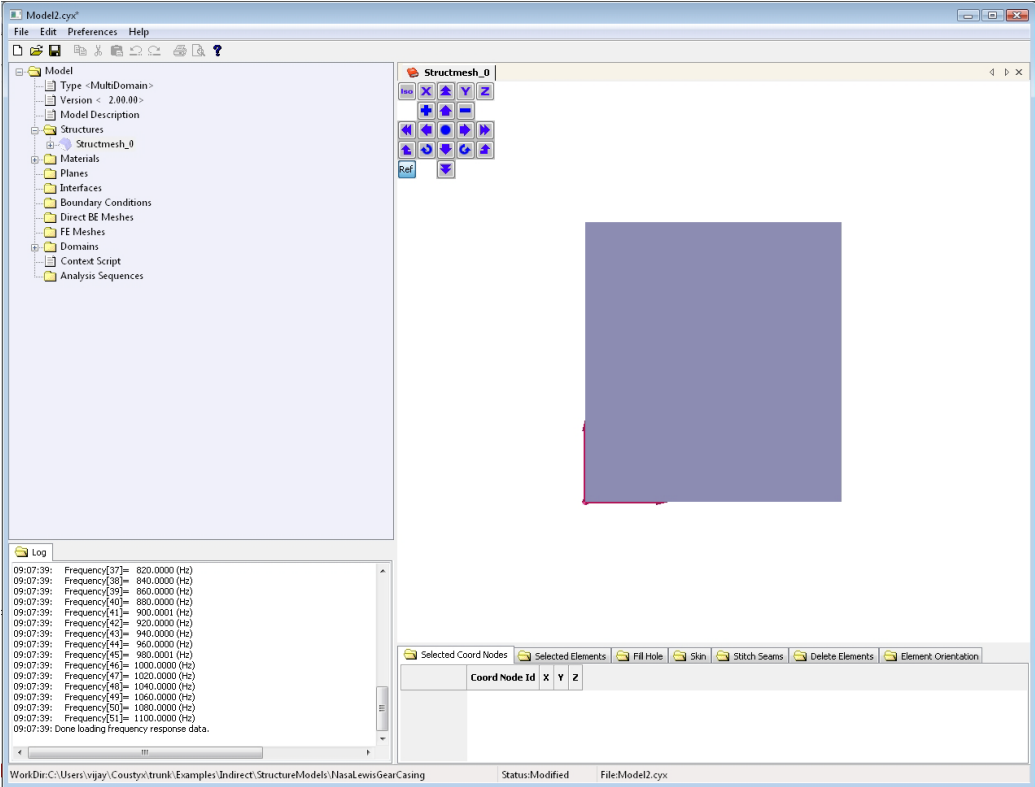


Figure A.7 Structure mesh opened in Coustyx GUI.

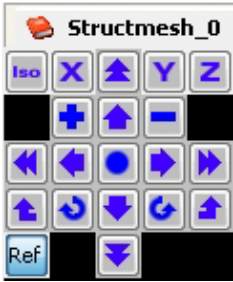
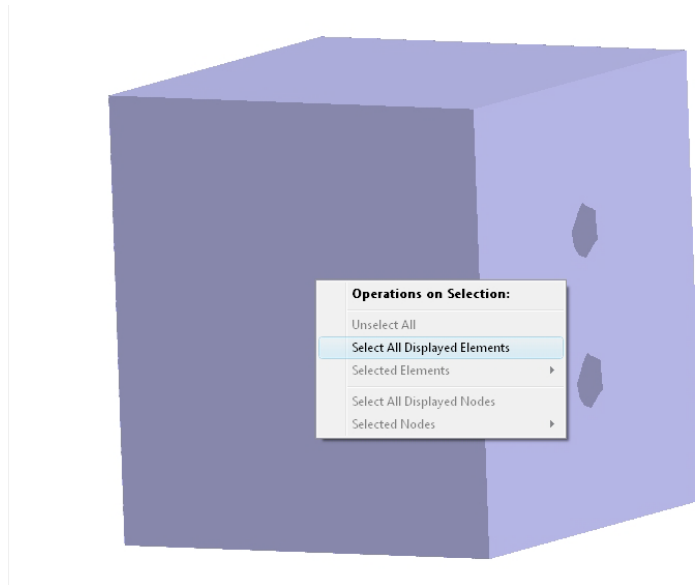
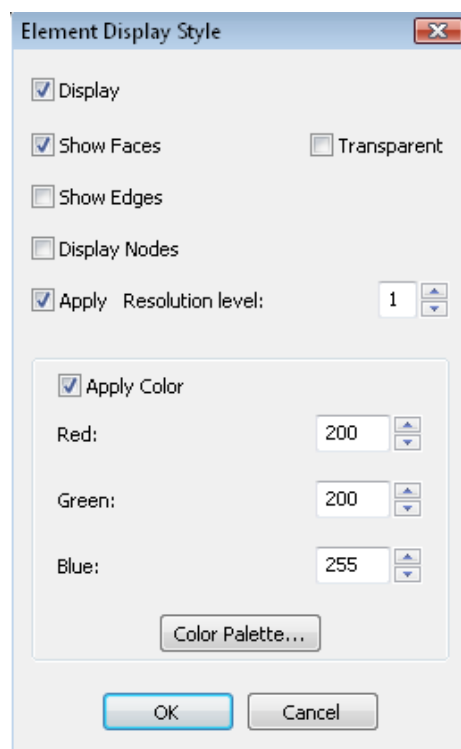


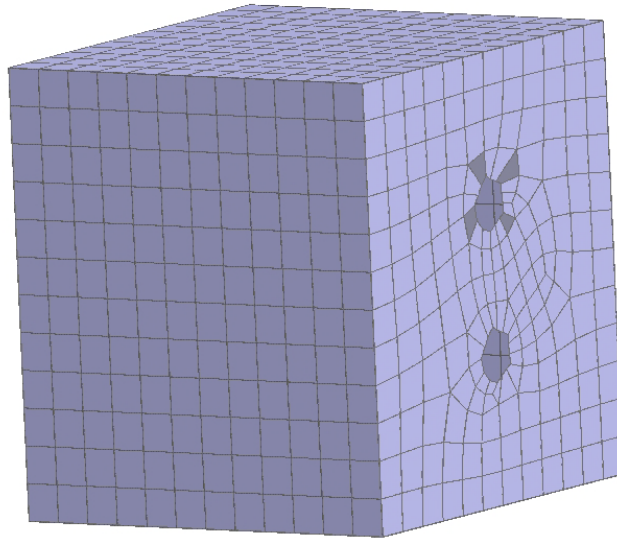
Figure A.8 Coustyx GUI control panel tools.



**Figure A.9** Select all displayed elements in the GUI.

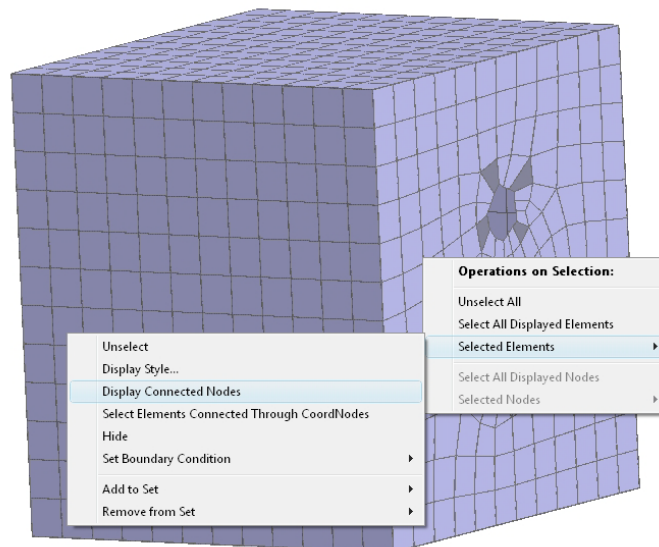


**Figure A.10** Element Display Style Window.



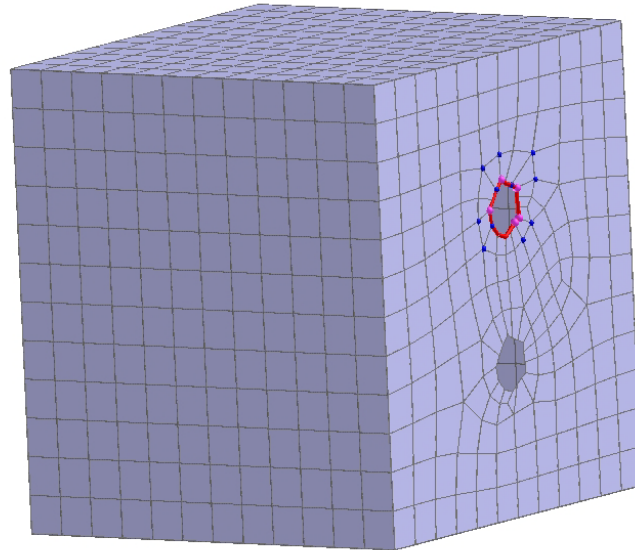
**Figure A.11** Select elements for creating a seam.

- 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 A.11.
- 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 A.12.



**Figure A.12** Display connected nodes for creating a seam.

- 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 A.13.



**Figure A.13** Pick nodes to create a seam.

- 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.
- 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.

#### **A.4.1.5. Define Material Properties**

- In the main model menu select: **Model** → **Materials** → **Air**. Right-click on **Air** and select **Edit...** Figure A.14 will appear.
- Type-in the name of the material as 'Air'.
- Define **SpeedSound** as a constant with value  $343\text{ m/s}$ . The unit is consistent with the unit of length ( $m$ ) in the structure mesh.
- Define **AmbientDensity** as a constant with value  $1.21\text{ kg/m}^3$ . The unit is consistent with the unit of length ( $m$ ) in the structure mesh.



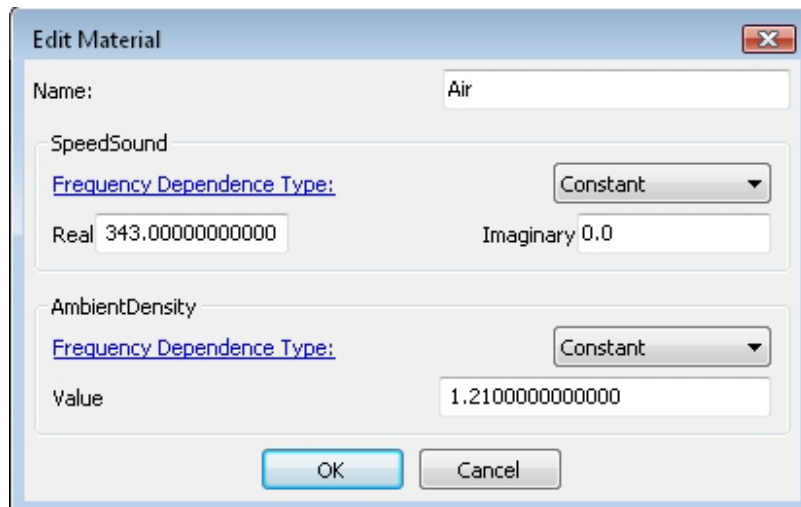


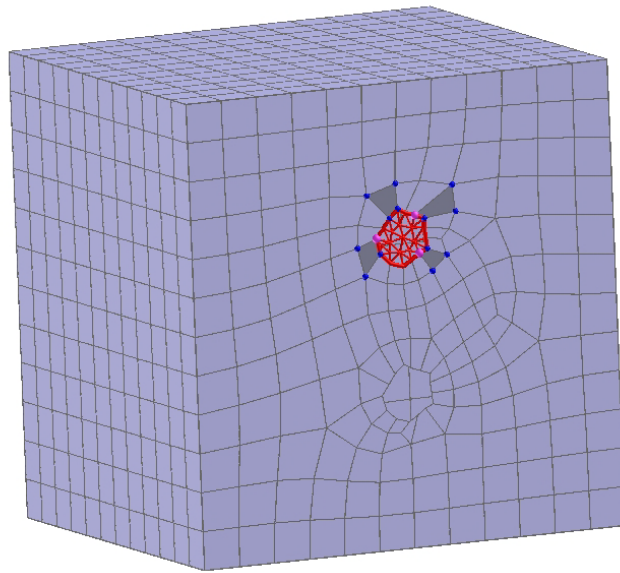
Figure A.14 Edit material properties.

#### A.4.1.6. Fill Holes

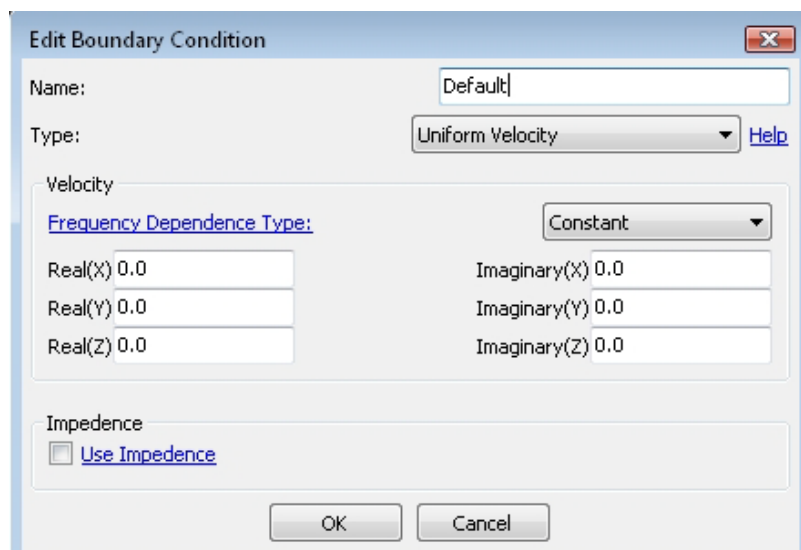
- Open the Coustyx BE mesh from the main model menu by selecting: **Model** → **Direct BE Meshes** → **NewMeshCreatedFromSkin**. Right-click on **NewMeshCreatedFromSkin** and select: **Open** to view the boundary element mesh in the GUI.
- Select the tabbed window **Fill Hole** from the series of tabs located below the BE mesh.
- Follow the instructions given earlier in Section A.4.1.4 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 A.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 A.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 A.15.
- 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.

#### A.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 A.16 will appear.
  - Type-in the new name 'Structural Velocity BC'.



**Figure A.15** Fill hole using triangle elements.



**Figure A.16** Edit boundary conditions window.

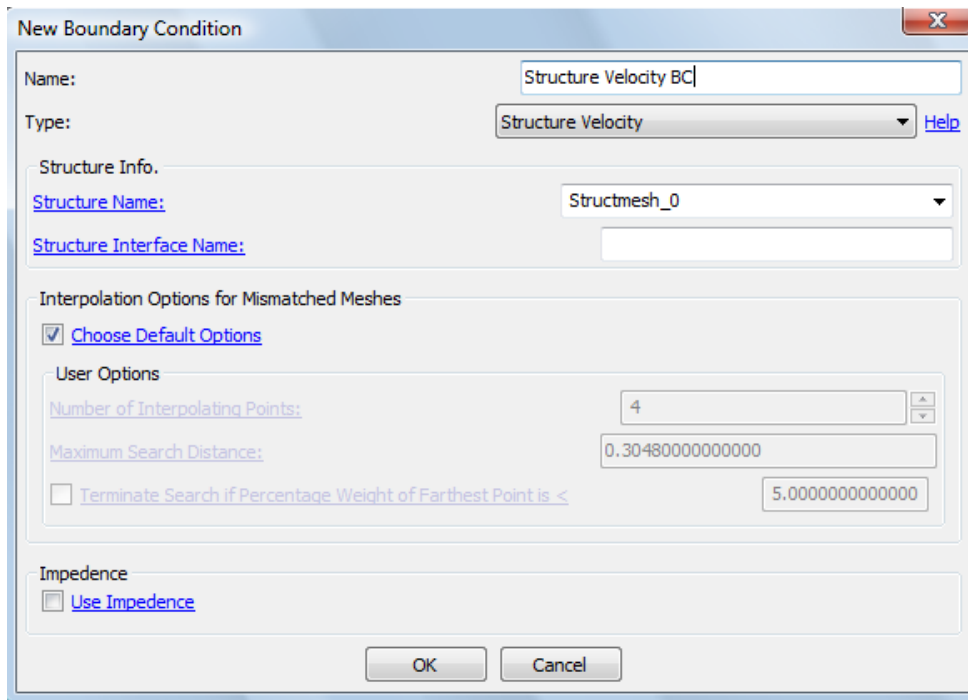


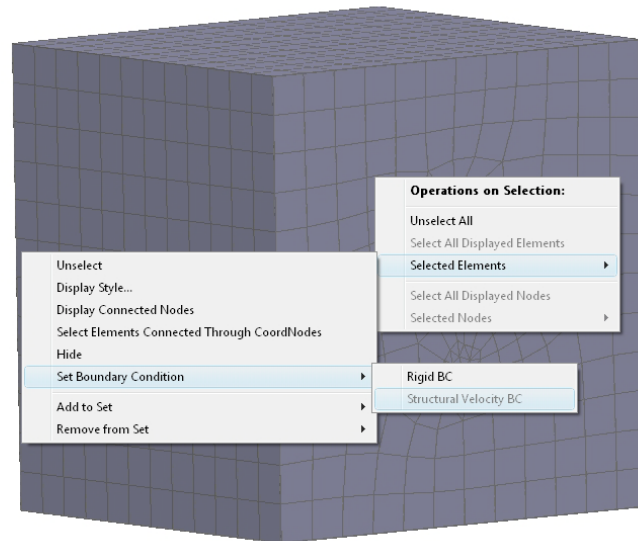
Figure A.17 Edit structure velocity boundary condition window.

- Select **Structure Velocity** from the drop-down menu for ‘Type’. The window in Figure A.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.
- 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.

**A.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 A.18). If the boundary

condition **Structure Velocity BC** is inactive, it implies that it has already been applied over the selected elements.

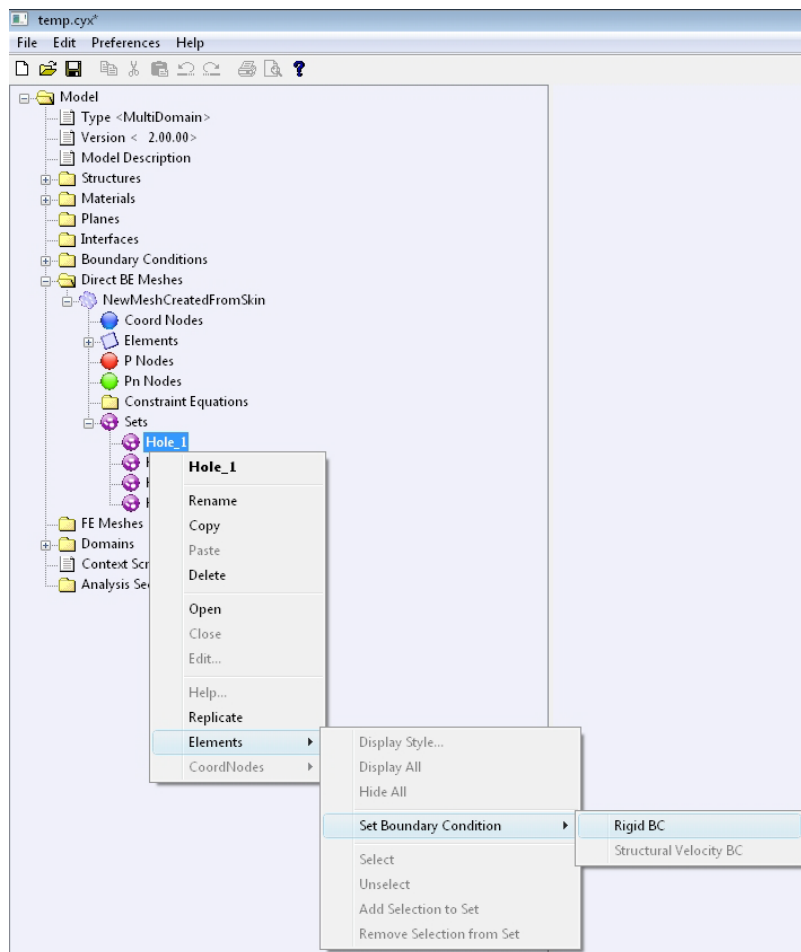


**Figure A.18** Apply boundary conditions through selected elements.

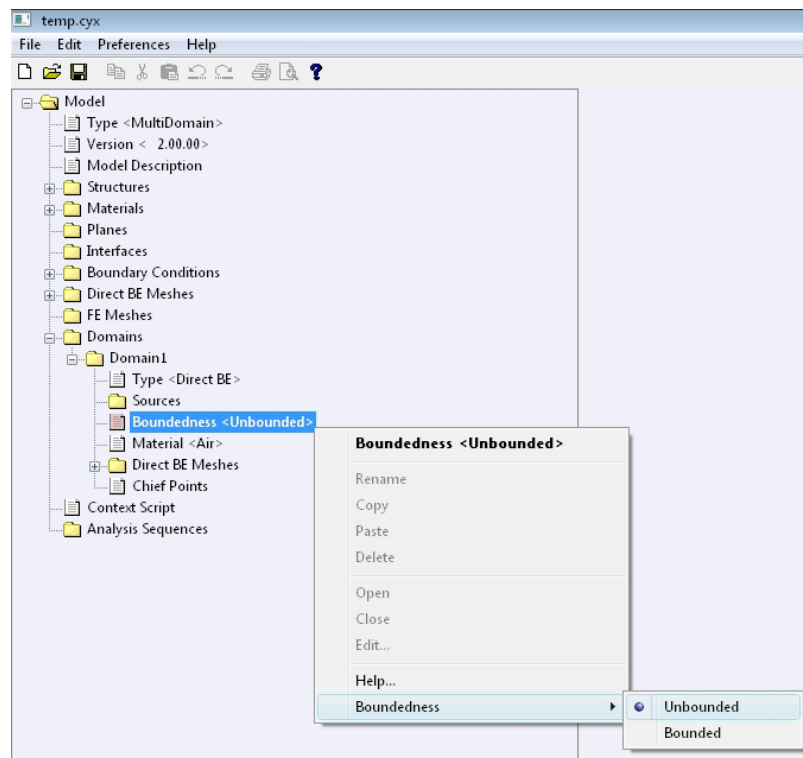
- Apply rigid boundary conditions on all the elements created to fill holes. Note that we don't have structure velocities for these as they are newly created in Coustyx and not present in the original structure. Since the elements filling the holes are conveniently added to sets named 'Hole\_1', 'Hole\_2' and so on, we can apply the boundary conditions on them through these sets.
  - In the main model menu select: **Model** → **Direct BE Meshes** → **NewMeshCreatedFromSkin** → **Sets** → **Holes\_1**.
  - Right-click on **Holes\_1** and select: **Elements** → **Set Boundary Condition** → **Rigid BC** as shown in Figure A.19.
  - Repeat the above for other holes as well.

#### A.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 A.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**.



**Figure A.19** Apply boundary conditions through sets.



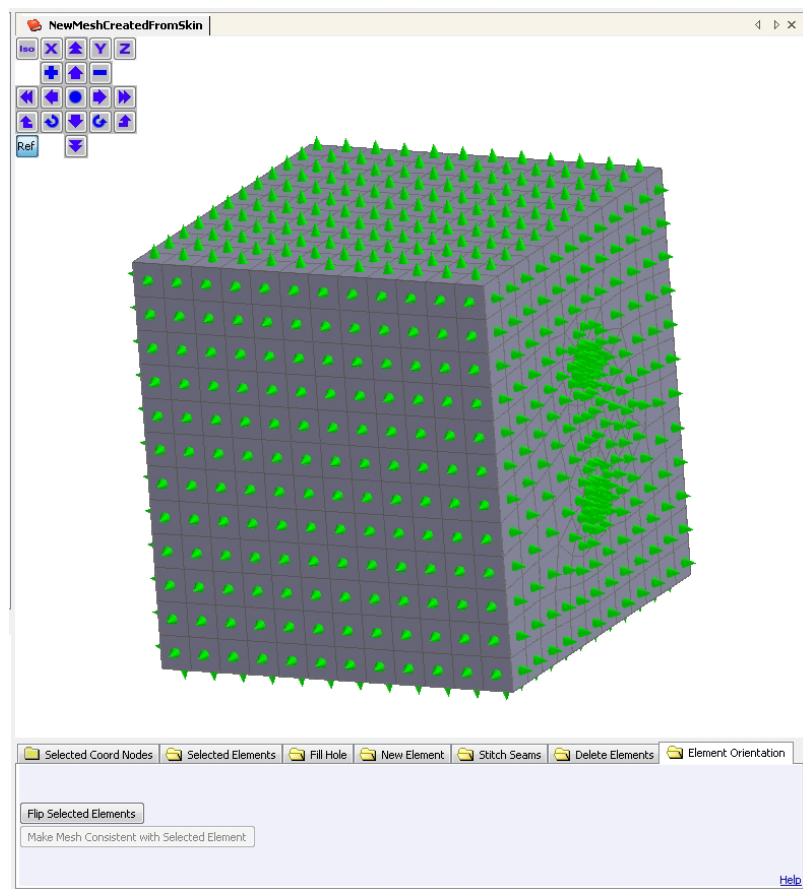
**Figure A.20** Specifying boundedness of the domain.

- From the tabbed windows located below the mesh select: **Element Orientation** to view the direction of element normals (refer to Figure A.21). The green arrow indicates the positive direction and the red arrow indicates the negative direction of 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.
- 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 A.22.

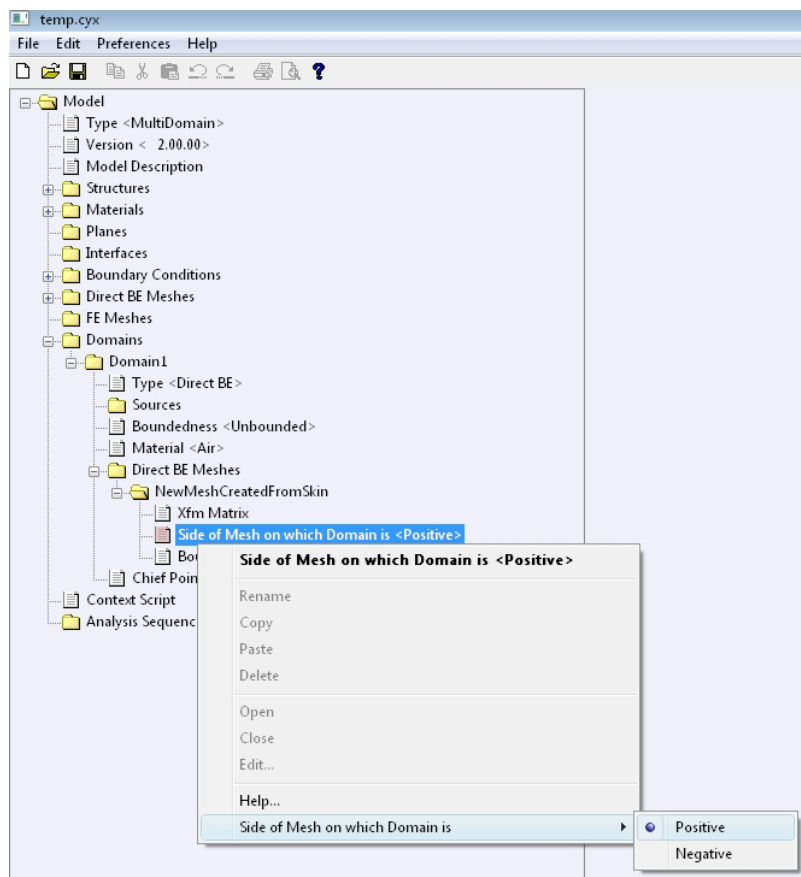
#### A.4.2. Run Acoustic Analysis

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 A.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 A.24.
  - Enter the starting frequency to be 100 Hz in the table under 'Start (Hz)'.



**Figure A.21** Element orientations.



**Figure A.22** Set the side of the mesh on which the domain is.



**Analysis Sequence**

Name: Analysis Sequence

Description Solver Controls Frequency Ranges Outputs Script

**Parallel Processing**

☒ Use Maximum Possible Number of CPUs

Number of CPUs requested: 1

**FMM**

☒ Use FMM

☒ Precompute Near Field Matrices

Number of FMM Levels: 6

Solution Method: GMRES

**GMRES**

Preconditioner Type: EBE

Number Bands including Diagonal: 1

Convergence Criterion: Residual

Residual as a Percentage: 0.5000000000000000

Change in Sound Power as a Percentage: 0.5000000000000000

Number of Vectors in Krylov Subspace at Restart: 1000

Initial Guess: Previous Solution

**Integration**

☒ Use Fixed Integration Order:

Integration Order for Triangular Elements: 3

Integration Order for Quadrilateral Elements: 3

Variable Order Integration Scheme: Medium

OK Cancel

Figure A.23 Analysis solver controls.

- 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)’.

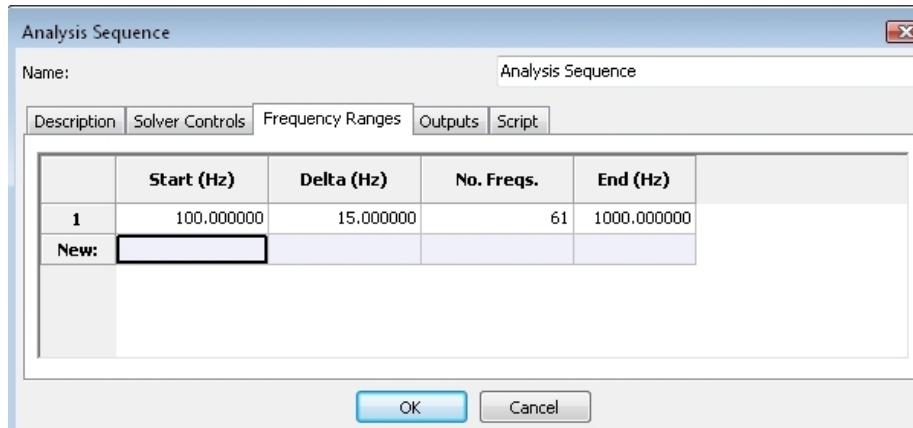


Figure A.24 Set analysis frequencies.

- 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.

### A.4.3. Post-processing/Outputs

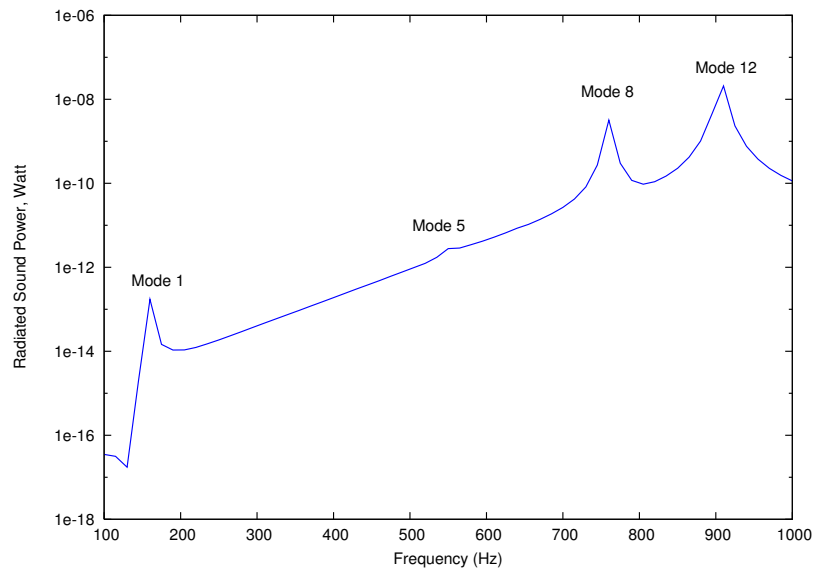
Coustyx creates the following output files based on the choices made in **Outputs** tab in **Analysis Sequence**.

**A.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.

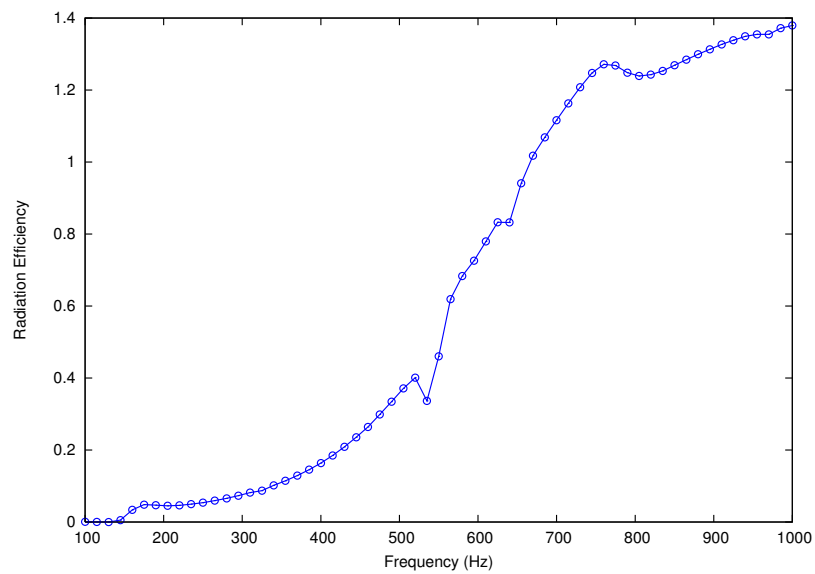
**A.4.3.2. sensors.dat** The pressure and particle velocity at the sensor locations are written into this *ASCII*-text file. Since we didn’t add any sensors to the gearbox housing radiation problem, this file is empty.

**A.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.

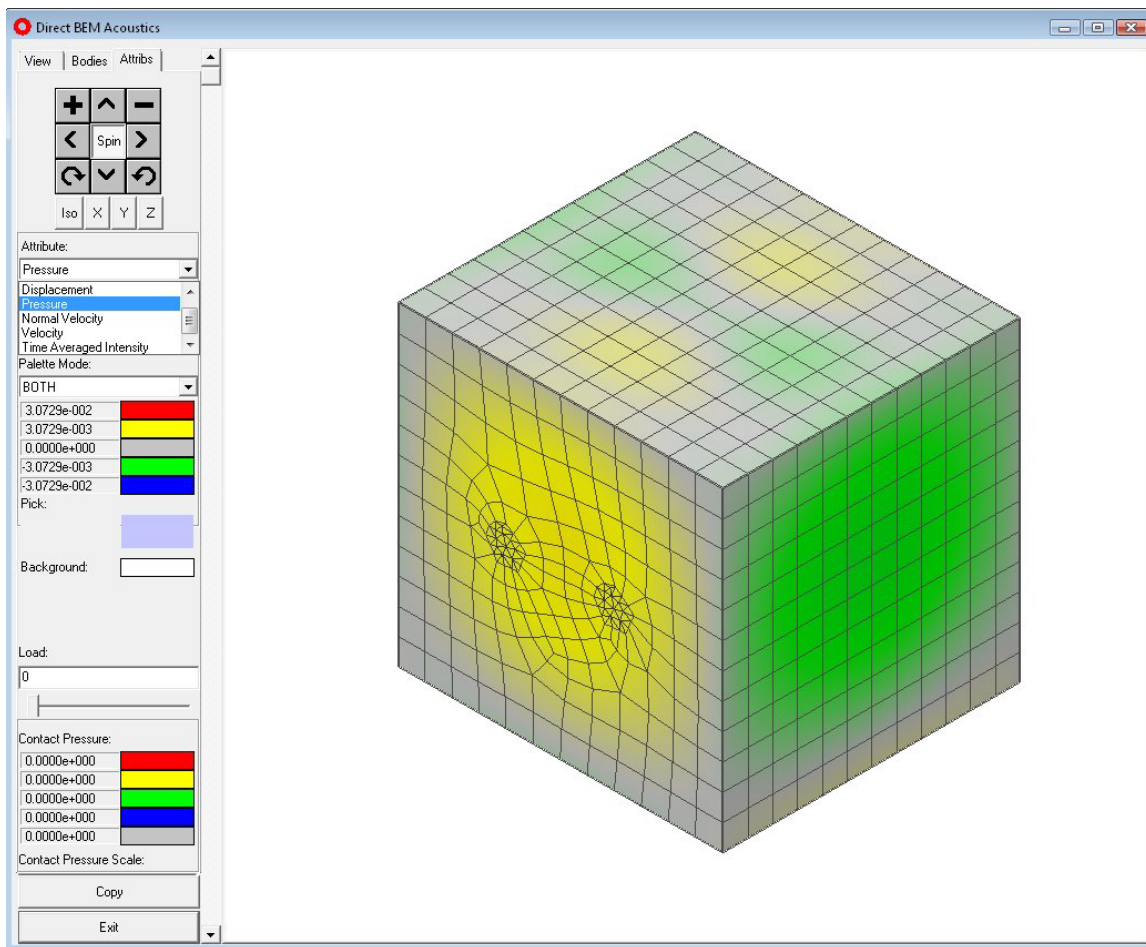
The radiated sound power and radiation efficiency are plotted against the analysis frequency in Figure A.25 and Figure A.26 using matlab plot command. The sound power radiated (Figure A.25) has peaks corresponding to the structural vibration modes that have a non-zero net volume velocity.



**Figure A.25** Sound power from the forced vibration response.



**Figure A.26** Radiation efficiency of the gearbox forced vibration.



**Figure A.27** IGlass viewer showing sound pressure distribution at 760 Hz.

**A.4.3.4. *iglass.cgl*** IGlass files are post-processing data files created by Coustyx to visualize the acoustic analysis results. Refer to Figure A.27.

- Double-click on ‘iglass.cgl’ 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**.

## A.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.

### A.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.

#### A.5.1.1. Create a New Model

- In the main menu select: **File** → **New Model**. The window in Figure A.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.

#### A.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 A.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.
- Select the appropriate FE structure data file to be imported from the browser and click ‘Open’.

#### A.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 A.30). The other valid data formats from which frequency response data can be loaded into Coustyx are **Nastran OP2 File** and **Ansys rst File**.
- Select the appropriate frequency response data file to be loaded from the browser and click ‘Open’.

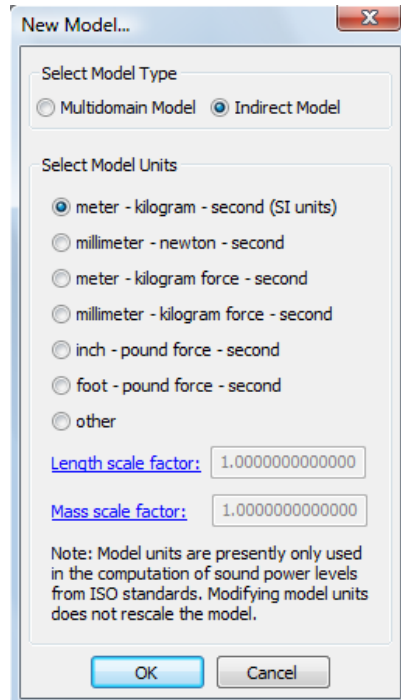


Figure A.28 New Model Selection Window.

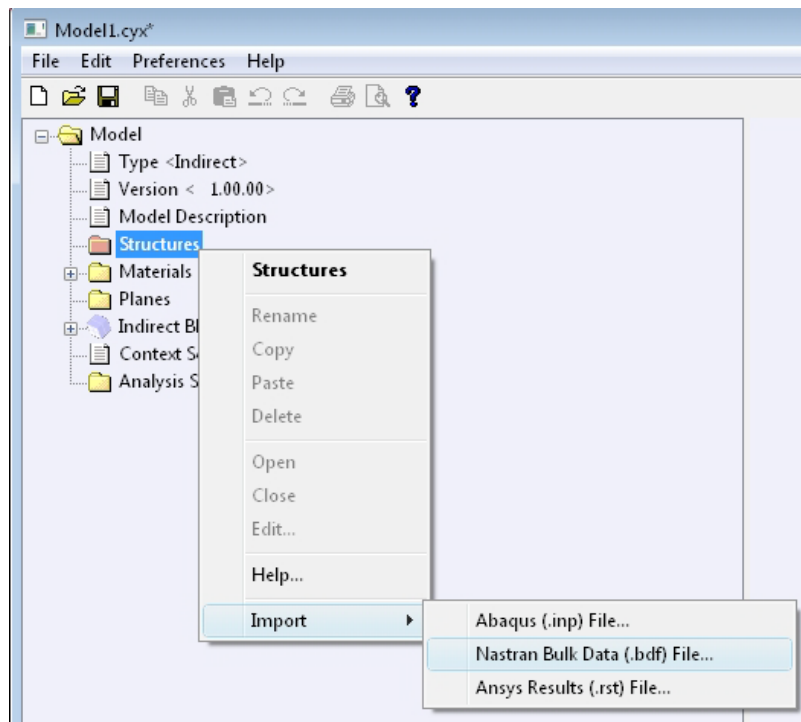


Figure A.29 Import a finite element structure mesh.

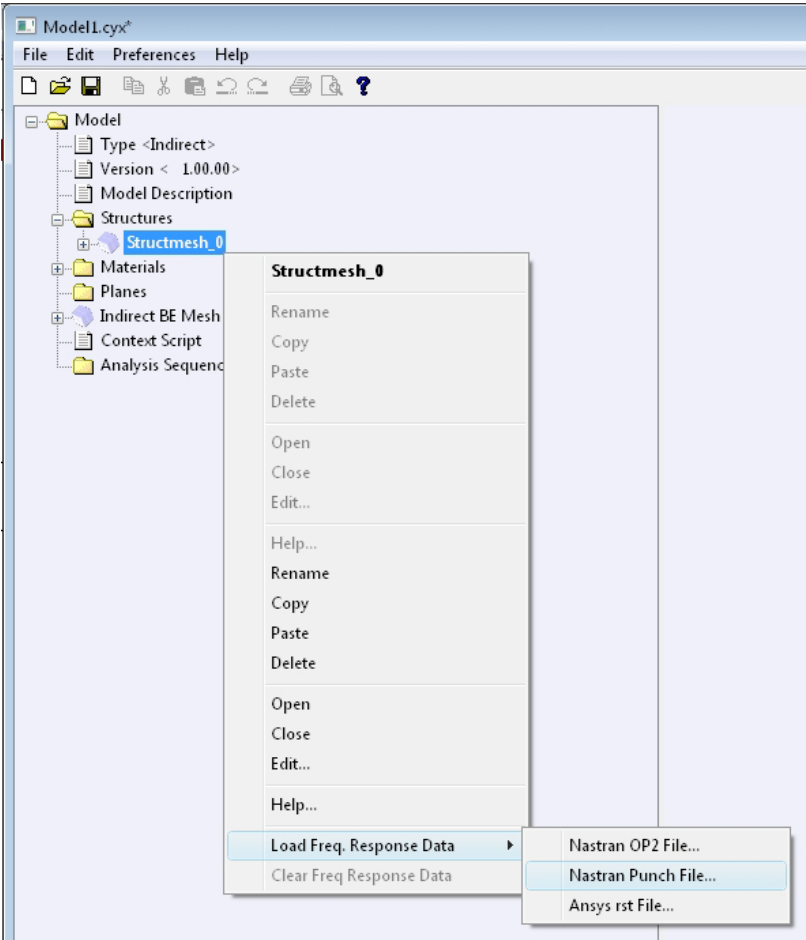
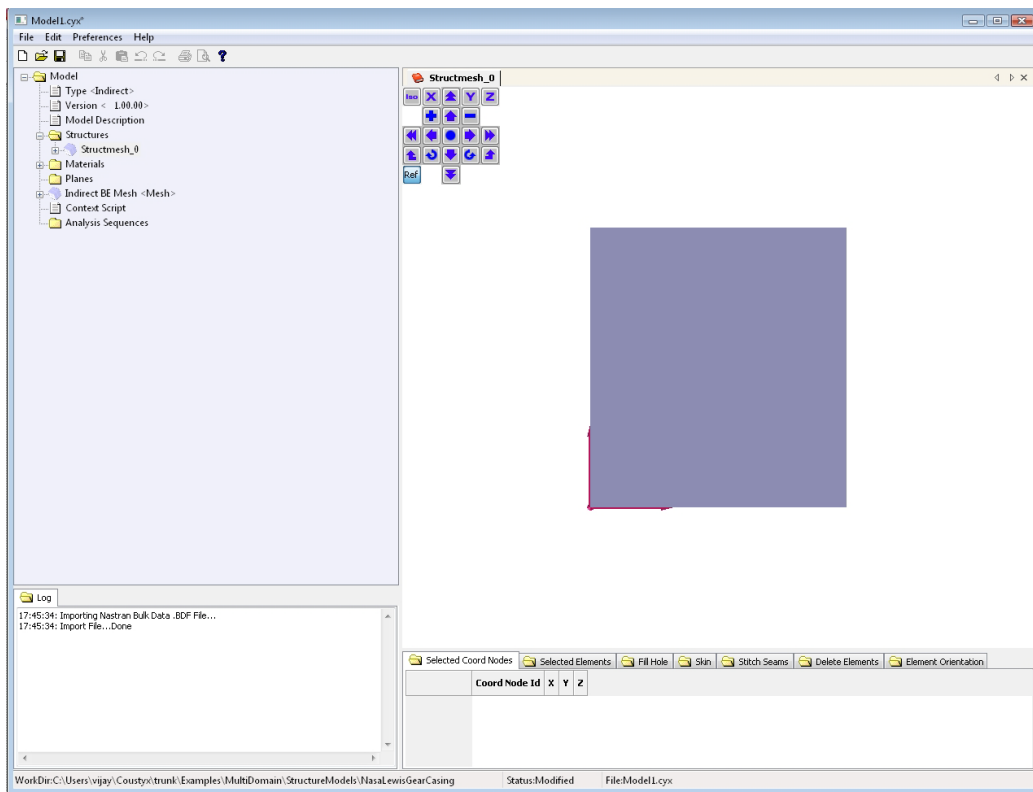


Figure A.30 Load frequency response data into Coustyx.



**Figure A.31** Structure mesh opened in Coustyx GUI.



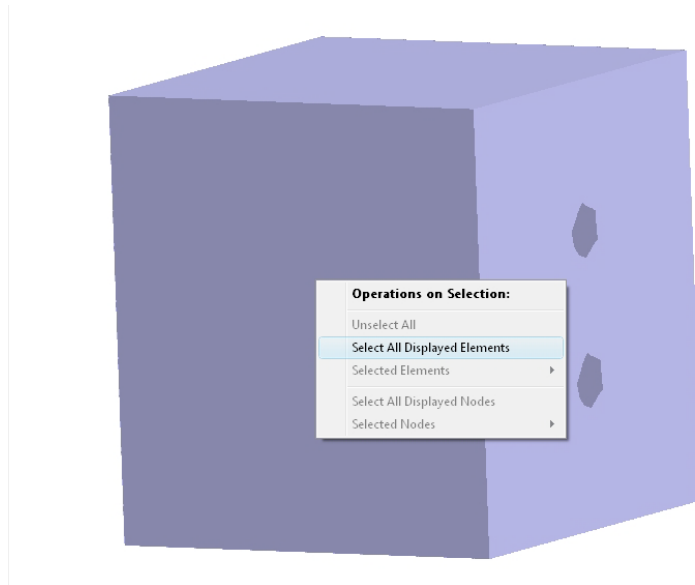
#### A.5.1.4. Generate BE Mesh

- 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 A.31.
- 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 A.32 to *zoom* and *rotate* the model.

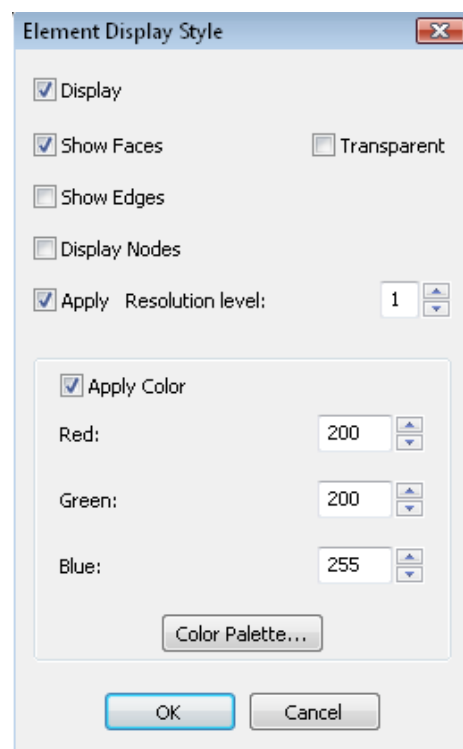


Figure A.32 Coustyx GUI control panel tools.

- 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.
- 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 A.33 and select: **Select All Displayed Elements**.
  - 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 A.34.
  - Pick the option **Show Edges** and click 'OK'.
  - 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 A.35.
  - 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 A.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 A.37.



**Figure A.33** Select all displayed elements in the GUI.



**Figure A.34** Element Display Style Window.

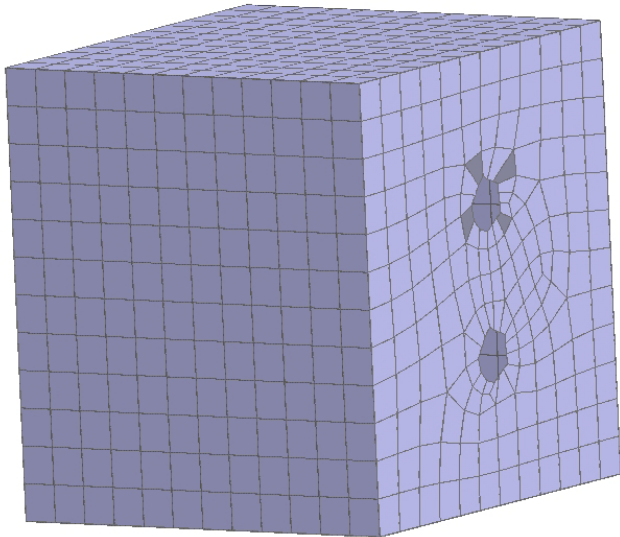


Figure A.35 Select elements for creating a seam.

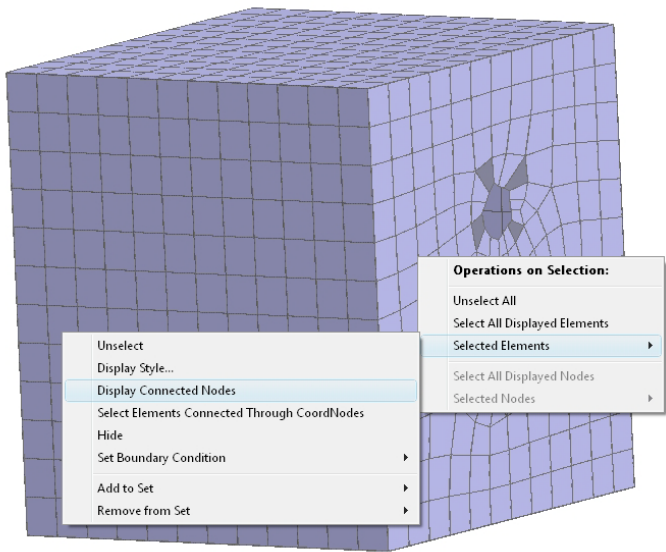
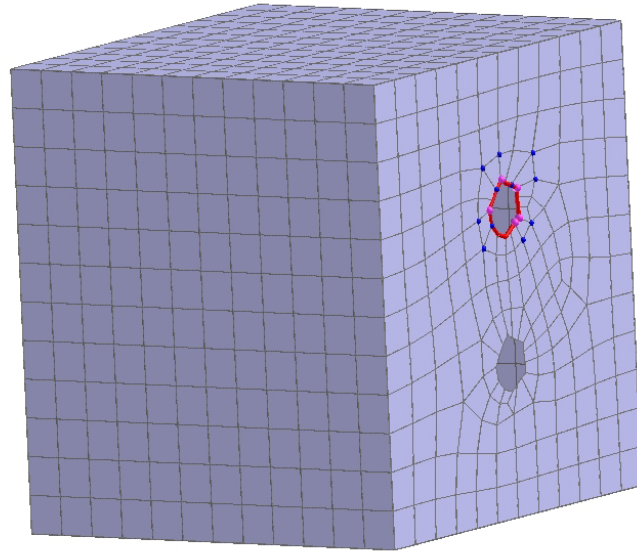


Figure A.36 Display connected nodes for creating a seam.



**Figure A.37** Pick nodes to create a seam.

- 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.
- 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.

#### **A.5.1.5. Define Material Properties**

- In the main model menu select: **Model** → **Materials** → **Air**. Right-click on **Air** and select **Edit...** Figure A.38 will appear.
- Type-in the name of the material as 'Air'.
- Define **SpeedSound** as a constant with value  $343\text{ m/s}$ . The unit is consistent with the unit of length ( $m$ ) in the structure mesh.
- Define **AmbientDensity** as a constant with value  $1.21\text{ kg/m}^3$ . The unit is consistent with the unit of length ( $m$ ) in the structure mesh.

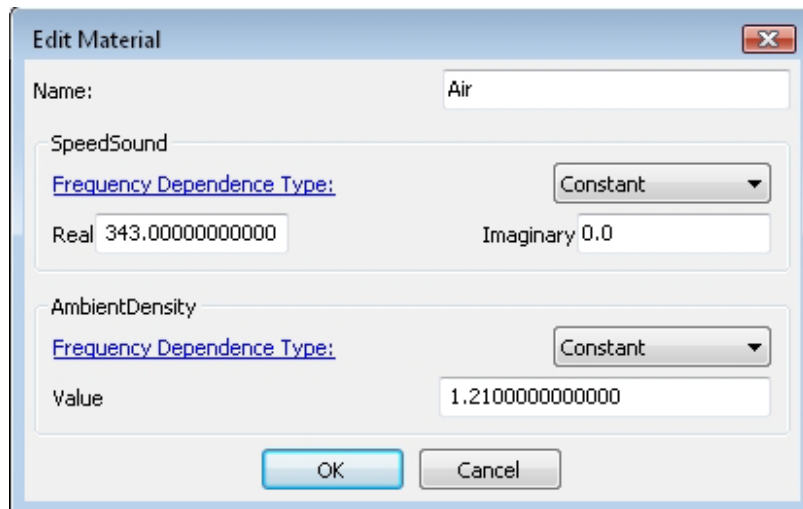


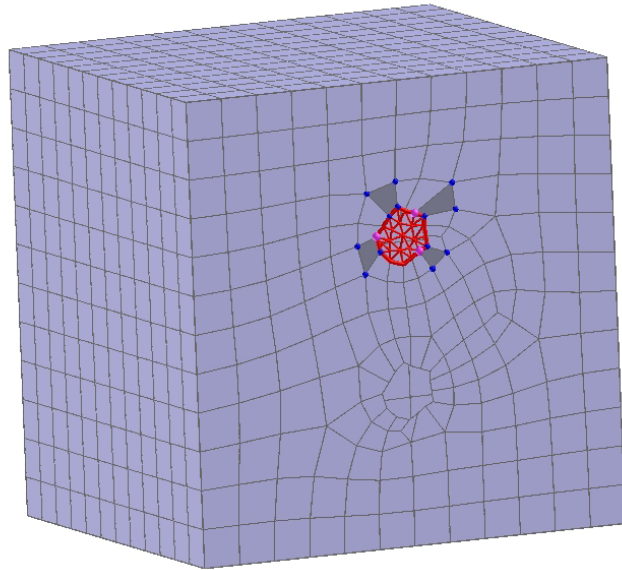
Figure A.38 Edit material properties.

#### A.5.1.6. Fill Holes

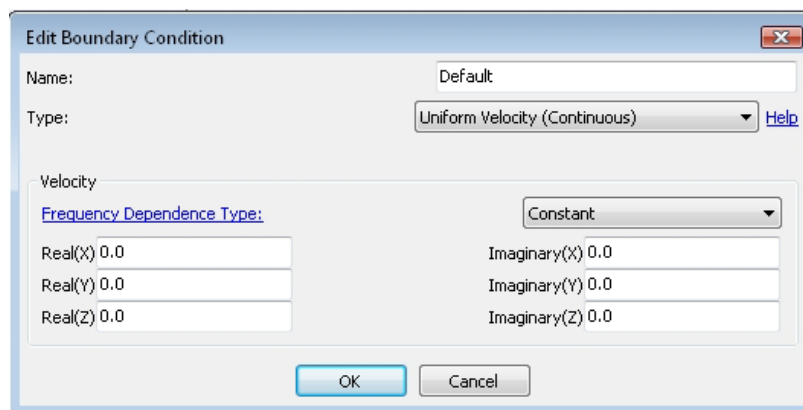
- Open the Coustyx BE mesh from the main model menu by selecting: **Model** → **Indirect BE Mesh**. Right-click on it and select: **Open** to view the boundary element mesh in the GUI.
- Select the tabbed window **Fill Hole** from the series of tabs located below the BE mesh.
- Follow the instructions given earlier in Section A.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 A.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 A.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 A.39.
- 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.

#### A.5.1.7. Define Boundary Conditions

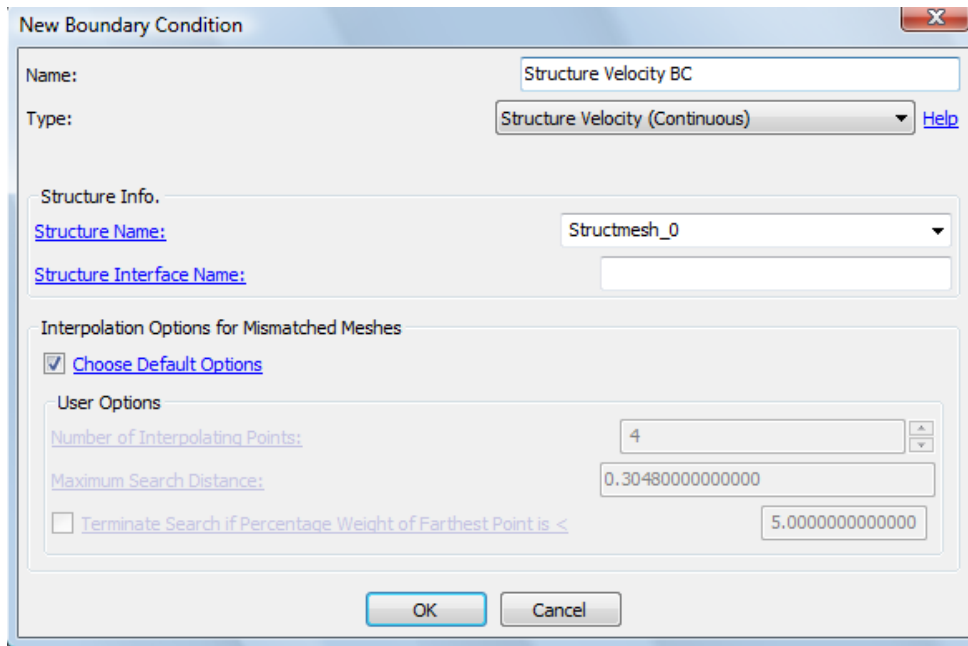
- Edit the existing default boundary condition.
  - In the Coustyx main model menu select: **Model** → **Indirect BE Mesh** → **Boundary Conditions** → **Default**.
  - Right-click on **Default** and select: **Edit...** to make changes to the default boundary conditions applied to all the boundary elements. The window in Figure A.40 will appear.
  - Type-in the new name 'Structural Velocity BC'.



**Figure A.39** Fill hole using triangle elements.



**Figure A.40** Edit boundary conditions window.

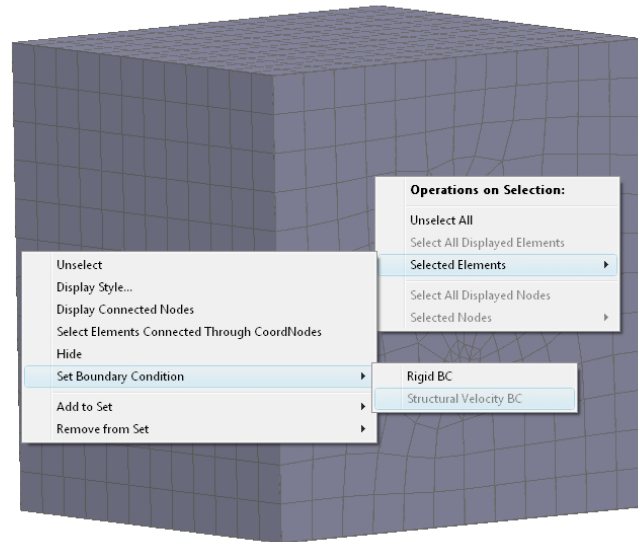


**Figure A.41** Edit structure velocity boundary condition window.

- Select **Structure Velocity (Continuous)** from the drop-down menu for ‘Type’. The window in Figure A.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’.
    - 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.

**A.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 A.42). If the boundary



**Figure A.42** Apply boundary conditions through selected elements.

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** → **Indirect BE Mesh** → **Sets** → **Holes.1**.
  - Right-click on **Holes.1** and select: **Elements** → **Set Boundary Condition** → **Rigid BC** as shown in Figure A.43.
  - Repeat the above for other holes as well.

### A.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 A.44.
  - 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 A.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)'.



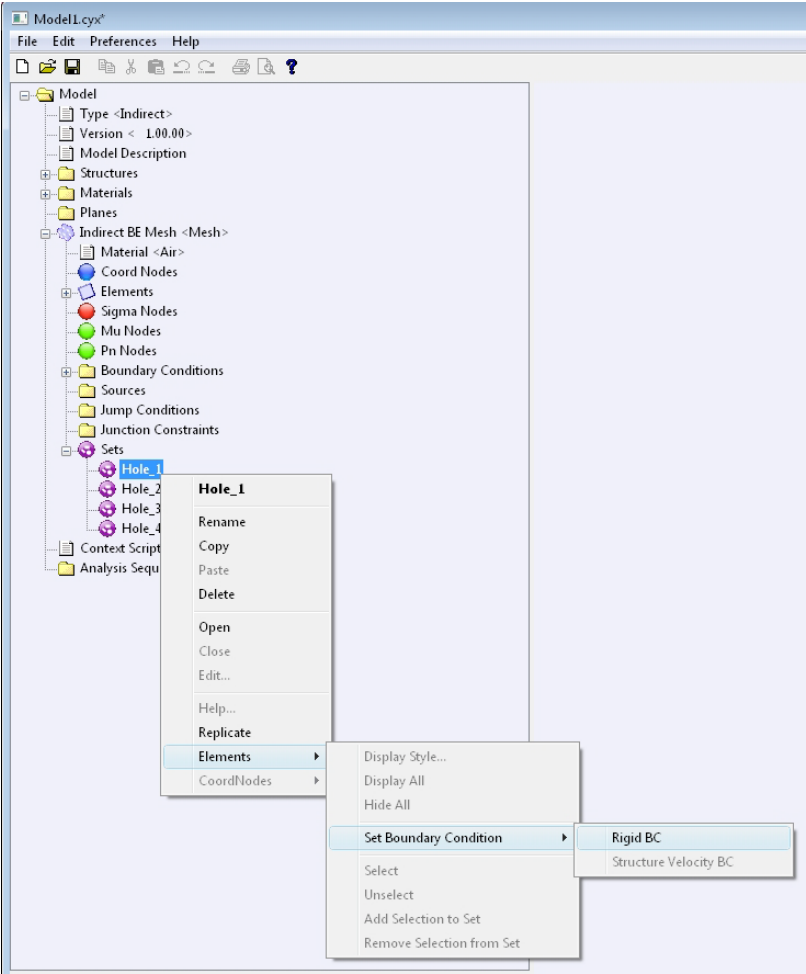


Figure A.43 Apply boundary conditions through sets.

**Analysis Sequence**

Name: Analysis Sequence

Description Solver Controls Frequency Ranges Outputs Script

**Parallel Processing**

☒ Use Maximum Possible Number of CPUs

Number of CPUs requested: 1

**FMM**

☒ Use FMM

☒ Precompute Near Field Matrices

Number of FMM Levels: 6

Solution Method: GMRES

**GMRES**

Preconditioner Type: EBE

Number Bands including Diagonal: 1

Convergence Criterion: Residual

Residual as a Percentage: 0.5000000000000000

Change in Sound Power as a Percentage: 0.5000000000000000

Number of Vectors in Krylov Subspace at Restart: 1000

Initial Guess: Previous Solution

**Integration**

☒ Use Fixed Integration Order:

Integration Order for Triangular Elements: 3

Integration Order for Quadrilateral Elements: 3

Variable Order Integration Scheme: Medium

OK Cancel

Figure A.44 Analysis solver controls.

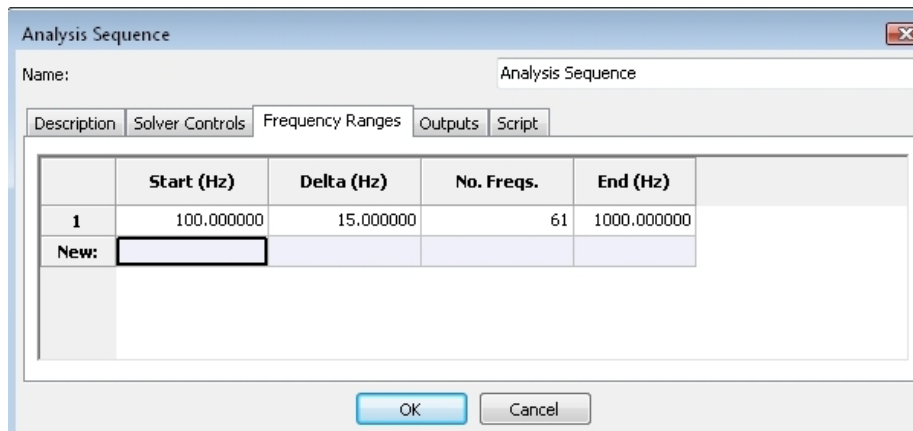


Figure A.45 Set analysis frequencies.

- 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'.

### A.5.3. Post-processing/Outputs

Coustyx creates the following output files based on the choices made in **Outputs** tab in **Analysis Sequence**.

**A.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.

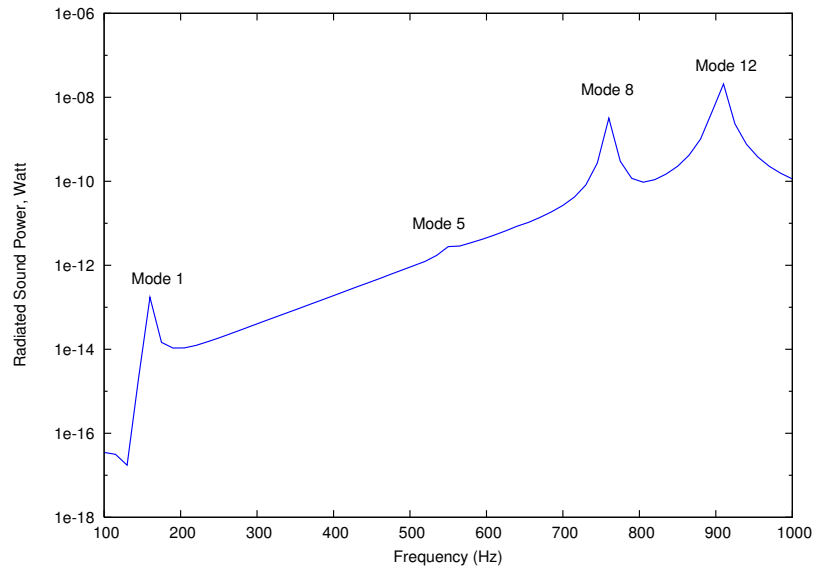
**A.5.3.2. sensors.dat** The pressure and particle velocity at the sensor locations are written into this *ASCII*-text file. Since we didn't add any sensors to the gearbox housing radiation problem, this file is empty.

**A.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.

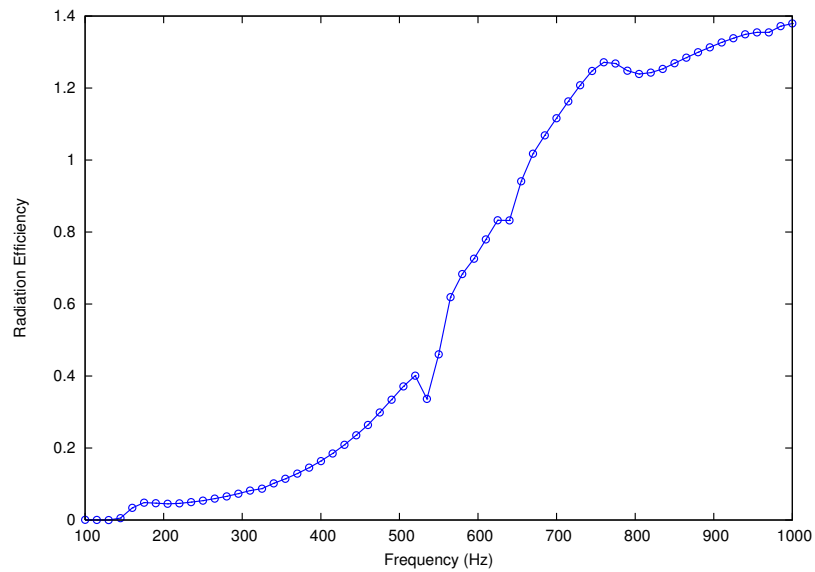
The radiated sound power and radiation efficiency are plotted against the analysis frequency in Figure A.46 and Figure A.47 using matlab plot command. The sound power radiated (Figure A.46) has peaks corresponding to the structural vibration modes that have a non-zero net volume velocity.

**A.5.3.4. iglass.cgl** IGlass files are post-processing data files created by Coustyx to visualize the acoustic analysis results. Refer to Figure A.48.

- Double-click on 'iglass.cgl' file to open it.



**Figure A.46** Sound power from the forced vibration response.



**Figure A.47** Radiation efficiency of the gearbox forced vibration.

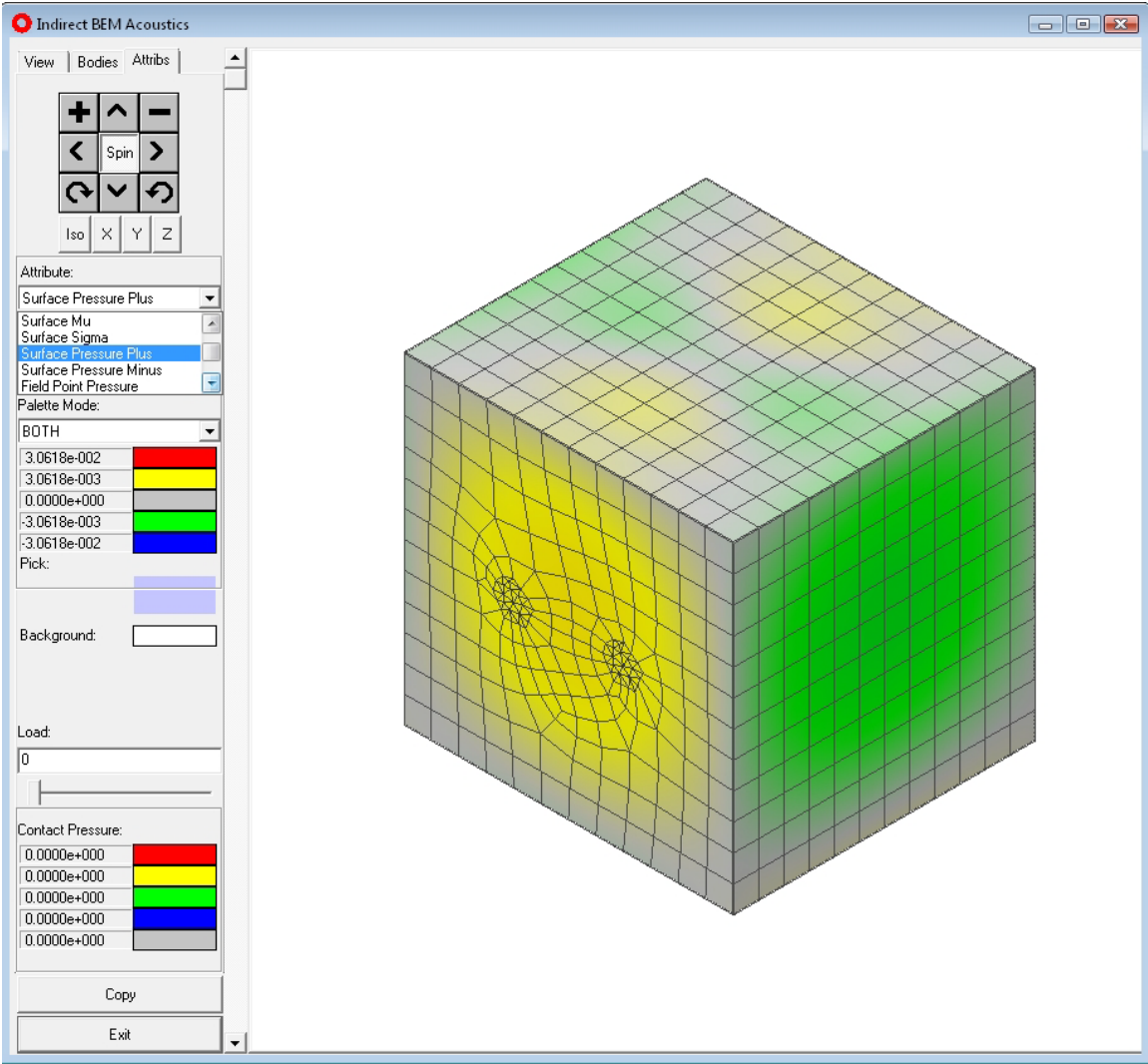


Figure A.48 IGlass viewer showing sound pressure distribution on the exterior surface of the housing at 760 Hz.

- Click on the **Attribs** tab on the top-left of the iglass viewer.
- 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**.

## REFERENCES

---

- [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.