

The Pennsylvania State University
The Graduate School

ENHANCED PROPAGATION MODELING OF DIRECTIONAL
AVIATION NOISE: A HYBRID PARABOLIC EQUATION-FAST
FIELD PROGRAM METHOD

A Dissertation in
Acoustics
by
Joyce E. Rosenbaum

© 2011 Joyce E. Rosenbaum

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2011

The dissertation of Joyce E. Rosenbaum was reviewed and approved* by the following:

Anthony A. Atchley
Professor of Acoustics
Dissertation Co-Advisor, Co-Chair of Committee

Victor W. Sparrow
Interim Chair and Professor of Acoustics
Dissertation Co-Advisor, Co-Chair of Committee

David C. Swanson
Associate Professor of Acoustics

Kenneth S. Brentner
Professor of Aerospace Engineering

*Signatures are on file in the Graduate School.

Abstract

Commercial air traffic is anticipated to increase rapidly in the coming years. The impact of aviation noise on communities surrounding airports is, therefore, a growing concern. Accurate prediction of noise can help to mitigate the impact on communities and foster smoother integration of aerospace engineering advances. The problem of accurate sound level prediction requires careful inclusion of all mechanisms that affect propagation, in addition to correct source characterization. Terrain, ground type, meteorological effects, and source directivity can have a substantial influence on the noise level. Because they are difficult to model, these effects are often included only by rough approximation.

This dissertation presents a model designed for sound propagation over uneven terrain, with mixed ground type and realistic meteorological conditions. The model is a hybrid of two numerical techniques: the parabolic equation (PE) and fast field program (FFP) methods, which allow for physics-based inclusion of propagation effects and ensure the low frequency content, a factor in community impact, is predicted accurately. Extension of the hybrid model to a pseudo-three-dimensional representation allows it to produce aviation noise contour maps in the standard form.

In order for the model to correctly characterize aviation noise sources, a method of representing arbitrary source directivity patterns was developed for the unique form of the parabolic equation starting field. With this advancement, the model can represent broadband, directional moving sound sources, traveling along user-specified paths. This work was prepared for possible use in the research version of the sound propagation module in the Federal Aviation Administration's new standard predictive tool.

Table of Contents

List of Figures	x
List of Tables	xix
Acknowledgments	xx
Chapter 1	
Introduction	1
1.1 Motivation	1
1.2 Outdoor Sound Propagation	2
1.2.1 Geometric Spreading	3
1.2.2 Atmospheric Absorption	3
1.2.3 Ground Effects	4
1.2.3.1 Ground Impedance	4
1.2.3.2 Ground Impedance Discontinuities	8
1.2.4 Meteorological Effects	8
1.2.4.1 Atmospheric Refraction	9
1.2.4.2 Atmospheric Turbulence	11
1.2.5 Terrain Effects	12
1.2.6 Source Characteristics	14
1.3 Standard and State of the Art Aviation Noise Models	15
1.3.1 Existing Aviation Noise Models	16
1.3.2 State of the Art in Propagation Modeling	17
1.3.2.1 Ray Model	18
1.3.2.2 Green’s Function Parabolic Equation Model	18
1.3.2.3 Crank-Nicholson Parabolic Equation Model	19
1.3.2.4 Fast Field Program	19

1.4	Research Objectives and Outline	20
Chapter 2		
	The Two-Dimensional Parabolic Equation Model	22
2.1	Introduction	22
2.2	Derivation of the Parabolic Equation	23
2.2.1	Crank-Nicholson Parabolic Equation Model	24
2.2.1.1	Grid Boundary Conditions	30
2.2.1.1.1	Lower Grid Boundary	30
2.2.1.1.2	Upper Grid Absorbing Layer	31
2.2.2	Generalized Terrain Parabolic Equation Model	32
2.2.2.1	Grid Boundary Conditions	36
2.3	Numerical Implementation	42
2.3.1	Grid Parameters	42
2.3.2	Starting Field	44
2.3.2.1	Narrow-Angle Starting Field	45
2.3.2.2	Wide-Angle Starting Field	47
2.3.2.3	Sources Near the Ground	49
2.3.3	Atmospheric Absorption	50
2.3.4	Refractive Atmosphere	51
2.3.5	Turbulence	52
2.3.6	Ground Impedance	56
2.3.7	GTPE-Specific Numerical Implementation	56
2.3.7.1	Terrain	56
2.3.7.2	First Three Range Steps	57
2.4	Numerical Examples	57
2.5	Conclusions	58
Chapter 3		
	The Two-Dimensional Fast Field Program Model	60
3.1	Introduction	60
3.2	FFP Derivation	61
3.2.1	FFP Method Solution of the Horizontal Wave Number Do- main Helmholtz Equation	63
3.2.1.1	Inter-Layer Boundary Conditions	64
3.2.1.2	Intra-Layer Solution	65
3.2.1.3	Calculating the Sound Field with the FFP	68
3.2.1.3.1	Assuming an Arbitrary Solution at the Ground	68
3.2.1.3.2	Extrapolating Up From the Ground	69

3.2.1.3.3	Assuming an Arbitrary Solution at the Top of the Grid	69
3.2.1.3.4	Extrapolating Down From the Top of the Ground	70
3.2.1.3.5	Calculating Correct Sound Field Results	70
3.2.2	Numerical Considerations of the Inverse Transform Evaluation	71
3.3	Numerical Implementation	73
3.3.1	Grid Parameters	73
3.3.1.1	Grid Point Spacing	74
3.3.2	Atmospheric Absorption, Refractive Atmosphere, and Ground Impedance	75
3.4	Conclusions	75

Chapter 4

	Hybrid PE-FFP Model	76
4.1	Introduction	76
4.2	Procedure for Model Combination	76
4.3	Validation and Verification	78
4.3.1	Verification: Individual Model Comparisons with Published and Accepted Results	78
4.3.2	Validation: Near Ground Operation Test Cases	78
4.3.2.1	Description of Test Cases	79
4.3.2.2	Numerical Implementation of the Test Cases in the Hybrid Model	80
4.3.2.2.1	PE Start Field	80
4.3.2.2.2	Grid Step Spacing	80
4.3.2.2.3	Atmospheric Absorption	80
4.3.2.2.4	Refractive Atmosphere	81
4.3.2.2.5	Terrain	81
4.3.2.2.6	Propagation Range	81
4.3.2.3	Test Case Results	82
4.3.2.4	Comparison with NMSim	95
4.3.2.4.1	Test Case 1	97
4.3.2.4.2	Test Case 3	97
4.3.2.4.3	Test Case 4	99
4.3.2.5	Comparison with the INM Line-Of-Sight Adjustment	101
4.4	Conclusions	102

Chapter 5	
Pseudo-Three-Dimensional Hybrid PE-FFP Model	109
5.1 Introduction	109
5.2 Summary of the Two- to Three-Dimension Expansion Process . .	110
5.3 Propagation from a Single Point Source	112
5.4 Propagation from a Flight Path	114
5.5 Numerical Examples of Contouring with Range-Dependent Effects	118
5.5.1 Results from Flat Terrain Case	119
5.5.2 Results from Uneven Terrain Case	120
5.6 Conclusion	122
Chapter 6	
Inclusion of Elevation Angle Source Directivity in the Parabolic Equation Method with an Inverse Fourier Transform Technique	124
6.1 Introduction	124
6.2 Array Synthesis for Directional PE Sources	126
6.3 Two Approaches for the Generalized Source Directivity Technique	129
6.3.1 Approach 1: Synthesize Array \rightarrow Extend Resolution . . .	131
6.3.1.1 Numerical Considerations for Approach 1	133
6.3.1.2 Numerical Examples with Modified Approach 1 .	136
6.3.2 Approach 2: Extend Resolution \rightarrow Synthesize Array . . .	139
6.3.2.1 Defining the Directional Function	139
6.3.2.2 Calculating the Starting Field	140
6.3.2.3 Numerical Examples with Approach 2	144
6.4 Comparison of Approaches 1 and 2	145
6.5 Conclusions	146
Chapter 7	
Conclusion	151
7.1 Summary	151
7.2 Future Work	154
Appendix A	
Existing Standard Aviation Noise Models	156
A.1 Introduction	156
A.2 Source Representations	157
A.2.1 INM	157
A.2.2 NOISEMAP	157
A.2.3 AAM	157

A.3	Propagation Algorithm Adjustments	158
A.3.1	INM	158
A.3.1.1	Noise Fraction	160
A.3.1.2	Lateral Attenuation	161
A.3.1.3	Line-of-Sight Blockage	163
A.3.2	NOISEMAP	165
A.3.3	AAM	166
A.4	Discussion of Algorithms	168
A.4.1	Source Representation	168
A.4.2	Segmentation Model Adjustments	169
A.4.2.1	NPD-specific adjustments	169
A.4.2.2	Lateral attenuation	169
A.4.3	Physical Propagation Effects	170
A.4.3.1	Uneven terrain	170
A.4.3.2	Ground Effects	170
A.4.3.3	Meteorological Effects	171
A.5	Conclusion	171

Appendix B

	Guidance on Full Model-Fast Access Model Conversion	172
B.1	Introduction	172
B.2	Meteorology	172
B.2.1	Refractive Atmosphere Classes	172
B.2.2	Turbulence	175
B.3	Ground Impedance Discontinuities	176
B.4	Terrain	177
B.5	Conclusion	178

Appendix C

	Assessment of the Potential for Using Emissions Meteorological Models for Noise Propagation	179
C.1	Introduction	179
C.2	Model Requirements	179
C.2.1	Refractive Sound Speed Profiles	180
C.2.2	Atmospheric Turbulence	180
C.3	AERMOD Outputs Transformed to Noise Propagation Model Inputs	181
C.3.1	Refractive Atmosphere Parameters	181
C.3.2	Turbulence Parameters	183
C.4	Conclusion	185

Appendix D	
Hybrid PE-FFP Model Code Diagram	186
D.1 Description of the Diagram	186
Appendix E	
Hybrid PE-FFP Model Code	193
E.1 Description of the Code Presentation	193
Appendix F	
Benchmarking the PE and FFP Models	269
F.1 Introduction	269
F.2 Comparisons for the PE and FFP Under Different Sound Speed Profile Conditions	269
F.3 Comparison of PE Under Turbulent Conditions	278
Bibliography	279

List of Figures

1.1	Geometry of source and receiver geometry above a ground surface, after [1] Figure D.1.	7
1.2	Refraction caused by themal gradients, after [2]	10
1.3	Refraction caused by wind gradients, after [2]	10
1.4	Multiple ray paths between the source and receiver from [3]. Re-produced with permission.	11
1.5	A shadow zone is formed behind a hill, after [1] Figure 5.1.	13
1.6	Parallels between terrain features and atmospheric refraction: A hill corresponds to a upward refraction atmosphere and a valley corresponds to a downward refracting atmosphere, after [1] Figure 6.3.	13
2.1	Diagram of PE propagation method: begin with the initial field and march forward in range, extrapolating from the previous range step.	24
2.2	Wave number break-down into horizontal and vertical components at different angles from a monopole source.	26
2.3	Diagram of plane wave propagating at a small angle θ from the horizontal.	27
2.4	Coordinate transformation used in the GTPE, after [1] Figure M.4.	32
2.5	Diagram of the ground angle in the GTPE grid, after [1] Figure M.5.	37
2.6	Diagram of grid spacing to common points for all frequencies.	43
2.7	Magnitude of the pressure plotted against height for a start field representing a 10 Hz source at 5 m height above a ground with a normalized impedance of $Z = 38.79 + i38.41$. The dashed red line represents the source, the dash-dot cyan line represents the image source, and the solid dark blue line shows the combined contribution of both components.	51

2.8	Sound pressure level relative to free field [dB] for 100 Hz source at 20 m height calculated with PE. (a) Homogeneous atmosphere, flat, rigid ground. (b) Homogeneous atmosphere, rigid ground with hill-peak at range of 400 m, height of 30 m, and width of 500 m. (c) Homogeneous atmosphere, flat ground with $\sigma = \infty$ for range 0-300 m and $\sigma = 150$ cgs Rayls for range 300-1000 m. σ is the effective flow resistivity. (d) Refractive atmosphere: $b = -1$ m/s, flat, rigid ground. b is the parameter of logarithmic sound speed profile. (e) Refractive atmosphere: $b = -1$ m/s, flat, rigid ground, 50 realizations of turbulence, $C_T^2/T_0^2 = 6 * 10^{-7} m^{-2/3}$, $C_v^2/c_0^2 = 2 * 10^{-6} m^{-2/3}$. C_T^2 and C_v^2 are the structure parameters of turbulent temperature and wind velocity fluctuations, respectively.	59
3.1	The atmosphere in the FFP method is modeled as a combination of horizontally homogeneous layers within which the wave number is constant, after [1] Figure F.1.	64
3.2	Integration path for the inverse transform integral that avoids poles at $k_r = \pm k$, after [1] Figure F.2.	71
4.1	Method of combining the PE and FFP methods into the hybrid model. Upper Left Figure: contribution of the FFP to the vertical two-dimensional sound field, retaining data at elevation angles greater than 35° from the source. Upper Right Figure: contribution of the PE, retaining data at elevation angles of 35° and smaller. Bottom Left Figure: PE-FFP union. Bottom Right Figure: PE-FFP union at a height of 2 m over the ground. The arrow marks the transition from FFP to PE data.	77
4.2	Smoothed terrain geometry plotted against discontinuous-slope terrain, Test Case 5.	81
4.3	Results for Test Case 1, flat, soft ground. The first, second, and third columns show results for 50, 500, and 2500 ft source heights, respectively. The first, second, and third row show results for 100, 800, and 4000 Hz source, respectively. The fourth row shows averaged results for frequencies 50-1000 Hz in 1/3-octave bands and 2000-8000 Hz in octave bands for the 50 and 500 ft source heights. Geometries of the test cases are illustrated below the result figures. Note the different range used for the 2500 ft source.	89

4.9	Results for Test Case 4, downward sloping, soft ground, in a homogeneous atmosphere with and without terrain smoothing. The first and second columns show results for 50 and 500 ft source heights, respectively. The first and second rows show results for 100 and 800 Hz sources, respectively. Geometries of the test cases are illustrated below the result figures. The dash-dot lines show results for which the terrain profile was represented as a series of straight line segments and are compared to the solid lines, which show results for a smoothed terrain profile.	95
4.10	Comparison of Hybrid Model and NMSim results for Test Case 1, with (a) 100 Hz and (b) 4000 Hz sources. The solid line shows the Hybrid Model results, the dots show NMSim results, and the dash line shows spherical spreading and atmospheric absorption. Geometries of the test cases are illustrated below the result figures.	103
4.11	Screenshot of the NMSim run environment for Test Case 3. A cross section of the upward sloping terrain geometry and ground impedance specification are shown at top. The black line shows the terrain profile and the red line draws a straight connection between the source and the currently selected receiver. It is represented in red here because the path is obstructed by the terrain. A time history of level at a specific receiver location is shown in the center window. A top down, elevation contour view of the upward sloping terrain is shown with receiver locations at bottom.	104
4.12	Comparison of Hybrid Model and NMSim results for Test Case 3, with (a) 100 Hz and (b) 4000 Hz sources. The solid line shows the Hybrid Model results, the dots show NMSim results, and the dash line shows spherical spreading and atmospheric absorption. Geometries of the test cases are illustrated below the result figures.	105
4.13	Screenshot of the NMSim run environment for Test Case 4. A cross section of the downward sloping terrain geometry and ground impedance specification are shown at top. The black line shows the terrain profile and the red line draws a straight connection between the source and the currently selected receiver. It is represented in red here because the path is obstructed by the terrain. A time history of level at a specific receiver location is shown in the center window. A top down, elevation contour view of the downward sloping terrain is shown with receiver locations at bottom.	106

4.14	Comparison of Hybrid Model and NMSim results for Test Case 4, with (a) 100 Hz and (b) 4000 Hz sources. The solid line shows the Hybrid Model results, the dots show NMSim results, and the dash line shows spherical spreading and atmospheric absorption. Geometries of the test cases are illustrated below the result figures.	107
4.15	Comparison of Hybrid Model and NMSim results for Test Case 4 with a 100 Hz in a turbulent atmosphere with normalized structure parameters $C_T^2/T_0^2 = 6 \times 10^{-7} \text{ m}^{-2/3}$ and $C_v^2/c_0^2 = 2 \times 10^{-6} \text{ m}^{-2/3}$. The solid line shows the Hybrid Model results without turbulence, the dots show NMSim results, the dash-dot line shows the Hybrid Model results with turbulence, and the dash line shows spherical spreading and atmospheric absorption. The geometry of the test case is illustrated below the result figure.	108
5.1	Radial systems centered at each source point along a flight path, superimposed on the Cartesian coordinate results grid.	111
5.2	Vertical flight profile in the $x - z$ plane: symbols represent locations of 21 point sources along a sample flight profile.	113
5.3	Radial run results are defined in a cylindrical coordinate system.	114
5.4	Polar gridpoints calculated with hybrid model (x) and interpolated Cartesian gridpoints (+).	114
5.5	Transmission loss [dB] contours for 50 Hz source at 100 m height, homogeneous atmosphere, flat, soft ground with $\sigma = 200$ cgs rayls, receiver height of 2 m, calculated with hybrid model.	115
5.6	Transmission loss [dB] for a 50 Hz source following a given flight profile at a y -range of 0 m left to right, homogeneous atmosphere, flat, soft ground with $\sigma = 200$ cgs rayls, calculated with the hybrid model. (a) Sound field. (b) Contours.	116
5.7	Transmission loss [dB] for a broadband source 50-2000 Hz, 1/9-octave bands, following a given flight profile at a y -range of 0 m left to right, homogeneous atmosphere, flat, soft ground with $\sigma = 200$ cgs rayls, calculated with the analytical model. (a) Sound field. (b) Contours.	117
5.8	Geometry of Scenario 1, Flat Terrain. Hard ground, shown in tan, effective flow resistivity, $\sigma = 20,000$ cgs rayls from $y = -30$ m to 30 m. Soft ground, shown in green, effective flow resistivity, $\sigma = 150$ cgs rayls from $y = 30$ m to 320 m. Source path, shown in light blue, is level to the ground at 5 m height until $x = -90$ m, then increases height at 10° angle. Dark blue stem plot shows representative source locations.	119

5.9	Geometry of Scenario 2, Uneven Terrain. Hard ground, shown in tan, effective flow resistivity, $\sigma = 20,000$ cgs rayls from $y = -30$ m to 30 m. Soft ground, shown in green, effective flow resistivity, $\sigma = 150$ cgs rayls from $y = 30$ m to 320 m. Source path, shown in light blue, is level to the ground at 5 m height until $x = -90$ m, then increases height at 10° angle. Dark blue stem plot shows representative source locations. Hill spans $y = 100$ m to 200 m with peak height 10 m at $y = 150$ m.	120
5.10	Transmission Loss Results of Scenario 1, Flat Terrain, in dB. Contours are shown as a bandwidth-weighted average of 1/3-octave band center frequency results from 50-500 Hz. Propagation geometry is as shown in Figure 5.8. Region between dashed lines is outside valid PE elevation angle range.	121
5.11	Transmission Loss Results of Scenario 1, Flat Terrain, in dB. Contours are shown for a single 500 Hz source at (0,0,15.9 m). Propagation geometry is as shown in Figure 5.8.	122
5.12	Transmission Loss Results of Scenario 2, Uneven Terrain, in dB. Contours are shown as a bandwidth-weighted average of 1/3-octave band center frequency results from 50-500 Hz. Propagation geometry is as shown in Figure 5.9. Region between dashed lines is outside valid PE elevation angle range.	123
6.1	Geometry of the starting field grid parameters. The shaded circles represent monopole elements of the source array.	127
6.2	Sound field, in dB, produced by a (a) horizontal and (b) vertical dipole in free field, calculated with an analytical model. The source is at 1500 m height.	130
6.3	Sound field, in dB, produced by a (a) horizontal and (b) vertical dipole 4 m above a rigid ground, calculated with an analytical model.	132
6.4	Sound field, in dB, produced by a (a) horizontal and (b) vertical dipole high above a soft ground, calculated with Approach 1. The source is at a height of 1500 m.	134
6.5	Absolute difference in the sound fields of a free field (a) horizontal and (b) vertical dipole in dB, calculated with Approach 1 and the analytical model. The source is at a height of 1500 m. The dashed line indicates the elevation angles of $\pm 35^\circ$, within which the PE results are expected to be valid. All absolute differences of 1 dB and larger are shown in white.	135

6.6	Absolute difference in the sound fields of a free field (a) horizontal and (b) vertical dipole in dB, calculated with modified Approach 1 and the analytical model. The source is at a height of 1500 m. The dashed line indicates the elevation angles of $\pm 35^\circ$, within which the PE results are expected to be valid. All absolute differences of 1 dB and larger are shown in white.	137
6.7	Absolute difference in the sound fields of (a) horizontal and (b) vertical dipole 4 m above a rigid ground, in dB, calculated with modified Approach 1 and the analytical model. The dashed line indicates the elevation angles of $\pm 35^\circ$, within which the PE results are expected to be valid. All absolute differences of 1 dB and larger in (a) and 3 dB and larger in (b) are shown in white.	138
6.8	Process of directional function calculation for Approach 2 for (a) horizontal and (b) vertical dipole sources.	147
6.9	Absolute difference in the sound fields of a free field (a) horizontal and (b) vertical dipole in dB, calculated with Approach 2 and the analytical model. The source is at a height of 1500 m. The dashed line indicates the elevation angles of $\pm 35^\circ$, within which the PE results are expected to be valid. All absolute differences of 1 dB and larger are shown in white.	148
6.10	Absolute difference in the sound fields of (a) horizontal and (b) vertical dipole 4 m above a rigid ground, in dB, calculated with Approach 2 and the analytical model. The dashed line indicates the elevation angles of $\pm 35^\circ$, within which the PE results are expected to be valid. All absolute differences of 1 dB and larger are shown in white.	149
6.11	Sound pressure level relative to spherical spreading, in dB, at a receiver height of 1.22 m for (a) horizontal and (b) vertical dipoles over a rigid ground. Lines <i>a</i> and <i>b</i> , <i>c</i> and <i>d</i> , and <i>e</i> and <i>f</i> represent sources at heights of 4, 60, and 200 m, respectively. Lines <i>a</i> , <i>c</i> , <i>e</i> and <i>b</i> , <i>d</i> , <i>f</i> are calculated with an analytical model and Approach 2, respectively. Lines <i>a</i> and <i>b</i> are largely indistinguishable in both figures.	150
A.1	Diagram of the elevation angle for helicopter lateral directivity adjustment. (Figure appeared as Figure 3-15 in the INM 7.0 Technical Manual [4]. Reproduced with permission.)	159
A.2	Diagram of the observer/flight-path geometry. (Figure appeared as Figure C-1 in the INM 7.0 Technical Manual [4]. Reproduced with permission)	161

A.3	Diagram of the lateral attenuation geometry. (Figure appeared as Figure 3-5 in the INM 7.0 Technical Manual [4]. Reproduced with permission.)	162
A.4	Directivity of the engine-installation effects for wing-mounted and fuselage-mounted engines of jet-powered airplanes. (Figure appeared as Figure 3-7 in the INM 7.0 Technical Manual [4]. Reproduced with permission.)	163
A.5	Diagram of the line-of-sight (LOS) blockage geometry. (Figure appeared as Figure 3-12 in the INM 7.0 Technical Manual [4]. Reproduced with permission.)	164
B.1	Scheme of meteorological model used in [5]. Reproduced with permission.	174
B.2	Scheme of long-term average transfer function computation used in [5]. Reproduced with permission.	175
B.3	Transfer function versus range for propagation in a downward refracting atmosphere over homogeneous water and grass (thick lines) and over various grounds with a grass-water transition (thin lines). Thin solid lines were calculated with a single PE run, using the exact grass-water configuration. Thin dashed lines were calculated using the interpolation method to combine homogeneous grass and homogeneous water PE runs [6]. Reproduced with permission. . .	177
C.1	Example of a von Kármán spectrum, after [1] Figure I.3. The von Kármán spectrum parameters used are $K_0^{-1} = 10$ m, $\frac{C_T^2}{T_0} = 6 * 10^{-7}$ m ^{-2/3} , $\frac{C_v^2}{c_0^2} = 2 * 10^{-6}$ m ^{-2/3} . The energy-containing, inertial, and dissipative subranges are shown.	184
F.1	Comparison of the PE module with benchmark cases from reference [7] for Case 1, a homogeneous atmosphere. The PE results of this research, shown in green, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.	270
F.2	Comparison of the PE module with benchmark cases from reference [7] for Case 2, a downward refracting atmosphere. The PE results of this research, shown in green, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.	271

F.3	Comparison of the PE module with benchmark cases from reference [7] for Case 3, an upward refracting atmosphere. The PE results of this research, shown in green, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.	272
F.4	Comparison of the PE module with benchmark cases from reference [7] for Case 4, a ducting atmosphere. The PE results of this research, shown in green, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.	273
F.5	Comparison of the FFP module with benchmark cases from reference [7] for Case 1, a homogeneous atmosphere. The FFP results of this research, shown in green or red, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.	274
F.6	Comparison of the FFP module with benchmark cases from reference [7] for Case 2, a downward refracting atmosphere. The FFP results of this research, shown in red, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.	275
F.7	Comparison of the FFP module with benchmark cases from reference [7] for Case 3, an upward refracting atmosphere. The FFP results of this research, shown in red, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.	276
F.8	Comparison of the FFP module with benchmark cases from reference [7] for Case 4, a ducting atmosphere. The FFP results of this research, shown in red, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.	277
F.9	Comparison of PE results of sound pressure level relative to a source in the free field including the effects of turbulence from reference [1] and the PE model used in this research. Source height = 2 m, receiver height = 2 m, frequency = 500 Hz, $b = -1$ m/s, rigid ground. Average over 50 realizations of turbulence, $K_0^{-1} = 10$ m, $C_T^2/T_0^2 = 2.5 \times 10^{-7} \text{ m}^{-2/3}$, and $C_v^2/c_0^2 = 1 \times 10^{-6} \text{ m}^{-2/3}$ (a) Results after reference [1] Figure 5.7. (b) Results of coded PE model used in this research.	278

List of Tables

1.1	Effective flow resistivities as calculated by comparing predicted and measured spectra, from [3], originally appeared in [8]. Reproduced with permission.	6
4.1	Description of test case geometries for the 50 and 500 ft source heights. 87	
4.2	Maximum horizontal grid step spacing, by case number and source frequency. 88	
4.3	Atmospheric absorption coefficients of the SAE-ARP-866A [9] and ANSI S1.26 [10] standards, used in INM and NMSim, respectively. 25 °C air temperature and 70% relative humidity. 98	

Acknowledgments

I very gratefully acknowledge my advisors, Dr. Atchley and Dr. Sparrow. Dr. Atchley not only guided me through my doctoral program as a true mentor, helping me think deeply about my research and opening opportunities, but his class helped impart something I will keep forever—Lee. Dr. Sparrow’s knowledge of the field was a huge resource and his advice, invaluable. Their genuine concern for and encouragement of my professional and personal growth have been unwavering.

I would like to thank my committee members, Dr. Swanson and Dr. Brentner for their valuable insights, and the different perspectives they brought at various stages of my progress. Thank you, also, to Dr. Gabrielson whose thoughtful suggestions and classes helped me get through some of my toughest problems.

This research was funded by the Office of Environment and Energy, U.S. Federal Aviation Administration under Contract FAA 07-C-NE-PSU, and U.S. Department of Transportation Volpe National Transportation Systems Center. I would like to acknowledge the support of the FAA/NASA/Transport-Canada PARTNER Center of Excellence (www.partner.aero) and Penn State University. A special thanks to Eric Boeker and Bill He for their guidance in this research.

I thank my friends in State College and beyond who kept up the encouragement, especially Jes for my pre-candidacies care package and Running Buddy Becky for motivating some much needed endorphins. Acousticians of the Research West lab (resident and distant)—Alexandre, Amanda, Andy, Beom Soo, Denise, Joe, Kieran, Kim, Sang, Whitney—and graduated labmate, Ki Won, thank you for the discussions and digressions. Thank you to my Starbucks buddies, Rege and Ron.

To my parents, who always believe in me and inspire me to do my best—Mom who meticulously edits writing in languages she doesn’t understand and Dad who listens about my research and works to understand—thank you for your unconditional love.

Lee, since I met you, you have been a tremendous source of support, love, and strength. You inspired me with your successes and patiently stuck by me while I finished up. Now I’m graduating and ready.

Introduction

1.1 Motivation

For people who live and work in close proximity to airports, aviation noise can be an unwelcome disturbance. Its impact on communities has been of concern since the late 1950s, when high volumes of jet aircraft were used for commercial services. Soon after, regulations were established to help control and restrict the noise [3]. Today, the Federal Aviation Administration predicts a 60% increase in operations at the nation's 35 busiest airports over the next twenty years [11]. Accordingly, concerns about aviation noise are more important than ever.

Methods of accurately predicting sound levels near airports can help to minimize the impact of aviation noise on communities by facilitating compatible land use management, and informing airport expansion projects. They can also help direct mitigation efforts to areas most in need. For these reasons, viable numerical models are desirable.

The Integrated Noise Model (INM) is the current American standard in numerical aviation noise propagation. Employed by the Federal Aviation Administration since 1978, INM is used by more than 1000 companies in more than 65 countries around the world [12]. INM relies on an extensive database of source level information, for a vast variety of aircraft, under reference conditions. INM algorithms then adjust the reference levels to account for realistic flight and propagation conditions. While numerous effects are included, simplified approximations are used for many in the interest of computational speed.

More sophisticated algorithms based on ray tracing have been implemented in some existing transportation noise models. Such models, for example the recently developed Advanced Acoustic Model (AAM) for military air base noise [13], can employ more detailed representations of the noise source and account, more accurately, for a larger number of noise propagation effects. However, they use high frequency assumptions that may not accurately capture the behavior of low frequency sound.

The research discussed in this dissertation is based on two popular methods of propagation called the parabolic equation (PE) model and the fast field program (FFP) which, together, can accommodate a wide range of propagation effects for low to high frequency sound. They are joined to form a hybrid propagation model. This chapter describes the mechanisms of outdoor sound propagation, including an introduction to established practices of quantifying and modeling their effects. It then provides a summary of the methodology of both the existing standard aviation noise models as well the state of the art in propagation modeling. Finally, it presents the goals and approach of this research and outlines the remainder of the dissertation.

1.2 Outdoor Sound Propagation

Aviation noise moving through outdoor spaces is acted on by a variety of different propagation mechanisms. Effects such as geometric spreading, atmospheric absorption, ground impedance, meteorology, and terrain effects may have significant influence on the amount of sound that reaches nearby communities. Extensive research in the field of outdoor sound propagation has provided methods of quantitatively modeling and including these effects in predictive models. A brief description of each effect, with a focus on methods from the published literature for quantifying them, is presented in this section. Some references will be made to specific noise models. However, the procedures used in these models will be elaborated on further in the following section.

Effective propagation algorithms alone are not enough to ensure accurate predictions of aviation noise. Propagation must begin with an input that correctly characterizes the source. Therefore, a short background of the directional nature

of aviation noise sources and the impact of source directivity on propagation is also included.

1.2.1 Geometric Spreading

As sound travels outward from a source like a point or line, it spreads in space. In so spreading, the sound is attenuated with distance at a rate determined by the source geometry. Point sources and sources small in extent compared to the wavelength of sound experience a 6 dB decrease in sound level per doubling of distance from three-dimensional spherical spreading. Infinite line sources experience a decrease of 3 dB per distance doubling from two-dimensional cylindrical spreading [14, 15]. Geometric spreading is one of the most elementary and well-understood propagation effects. Inclusion of this effect in numerical propagation models is standard.

1.2.2 Atmospheric Absorption

Atmospheric absorption is another fundamental mechanism of attenuation. It includes, collectively, the effects of thermal losses, shear viscosity losses, and molecular relaxation losses of oxygen and nitrogen, experienced as sound travels through the air. The amount of atmospheric absorption is frequency dependent and, acting as a low pass filter, the effect significantly attenuates high frequency sound while allowing low frequencies to propagate further. It is also a function of humidity and temperature [14, 16].

Different standards exist for calculating atmospheric absorption in numerical propagation models. It is commonly quantified as a decibel attenuation over a unit of distance, called the absorption coefficient. INM adheres to the calculation methods of the Society of Automotive Engineers (SAE) Aerospace Recommended Practice (ARP) 866A [17] and SAE Aerospace Information Report (AIR) 1845 [9]. The International Standard Organisation (ISO) 9613-1 [18] is also commonly used.

1.2.3 Ground Effects

Properties of the ground surface under a propagation path can also significantly affect sound. For example, propagation over freshly fallen snow will lead to substantially lower noise levels than concrete-covered ground. The effect of the ground also has an important influence on the interference patterns produced by propagated sound of a single frequency. Sound traveling directly from the source interacts with sound reflected off the ground surface to create patterns of peaks and dips where sound adds constructively and destructively, respectively. The characteristics as well as locations of these peaks and dips are affected by the properties of the ground surface. These properties are quantified in the complex impedance of the ground surface, defined as the ratio of the complex pressure amplitude to the normal component of complex velocity amplitude, evaluated at the surface of the ground [1]. In its purest form, the impedance of the ground surface is calculated as a function both of properties of the ground and of the incident sound, which can become prohibitively complicated. However, applying various approximations can simplify the equations into implementable calculations for numerical propagation models.

Certain propagation cases include more than one type of ground surface along the path between source and receiver. For example a source may be over water with the receiver in a grassy field. In such cases, both types of ground affect propagation and can be incorporated by including a ground impedance discontinuity in propagation calculations.

1.2.3.1 Ground Impedance

Two definitions of ground impedance are first distinguished to clarify the discussion of ground effects. The characteristic impedance of a ground material is defined as the ratio of the complex pressure amplitude to the complex velocity amplitude for a sound wave traveling inside a medium [1]. On the other hand, the impedance of a ground surface, as described above, is the ratio of the complex pressure amplitude to the normal component of the complex velocity amplitude evaluated at the ground surface. However, if the sound speed in the ground is much smaller than that in the air, the wave transmitted through the ground can be assumed

to travel normal to the ground surface, regardless of the incident angle. Such an approximated ground is called locally reacting. For a locally reacting ground, the characteristic impedance of the ground material, a complex quantity that is not a function of incident angle, can be used to approximate the impedance of the ground surface [1, 15].

Delany and Bazley [19] developed an empirical method to determine the characteristic impedance of a range of fibrous absorbent materials as a function of frequency, using a single parameter model based on flow resistance. The ground impedance normalized by the acoustic impedance of the air is calculated as [19]:

$$Z = 1 + 9.08 \left(\frac{\sigma}{f} \right)^{0.75} + i11.9 \left(\frac{\sigma}{f} \right)^{0.73} \quad (1.1)$$

where f is frequency and σ is the effective flow resistivity of the ground surface in cgs rays. (The sign of the imaginary part of the impedance has been changed from the Delany and Bazley paper to accommodate a $e^{-i\omega t}$ time dependence [8]). This work was extended by Chessell [20] and Embleton, et al.[8] to substantiate the model for a larger variety of ground and propagation conditions. Note that effective flow resistivity is a term for values deduced from measured sound pressure levels using the single parameter model, and may depend not only on the flow resistivity, but additional parameters of the ground surface, as described in [8]. The effective flow resistivities for various types of ground surface are shown in Table 1.1.

A more precise four-parameter method of calculating impedance was later developed by Attenborough [21], requiring inputs of flow resistivity, grain shape factor, pore shape factor ratio, and tortuosity [1]. While a different impedance calculation method could be substituted into the hybrid model discussed in this dissertation, the remainder of this discussion will rely on the simpler single parameter method of ground impedance calculation, assuming a locally reacting ground.

The ground impedance is important to propagation because it can affect the overall levels of sound reaching a community and the interference patterns produced by each individual frequency of sound. An acoustically hard surface, like water or concrete, can cause an approximate pressure doubling of sound above the ground. In such a case, the interference pattern will have pronounced minima and maxima in regions of destructive and constructive interference. Over porous, ab-

Table 1.1. Effective flow resistivities as calculated by comparing predicted and measured spectra, from [3], originally appeared in [8]. Reproduced with permission.

Description of surface	Effective flow resistivity kPa s/m ²
0.1-m new fallen snow, over older snow	10–30
sugar snow	25–50
floor of evergreen forest	20–80
airport grass or old pasture	150–300
roadside dirt, ill-defined, small rocks up to 0.01-m mesh	300–800
sandy silt, hard packed by vehicles	800–2500
thick layer of clean limestone chips, 0.01- to 0.025-m mesh	1500–4000
old dirt roadway, small stones with interstices filled by dust	2000–4000
earth, exposed and rain-packed	4000–8000
very fine quarry dust, hard packed by vehicles	5000–20 000
asphalt, sealed by dust and use	~30 000
upper limit, set by thermal conduction and viscosity	2×10^5 to 1×10^6

sorbing ground surfaces like grass or freshly-fallen snow, phase changes stem from the complex ground impedance and cause the interference pattern minima to be less pronounced and shifted to lower frequencies [1, 14].

Knowledge of the ground surface impedance is sufficient for use in most of the calculations of the PE. However, analytical model calculations, as well as the FFP model and PE starting field input calculations require use of a reflection coefficient parameter. The reflection coefficient is a function of the impedance of the ground surface and is defined as the ratio of reflected to incident pressure. Plane-wave and spherical-wave reflection coefficient equations are presented in this section. However, detailed derivation of these equations is omitted for brevity. They can be found in reference [1].

In plane wave propagation, the reflection coefficient is calculated as [1]:

$$R_p = \frac{Z \cos \theta - 1}{Z \cos \theta + 1} \quad (1.2)$$

where Z is the normalized impedance of the ground surface and θ is the incident or reflected angle as shown in Figure 1.1

To represent sound emitted from a point source, which spreads to all angles, a single incident angle input for Equation 1.2 cannot be assigned and a different re-

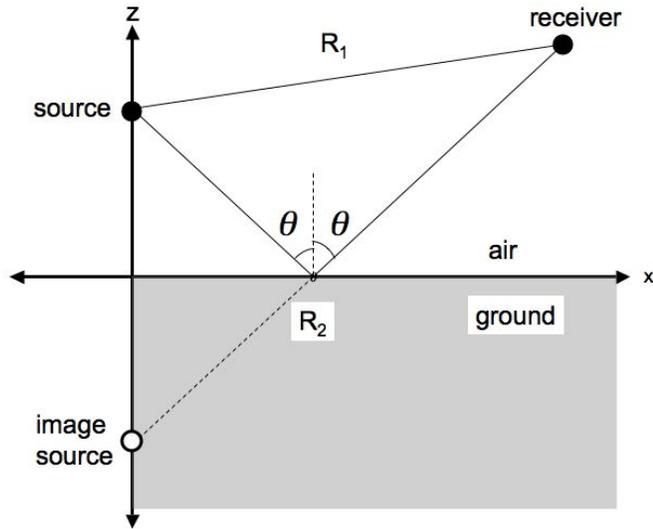


Figure 1.1. Geometry of source and receiver geometry above a ground surface, after [1] Figure D.1.

flection coefficient must be used. The spherical-wave reflection coefficient captures the more complicated reflections of sound from a point source. The physical representation of the image source may be imagined as a source distributed in space, with the largest values at the standard image source location, extending out to infinity, both horizontally and in depth [3]. It is calculated as [1]:

$$Q = 1 - 2 \frac{k_1}{Z} \frac{R_2}{\exp(ik_1 R_2)} \int_0^\infty \exp\left(-\frac{qk_1}{Z}\right) \frac{\exp(ik_1 \sqrt{r^2 + (z + z_s + iq)^2})}{\sqrt{r^2 + (z + z_s + iq)^2}} dq \quad (1.3)$$

where k_1 is the wave number in the air above the ground, Z is the normalized impedance of the ground surface, R_2 is the distance between the image source and receiver, and z_s is the source height.

The spherical-wave reflection coefficient can be used to calculate the analytical complex pressure amplitude of a point source above a finite impedance ground surface as [1]:

$$p_c = S \frac{\exp(ik_1 R_1)}{R_1} + QS \frac{\exp(ik_1 R_2)}{R_2} \quad (1.4)$$

where S is a constant, and R_1 is the distance between the source and receiver. Use

of this spherical-wave reflection coefficient describes the behavior not only of the reflected sound as considered for a plane wave, but also of the ground and surface waves that form from the spherical nature of the reflected sound waves [15].

This solution can be further simplified to the Weyl-van der Pol form of solution, used in electromagnetic propagation theory, which takes the form [14]:

$$p_c = \frac{S \exp(ik_1 R_1)}{R_1} + [R_p + (1 - R_p)F(w)] \frac{S \exp(ik_1 R_2)}{R_2} \quad (1.5)$$

where $F(w)$ is the boundary loss factor that accounts for the non-planar wavefronts and is calculated through use of the complementary error function. However, this form of the spherical-wave reflection coefficient is not used in the hybrid model presented in this dissertation and further explanation is, therefore, beyond the scope of this research.

1.2.3.2 Ground Impedance Discontinuities

Transitions between different types of ground further complicate the calculations of sound propagation. For example, ground transitions may occur near airports where concrete runways meet surrounding grassy fields. Airports near oceans may face transitions between grass and water. In such cases, a diffracted field is produced by the impedance discontinuity at the interface of the two ground types [22, 23].

The effects of ground impedance discontinuities have been measured and found significant in experiments involving different types of ground surfaces [22, 23]. A number of different models exist for calculating the effect of ground impedance discontinuities: the semi-empirical solution models of de Jong [24] and Koers [25], the numerical solution of Rasmussen [26], Fresnel zone methods, and the Boundary Element Method [27]. The parabolic equation method includes impedance transitions with a simple change in the value of the ground impedance input between range steps. Limitations of the model prevent inclusion of the diffracted sound at the transition [28].

1.2.4 Meteorological Effects

Meteorological effects are made up of two main mechanisms: atmospheric refraction and atmospheric turbulence. A refractive atmosphere, caused by gradients

of temperature and/or wind speeds, can have a large effect on the sound field, especially at longer ranges, and when shadow zones are formed.

Atmospheric turbulence manifests as small time scale fluctuations in the temperature and wind velocity around some mean value. It has the effect of filling in interference pattern dips and increasing sound levels in shadow zones caused by refraction or terrain obstructions [1].

1.2.4.1 Atmospheric Refraction

During the daytime, solar radiation warms the ground more quickly than the air. As the ground warms, heat is conducted into the air above, raising the temperature from the ground up. In typical daytime conditions, the temperature of the air can decrease with increasing height by about 5°C in the meter above the ground and about 3 to 5°C more up to 100 m above the ground [3].

The speed of sound in air is proportional to the square-root of temperature. Therefore, a negative gradient in sound speed results from the daytime's decrease in air temperature with height. Sound in a negative sound speed gradient refracts upward, as shown on the left in Figure 1.2.

Upward refraction also occurs in upwind conditions, where viscous drag effects at the ground cause the wind speed to increase with height from the ground [3]. Because the effective speed of sound is decreased opposite to the direction of wind, a negative sound speed gradient results and causes upward refraction, seen on the right side of the source in Figure 1.3.

During the nighttime, the ground cools faster than the atmosphere and the temperature of the air increases with height above ground [3]. An increasing sound speed follows from the increasing temperature, and corresponds to a positive sound speed gradient. The positive sound speed gradient causes sound to refract downward, as shown on the right in Figure 1.2. In such cases, additional sound paths involving multiple ground reflections are possible, as shown in Figure 1.4.

Downwind conditions have an opposite effect on the change in the effective sound speed gradient as upwind. In the direction of the wind, the gradient is positive, and downward refraction results, seen on the left side of the source in Figure 1.3.

The effects of temperature and wind can be captured in an effective sound

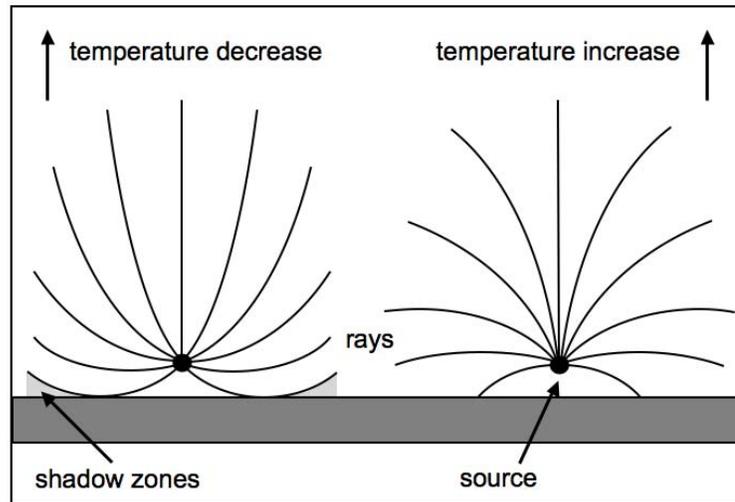


Figure 1.2. Refraction caused by thermal gradients, after [2]

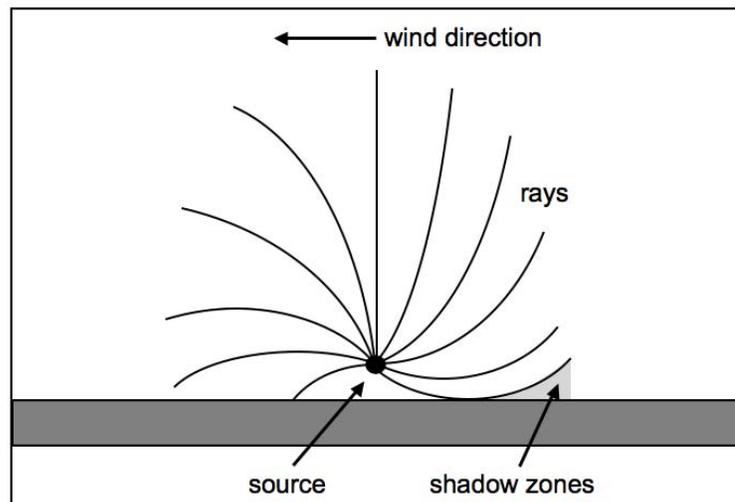


Figure 1.3. Refraction caused by wind gradients, after [2]

speed profile that defines the speed of sound at each position in height. Different functions are used to model the profile, the most popular of which are linear functions, logarithmic functions, or a combination of both. A linear sound speed profile takes the form of [1]:

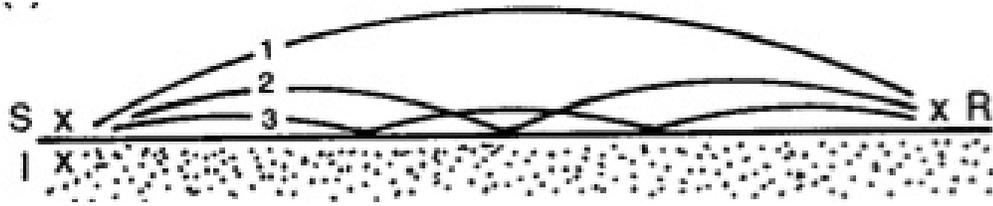


Figure 1.4. Multiple ray paths between the source and receiver from [3]. Reproduced with permission.

$$c(z) = c_0 + az \quad (1.6)$$

where c_0 is the sound speed at the ground, a is the sound speed gradient, and z is height. A logarithmic sound speed profile takes the form of [1]:

$$c(z) = c_0 + b \ln \left(\frac{z}{z_0} + 1 \right) \quad (1.7)$$

where b is the parameter of the logarithmic sound speed profile and z_0 is the aerodynamic roughness length of the ground surface (usually between 0.01 m and 0.1 m for an open field of grassland) [1].

The linear and logarithmic profiles can represent either the adiabatic (thermal) sound speed $c(z)$ or the horizontal wind velocity in the direction of propagation $u(z)$. While the adiabatic sound speed profile is the same in all directions horizontally from the source, the horizontal wind velocity profile will change depending on the angle between propagation and the direction of the wind. The effective sound speed profile is the sum of the adiabatic and wind speeds as [1]:

$$c_{eff}(z) = c(z) + u(z) \quad (1.8)$$

This effective sound speed approximation is valid for conditions in which sound travels at relatively small elevation angles [1].

1.2.4.2 Atmospheric Turbulence

Atmospheric turbulence is characterized by the fluctuations in the sound speed around some average value due to variations of wind velocity and temperature. The acoustic refractive index $n = c_0/c_{eff}$ is a different form of effective sound

speed representation, and can be divided into two components: the average value and the fluctuation [1]:

$$n = \bar{n} + \mu \quad (1.9)$$

where \bar{n} is the average value of the acoustic refractive index, which may be a function of position, and μ is the fluctuation where $\mu \ll \bar{n}$ and the average of the fluctuation $\bar{\mu} = 0$ [1]. Turbulence increases the sound levels within interference pattern minima and shadow regions caused by upward refracting atmospheres and raised terrain [1]. Effects of turbulence increase with distance [3].

Turbulence is manifested as eddies ranging in size from the largest, energy containing eddies of the order of the height of the atmospheric boundary layer, to the smallest, dissipative eddies, of the order of 1 mm. Energy cascades down from the larger eddies to the smaller eddies, eventually being dissipated by viscous and thermal losses. The spectral density of the random fluctuations (both thermal and velocity) reveals the distribution of the different sizes of eddies in the turbulent field. The shape of the spectral density is influenced by the structure of the atmospheric boundary layer and the cascade process [3].

In turbulence models, variations in the atmospheric turbulence are assumed to be much slower than the speed of sound. The atmospheric turbulence is, therefore, assumed to be “frozen” as compared to the speed of the sound waves moving through. The sound field is calculated for each frozen “snapshot” of a random turbulence realization, defined by the refractive-index fluctuation μ . An ensemble of these realizations is amassed using a numerical propagation model and averaged over to determine the averaged effects of the turbulence [1].

1.2.5 Terrain Effects

Terrain features can have substantial effects on the paths of propagated sound. Interestingly, the effects of features like hills and valleys are closely related to effects caused by atmospheric refraction. For example, a shadow zone formed behind a hill, as illustrated in Figure 1.5, behaves similarly to one formed in an upward refracting atmosphere. Parallels between uneven terrain and refractive atmospheres are further depicted in Figure 1.6.

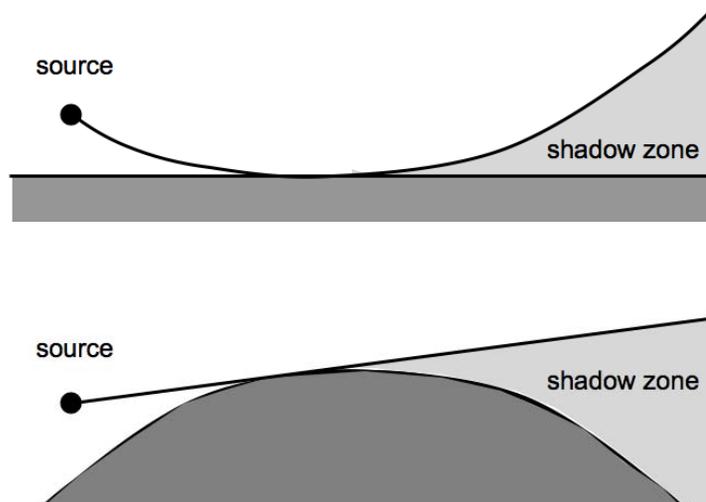


Figure 1.5. A shadow zone is formed behind a hill, after [1] Figure 5.1.

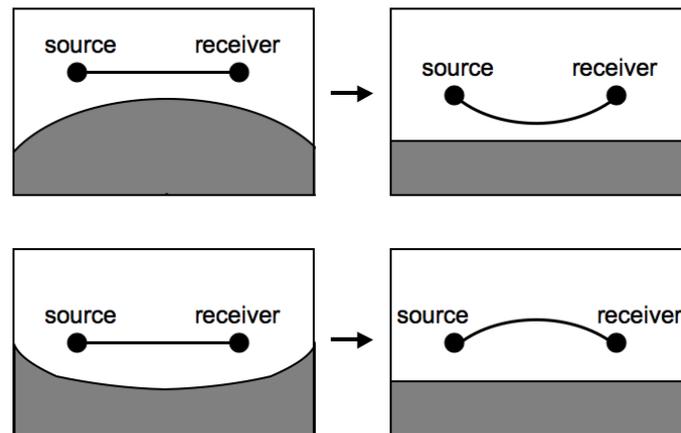


Figure 1.6. Parallels between terrain features and atmospheric refraction: A hill corresponds to an upward refracting atmosphere and a valley corresponds to a downward refracting atmosphere, after [1] Figure 6.3.

The complexities of terrain are handled with varying levels of sophistication in numerical models. INM uses a simplified method of calculation based on the difference in path length between the direct, unobstructed path from source to receiver, and the path over the terrain (Section A.3.1.3) [4]. The Geometrical Theory of Diffraction is also commonly applied, as in the ray-based propagation models,

AAM and NMSim [29, 13]. The PE method employs a coordinate system transformation to propagate sound over a flattened ground using altered propagation algorithms.

1.2.6 Source Characteristics

Sound is not radiated uniformly in all directions from aviation noise sources. Different sources—jet, propeller, airframe, etc.—have different characteristic directivity patterns. The individual aviation noise sources can be modeled by elementary monopole, dipole, and quadrupole sources. For example, a monopole can model a pulsating jet as in a piston-engine exhaust, a dipole can model propeller noise, and a quadrupole has been shown by Lighthill [30] to model aerodynamic noise from a subsonic turbulent jet [31].

A solution for a monopole source above an impedance ground surface was explored in Section 1.2.3.1. Closed-form analytical solutions for dipole and quadrupole sources, arbitrarily oriented above a ground surface, have also been determined [32, 33].

The sound field for an arbitrarily oriented dipole source above an impedance plane can be represented as a superposition of horizontally- and vertically-oriented dipoles and can be written in a similar form as a monopole source, including terms for the direct sound field, the image sound field, and the ground wave component [32]. It was found that the difference between sound fields of a monopole and a horizontal dipole is small at near-grazing angles over an impedance surface. At larger angles, the difference is more significant.

Because a vertical dipole radiates more energy downward, toward the ground, the ground wave component represents a larger portion of the total sound field. Consequently, the vertical dipole field shows large differences from the monopole field at near-grazing angles. In addition, the straightforward application of the reciprocity theorem, in which identical results can be obtained by interchanging source and receiver positions, was determined to be invalid for the vertical dipole because of angular dependencies [32].

The sound field for an arbitrarily oriented quadrupole source above an impedance plane cannot be manipulated to a similar form as a monopole, though

a closed form solution can still be written. A lateral quadrupole over a ground surface tends to behave similarly to a vertically oriented dipole at further distances from the source. Both the vertical quadrupole and vertical dipole show larger excess attenuation, as compared to the same source in free field, than a monopole. A vertical longitudinal quadrupole shows lower excess attenuation than a vertical dipole. Both, again, are greater than the monopole [33].

The differences between the sound fields for different fundamental sources above a ground indicate the importance of including source directivity effects in sound propagation models. Sources with arbitrary directivity patterns have not previously been explored in the literature for the parabolic equation method of atmospheric propagation. Therefore a technique for representing directional sources was developed in this research.

1.3 Standard and State of the Art Aviation Noise Models

Standard models in widespread use are inclined to meet computational requirements of a typical PC, achieve fast runtimes, and minimize their learning curves to accommodate a diverse group of users. Often, to accomplish this, approximations that separate the model from the physics of propagation are made. However, as computational power increases, use of physics-based propagation models becomes more realistic.

This section briefly introduces some of the major aviation noise propagation models and state of the art modeling techniques. It begins with a description of the popular segmentation model INM, then discusses the ray-based model AAM. It concludes with short descriptions of the state of the art methods of noise propagation including the ray model, Green's Function Parabolic Equation model, Crank-Nicholson Parabolic Equation model, and Fast Field Program. The Crank-Nicholson Parabolic Equation and Fast Field Program methods, which are featured in this research, are discussed further in Chapters 2 and 3.

1.3.1 Existing Aviation Noise Models

INM, the FAA’s standard model of noise assessment, and NOISEMAP, the required method for modeling noise exposure near military air bases, are segmentation models. These models use source information compiled in large database structures call Noise-Power-Distance (NPD) tables. NPD tables contain data separated by aircraft type and flight operation, for a given engine power and distance from the flight path. NPD tables represent sound level information assuming an aircraft proceeding along straight flight paths, of infinite length, parallel to the ground, under reference conditions. Corrections are applied to these curves to represent realistic finite segments of more complex flight paths and to include complicated propagation effects [4].

Many of the adjustments used in segmentation models are based on simplified approximations. For example, in INM:

- One azimuthal directivity pattern of aircraft noise is assumed for all aircraft configurations and speeds [4], though these variables can affect aerodynamic noise source contributions to the directivity pattern.
- The ground effect adjustments, developed from field measurements over acoustically soft ground, are applied for all airports. Therefore, sound levels calculated around airports that border hard surfaces, such as water or concrete, may be under-predicted [4, 34].
- Uneven terrain is addressed by invoking the optional line-of-sight blockage adjustment, but no distinction is made between different terrain shapes (Section A.3.1.3).
- The meteorological effects of refraction and scattering are incorporated assuming a generalized atmosphere. No provisions are made for atmospheric conditions specific to the atmosphere’s sound speed profile or turbulence [4].

Ray-trace models are more sophisticated than segmentation models and can allow for more detailed characterization of noise sources and more precise inclusion of propagation effects. The Advanced Acoustic Model (AAM), a model developed to replace NOISEMAP for modeling noise exposure near military air bases, is ray-based. For example, in AAM:

- Detailed sound spheres are used to represent sound emanating from a single point, with a particular directivity, for different frequency representations.
- Ground effects are based on the methods that use a more descriptive ground impedance parameter for a more accurate ground effect representation.
- Terrain features are categorized by their shapes as flat, uphill, downhill, valley, as well as wedges and screens with varied levels of complexity.
- A curved ray model is used to incorporate meteorological effects, using a horizontally stratified atmosphere. However, its use is limited to single event analysis occurring over uniform, flat terrain [13].

While the ray method can account, more accurately, for a larger number of noise propagation effects, it is a high frequency approximation and does not always represent low frequency acoustic behavior properly. In addition, though sound spheres are more detailed characterizations of the source, the current reach of the available sound sphere database is small, limited by the time and expense required for data collection. Consequently, because an extensive NPD database already exists for a large range of aircraft and engine types, the NPD representation is more commonly used.

A more comprehensive overview of currently used transportation noise models and the algorithms they use can be found in Appendix A.

1.3.2 State of the Art in Propagation Modeling

Several state of the art methods of sound propagation have been developed and used in the literature for different conditions of propagation. This section will introduce the basic concepts of the ray model, Green's Function Parabolic Equation model, Crank-Nicholson Parabolic Equation model, and Fast Field Program. The Boundary-Element Method (BEM) is also commonly used for sound propagation near barriers, such as vertical screens. However, noise barriers are not considered in this research and BEM is not explored further.

1.3.2.1 Ray Model

The ray model propagates sound with an approach called geometrical acoustics. In the model, the full sound field is obtained by adding the contributions of rays propagated outward from a source. The ray model involves two components of calculations: determining the ray paths and finding the contribution of each ray to overall sound pressure [1]. The eikonal equation describes the paths sound rays take, and the transport equation calculates their contributions to the sound field. A series expansion, in the inverse of angular frequency, of the pressure is used to arrive at the transport equation. Truncation of the series precludes accurate propagation in the low frequency limit [35]. The ray model produces inaccurate results at focusing areas called caustics, where the model over-predicts sound levels, and in shadow zones, where it under-predicts sound levels [36]. However, as implemented in AAM, it can accommodate range-dependent effects.

1.3.2.2 Green's Function Parabolic Equation Model

The Green's Function Parabolic Equation (GFPE) [37], along with the Crank-Nicholson Parabolic Equation (CNPE) are two main formulations of the PE method. Both PE methods utilize a vector starting field, defined for all heights at the first range step, and extrapolate results in range. However, the method of solving the parabolic equation differs. The GFPE uses Fourier transforms combined with a transformed Green's function of a point source over a finite impedance ground to propagate sound. It can use far larger step sizes in range than can the Crank-Nicholson Parabolic Equation and, therefore, has faster computational speed. However, it includes the effect of a refractive atmosphere as a phase factor, multiplied into the solution with each step in range. With this approximation in including the effects of a refractive atmosphere, errors are introduced if the change in sound speed with height and the range step size are large. However, if a high-order starting function is used, the GFPE maintains accuracy to higher elevation angles than does the CNPE. It does not allow a high degree of flexibility in modeling boundaries, unlike finite-difference formulations of the PE, such as the CNPE, which can [1, 38, 37].

1.3.2.3 Crank-Nicholson Parabolic Equation Model

The Crank-Nicholson Parabolic Equation (CNPE) model begins with the same form of parabolic equation as does the GFPE, but implements a different method of solution. In the CNPE, an alternate expansion is utilized. The truncation of the CNPE expansion enforces an elevation angle limitation, for which accuracy is degraded at larger angles from the source as measured from the primary direction of propagation. The order of the expansion determines to what angle the CNPE results are considered accurate. In contrast to the GFPE, the CNPE method requires much smaller steps in range, increasing computation time. However, with a variation on the CNPE allowing it to incorporate a generalized terrain function, it can model more complicated boundaries. Neither the GFPE nor the CNPE are restricted to high frequencies and are, therefore, accurate for low frequencies. Both formulations can include range-dependent effects [1, 38].

Because of its accuracy at low frequencies, range-dependent effect capabilities, and increased flexibility and ease in modeling boundaries, the finite difference, CNPE formulation is chosen as the basis of this research and is described in detail in Chapter 2.

1.3.2.4 Fast Field Program

The Fast Field Program (FFP) is based on an approximate form of the wave equation that employs a transform from the horizontal spatial domain to the horizontal wave number domain. The equation is solved numerically in the horizontal wave number domain for a point source above a finite impedance ground, and transformed back into the spatial domain to return sound field results. Because it works only in horizontally stratified layers of the atmosphere, the FFP is limited to range-independent effects and, therefore, cannot incorporate terrain features or changes in the ground or atmosphere with range. It can incorporate a sound speed profile that varies in height and does not make assumptions that degrade its accuracy at low frequencies or large elevation angles [1].

The FFP has strengths that compliment the limitations of the finite difference formulation of the PE. It is, therefore, used in this research to supplement the PE and is described in detail in Chapter 3.

1.4 Research Objectives and Outline

The goal of this research is to accurately predict aviation noise in complex propagation environments, with a focus on low frequency sound, an issue that is particularly relevant to aviation noise. To realize this goal, focus was given to both the propagation of sound and to the correct representation of the noise source. The parabolic equation (PE) method, on which this research is based, is capable of incorporating range-dependent propagation effects including ground impedance, terrain, and meteorology. In addition, it can accurately capture low frequency propagation behaviors. In order to faithfully represent the directional characteristics of the sources of aeroacoustic noise, a new method was formulated to accommodate directional sources within the PE.

The PE model is inherently limited by degraded accuracy at large angles from the horizontal of the noise source. Because of this elevation angle limitation, a hybrid model was devised. The hybrid model, designed in this research and described in detail in this dissertation, uses the PE model in conjunction with an additional propagation method, the Fast Field Program (FFP). The FFP, accurate both at low frequencies and at large elevation angles, supplements the PE. It is, however, limited to inclusion of non-range-dependent propagation effects.

This dissertation discusses the design of the hybrid PE-FFP model including detailed descriptions of the PE and FFP methods, the formulation and capabilities of the full hybrid model, and the contouring capabilities developed to compliment those in current predictive aviation noise models. This chapter presented an overview of the effects that contribute to the challenge of aviation noise propagation in realistic conditions. Chapters 2 and 3 introduce the two-dimensional PE and FFP propagation methods, respectively, detailing the formulation of the models used in the research, and describing the models' capabilities and limitations. Chapter 4 describes the structure of the two-dimensional hybrid PE-FFP model, including input requirements and test case validation. Chapter 5 outlines the process of transforming the model from a two-dimensional space to a pseudo-three-dimensional representation, and finally to noise contour maps. Chapter 6 describes the method of including source directivity patterns in the PE. Finally, Chapter 7 provides a summary of this dissertation, conclusions of the research,

and ideas for future work.

The Two-Dimensional Parabolic Equation Model

2.1 Introduction

The parabolic equation (PE) method [39] is a common approach to modeling propagation when ground impedance [28, 40], meteorology [41, 42, 43], and terrain effects [44] are significant and vary with range. Because the parabolic equation method makes no high frequency assumptions, it can accurately capture low frequency propagation behaviors [45] that contribute to community impact.

The parabolic equation method is based on an approximate form of the wave equation with a preferred direction of propagation. It has been used in a wide variety of fields including tropospheric propagation of electromagnetic waves, most commonly radio waves, where it was first introduced by Leontovich and Fock [46, 47]. In the 1970s, parabolic equation method was applied to underwater acoustic propagation by Tappert [48] and to geophysics applications by Claerbout [49]. It was extended to outdoor sound propagation by Gilbert and White [41, 50] and has since been used extensively in the field of atmospheric acoustic propagation. More complete historical reviews of the parabolic equation method development can be found in [48] and [51].

The PE method calculates a two-dimensional sound field by assuming axial symmetry around a vertical axis through the source and extrapolating from a

starting field vector input that represents the source [1]. The starting field defines the sound at all heights in the grid for the first step in range. To extrapolate from the starting field, one-way propagation and small angle propagation approximations are applied to the Helmholtz equation.

There are various formulations of the PE method. This research uses the Generalized Terrain Parabolic Equation (GTPE) method [44], that employs a wide-angle Crank-Nicholson marching scheme. It can include the propagation effects of a refractive atmosphere, finite ground impedance, uneven terrain, and turbulence. This chapter outlines both the standard and Generalized Terrain PE model derivations, following [1] closely, describes the generalized terrain model implementation, and offers numerical examples to demonstrate the model's capabilities.

2.2 Derivation of the Parabolic Equation

The Helmholtz equation is the time-independent form of the wave equation, assuming a time harmonic acoustic pressure p . In its general form it is

$$\nabla^2 p + k^2 p = 0 \quad (2.1)$$

where ∇^2 is the Laplacian, p the complex sound pressure amplitude, and $k = \frac{\omega}{c}$ the effective wave number where ω is the angular frequency of sound and c the effective sound speed.

The PE derivation begins with the two-dimensional, cylindrical form of the Helmholtz equation, neglecting variations in the azimuthal direction:

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r} \frac{\partial p}{\partial r} + \frac{\partial^2 p}{\partial z^2} + k^2 p = 0 \quad (2.2)$$

where r is the range variable and z the height variable. The quantity $q = p\sqrt{r}$, a form of the acoustic pressure that excludes attenuation from cylindrical spreading, is then substituted into the Helmholtz equation:

$$\frac{\partial^2 q}{\partial r^2} + \frac{1}{4r^2} q + \frac{\partial^2 q}{\partial z^2} + k^2 q = 0 \quad (2.3)$$

where the second term, $q/4r^2$, emerges from applying the product rule in expanding

the derivatives of $\partial^2 p / \partial r^2 = \partial^2(qr^{-1/2}) / \partial r^2$ and $\partial p / \partial r = \partial(qr^{-1/2}) / \partial r$.

Applying the far-field assumption $r \gg \frac{1}{k}$, the second term on the left is neglected, leaving:

$$\frac{\partial^2 q}{\partial r^2} + \frac{\partial^2 q}{\partial z^2} + k^2 q = 0. \quad (2.4)$$

This equation, derived from the general form of the Helmholtz equation by assuming axisymmetry and far field evaluation, is the starting point for both the standard and generalized terrain versions of the PE method.

2.2.1 Crank-Nicholson Parabolic Equation Model

In the standard PE method, results at each vertical grid point at a given step in range are calculated from the results at all heights at the previous range step, as illustrated in Figure 2.1. In order to accomplish such an extrapolation, an estimate of the horizontal wave number must be reached. This is the motivation for the manipulation of Equation 2.4 that follows.

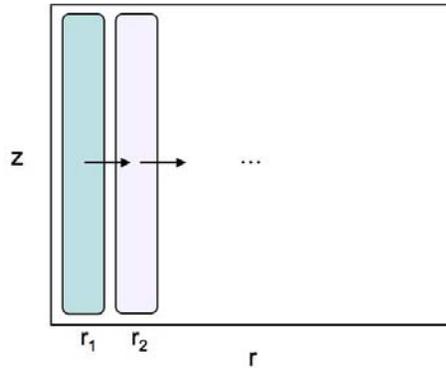


Figure 2.1. Diagram of PE propagation method: begin with the initial field and march forward in range, extrapolating from the previous range step.

In the standard PE, Equation 2.4 can be rewritten as:

$$\frac{\partial^2 q}{\partial r^2} + H(z)q = 0, \quad (2.5)$$

where $H(z) = \frac{\partial^2}{\partial z^2} + k^2(z)$. Here, the wave number $k(z)$ is assumed to be constant across each range step, but can vary with height. Equation 2.5 is factored into

forward- and backward-propagating components:

$$\left[\frac{\partial}{\partial r} - i\sqrt{H} \right] \left[\frac{\partial}{\partial r} + i\sqrt{H} \right] q = 0. \quad (2.6)$$

Assuming $e^{-i\omega t}$ time dependence, the first bracketed term represents forward-propagating sound and the second, backward. Neglecting backward propagating sound, Equation 2.6 reduces to:

$$\frac{\partial q}{\partial r} - i\sqrt{H}q = 0. \quad (2.7)$$

Anticipating a small elevation angle assumption, where sound is assumed to propagate primarily in the horizontal direction r , a solution to Equation 2.7 can be written in the form

$$q(r, z) = \psi(r, z)e^{ik_a r}. \quad (2.8)$$

where k_a is a reference wave number, for example the wave number at the ground or an average height value. This is a convenient representation that separates the component $e^{ik_a r}$, which varies rapidly in r for nearly horizontal propagation, and $\psi(r, z)$, which varies slowly in r . An illustration for a monopole source is shown in Figure 2.2, where the concentric circles mark the wave fronts. For sound emitted horizontally from the source, the wave number is composed only of a horizontal component. Thus, the wave fronts look, locally, like plane waves with rapid oscillation in the r -direction, and no variation in the z -direction. At slight angles from the horizontal, the oscillations are still rapid in the r -direction, but now there is also slow variation in the z -direction. At 45° , the rate of oscillation is equal in the r - and z -directions, and Equation 2.8 no longer an advantageous form of the solution.

To further simplify the equation for numerical implementation, the operator $\sqrt{H(z)}$, is approximated. The elevation angle range for which the PE formulation is valid is determined by the order of the approximation for $\sqrt{H(z)}$. The narrow-angle formulation uses two terms of a Taylor series expansion:

$$\sqrt{H(z)} = k_a \sqrt{1 + s} \approx k_a \left(1 + \frac{1}{2}s \right) \quad (2.9)$$

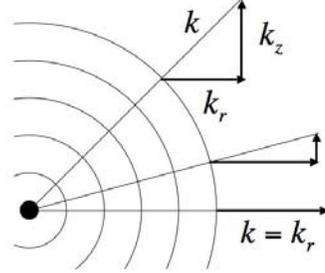


Figure 2.2. Wave number break-down into horizontal and vertical components at different angles from a monopole source.

where $s = \frac{k^2(z) - k_a^2}{k_a^2} + \frac{1}{k_a^2} \frac{\partial^2}{\partial z^2}$. This approximation is accurate up to elevation angles of 10° [1].

The implications of this approximation can be better understood by considering a plane wave propagating in a homogeneous atmosphere with acoustic wave number k_a at an angle θ from the horizontal, as shown in Figure 2.3, represented by the equation

$$q(r, z) = e^{ik_a r \cos \theta + ik_a z \sin \theta}. \quad (2.10)$$

The $\sqrt{H(z)}$ operator would then become

$$\sqrt{H(z)} = k_a \sqrt{1 - \sin^2 \theta} \approx k_a \left(1 - \frac{1}{2} \sin^2 \theta\right). \quad (2.11)$$

Therefore, the narrow-angle approximation assumes the vertical wave number k_z is small compared to the acoustic wave number k_a , where $\frac{k_z}{k_a} = \sin \theta$ [38].

The wide-angle formulation, used in this research, employs a Padé expansion of the form [49]

$$\sqrt{H(z)} = k_a \sqrt{1 + s} \approx k_a \frac{1 + \frac{3}{4}s}{1 + \frac{1}{4}s}. \quad (2.12)$$

This approximation, with quadratic accuracy in s , is accurate up to elevation angles of $\pm 35^\circ$ [1]. In later discussions, the limitation of $\pm 35^\circ$ will be understood, but the \pm sign will be omitted.

Equation 2.12 is substituted into Equation 2.7 to get the wide-angle parabolic

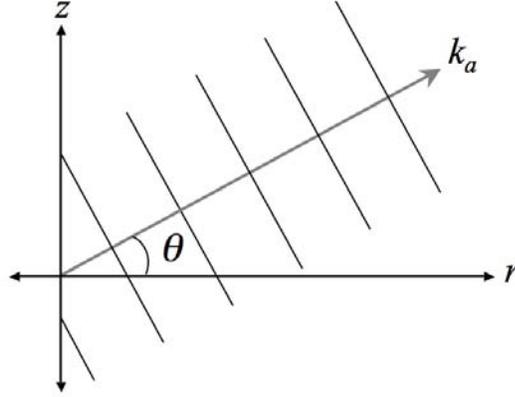


Figure 2.3. Diagram of plane wave propagating at a small angle θ from the horizontal.

equation

$$\left(1 + \frac{1}{4}s\right) \frac{\partial q}{\partial r} = ik_a \left(1 + \frac{3}{4}s\right) q \quad (2.13)$$

into which Equation 2.8 can be substituted to obtain the equivalent equation for the envelope function

$$\left(1 + \frac{1}{4}s\right) \frac{\partial \psi}{\partial r} = \frac{1}{2} ik_a s \psi. \quad (2.14)$$

To reach the final numerical, finite-difference form of the wide-angle parabolic equation, the central difference formula is applied to represent $\frac{\partial^2 \psi}{\partial z^2}$ within the variable s as

$$\left(\frac{\partial^2 \psi}{\partial z^2}\right)_{z_j} = \frac{\psi_{j+1} - 2\psi_j + \psi_{j-1}}{(\Delta z)^2} \quad (2.15)$$

where $z_j = j\Delta z$ with $j = 1, 2, \dots, M$, and $\psi_j = \psi(r, z_j)$, which corresponds to the vector $\vec{\psi}(r)$. In matrix form, $\frac{1}{2} ik_a s$ is

$$\frac{1}{2} ik_a s = \gamma \begin{pmatrix} \begin{bmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix} \end{pmatrix} +$$

$$\left(\begin{bmatrix} \beta_1 & & & & & \\ & \beta_2 & & & & \\ & & \beta_3 & & & \\ & & & \ddots & & \\ & & & & \beta_{M-1} & \\ & & & & & \beta_M \end{bmatrix} \right) \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \vdots \\ \psi_{M-1} \\ \psi_M \end{bmatrix} + \gamma \begin{bmatrix} \psi_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \psi_{M+1} \end{bmatrix} \quad (2.16)$$

where $\gamma = \frac{i}{2k_a}/(\Delta z)^2$ and $\beta_j = \frac{1}{2}i(k^2(z_j) - k_a^2)/k_a$. The final term of the equation adds in the contributions of ψ_0 and ψ_{M+1} that are lost from failure to include the preceding and following 1 in the first and last central difference matrix rows, respectively. For example, the equation derived from multiplication with the second row of the central difference matrix yields a term $\gamma(\psi_1 - 2\psi_2 + \psi_3)$ whereas the equation derived from multiplication with the first row yields only $\gamma(-2\psi_1 + \psi_2)$; therefore, the missing $\gamma\psi_0$ must be added back in separately.

Finite-difference equations can be used to approximate ψ_0 and ψ_{M+1} as

$$\psi_0 = \sigma_1\psi_1 + \sigma_2\psi_2 \quad (2.17)$$

$$\psi_{M+1} = \tau_1\psi_M + \tau_2\psi_{M-1} \quad (2.18)$$

where σ_1 and σ_2 are related to the ground impedance, as further discussed in Section 2.2.1.1.1, and τ_1 and τ_2 are related to the impedance at the top of the grid, as further discussed in Section 2.2.1.1.2. Thus, the missing $\gamma\psi_0$ can be added in by modifying the central difference matrix to the following tridiagonal matrix T

$$T = \begin{bmatrix} -2 + \sigma_1 & 1 + \sigma_2 & & & & \\ & 1 & -2 & 1 & & \\ & & 1 & -2 & 1 & \\ & & & \ddots & \ddots & \ddots \\ & & & & 1 & -2 & 1 \\ & & & & & 1 + \tau_2 & -2 + \tau_1 \end{bmatrix}. \quad (2.19)$$

where I is the identity matrix, a diagonal matrix with each value equal to unity. Thus, using matrix algebra for two tridiagonal matrices, the full sound field at one range step can be extrapolated from the sound field at the previous range step. In order to calculate the sound field in a rectangular grid, the starting field $\vec{\psi}(0) \equiv \psi(0, z) = q(0, z)$ is the required input.

2.2.1.1 Grid Boundary Conditions

This section discusses the treatment of the lower and upper boundaries of the vertical PE grid to simulate the influence of a ground surface, and reduce numerical artifacts created from a truncated “sky.”

2.2.1.1.1 Lower Grid Boundary The influence of the ground surface is included in the PE model through the parameters σ_1 and σ_2 , as introduced in Equation 2.17. Their calculation is based on the boundary condition at the ground

$$\begin{pmatrix} p \\ v_n \end{pmatrix}_{z=0} = Z\rho c \quad (2.27)$$

where p is the complex pressure amplitude, v_n is the complex velocity amplitude normal to the ground, Z is the specific acoustic impedance of the ground, for an assumed locally-reacting ground, and ρc is the impedance of the air just above the ground, pointing in the negative z direction [1].

The linearized Euler equation is used to relate the complex velocity amplitude in the normal direction to the complex pressure, assuming time harmonic sound waves, as

$$v_n = -\frac{1}{i\omega\rho} \frac{\partial p}{\partial z}. \quad (2.28)$$

Discretizing the derivative $\frac{\partial p}{\partial z}$ with the second-order finite-difference approximation

$$\frac{\partial p}{\partial z} = \frac{p_1 - p_0}{\Delta z} - \frac{1}{2}\Delta z \frac{p_2 - 2p_1 + p_0}{(\Delta z)^2} = \frac{-\frac{3}{2}p_0 + 2p_1 - \frac{1}{2}p_2}{\Delta z} \quad (2.29)$$

where p_0 , p_1 and p_2 are the pressures at the ground, and one, and two grid steps above the ground, respectively, and Δz is the vertical grid step spacing, a relation-

ship between p_0 , p_1 , and p_2 is obtained

$$p_0 = \left(3 - \frac{2ik_0\Delta z}{Z}\right)^{-1} (4p_1 - p_2) \quad (2.30)$$

where k_0 is the wave number of sound at the ground. This relationship also holds for the envelope function ψ , and can, therefore, be used to deduce σ_1 and σ_2 from Equation 2.17 by inspection as

$$\begin{aligned} \sigma_1 &= 4 \left(3 - \frac{2ik_0\Delta z}{Z}\right)^{-1} \\ \sigma_2 &= - \left(3 - \frac{2ik_0\Delta z}{Z}\right)^{-1}. \end{aligned} \quad (2.31)$$

2.2.1.1.2 Upper Grid Absorbing Layer In order to eliminate numerical artifact reflections from the upper grid boundary, an absorbing layer of approximately 50 wavelengths can be used at the top of the PE grid [1].

The equation for the top of the boundary, corresponding to Equation 2.30 for the lower boundary, is

$$p_{M+1} = (3 + 2ik_0\Delta z)^{-1} (4p_M - p_{M-1}) \quad (2.32)$$

where M is the point at the upper boundary of the grid, and Z has been replaced by 1, the specific acoustic impedance of air. This boundary condition is less important, however, due to an absorbing layer that is created to attenuate the sound before it reaches the upper boundary, to eliminate spurious reflections. The absorbing layer is achieved by introducing an imaginary term to the wave number within the layer. The imaginary term used in this research is determined as

$$iA_t(z - z_t)^2 / (z_M - z_t)^2 \quad (2.33)$$

where A_t is a coefficient that varies with frequency, z_t is the height where the absorbing layer begins, and z_M is the height of the upper grid boundary. The coefficient A_t is set to 1, 0.5, 0.4, and 0.2 at frequencies of 1000, 500, 125, and 30 Hz, respectively, is linearly interpolated between these frequencies, and is linearly extrapolated for frequencies below 30 Hz [1].

2.2.2 Generalized Terrain Parabolic Equation Model

To represent an uneven terrain profile with an arbitrary shape, the Generalized Terrain extension to the PE method is employed. This method begins with the same axisymmetric, reduced form of the Helmholtz equation, Equation 2.4, and then applies a coordinate transformation in which the height variable now measures the height above the ground surface, rather than an absolute reference height, as shown in Figure 2.4. The transformation is followed through the entire calculation method and reversed after the entire sound field has been calculated, returning to the standard coordinate representation. The derivation in this chapter follows [1], Appendix M.3, closely.

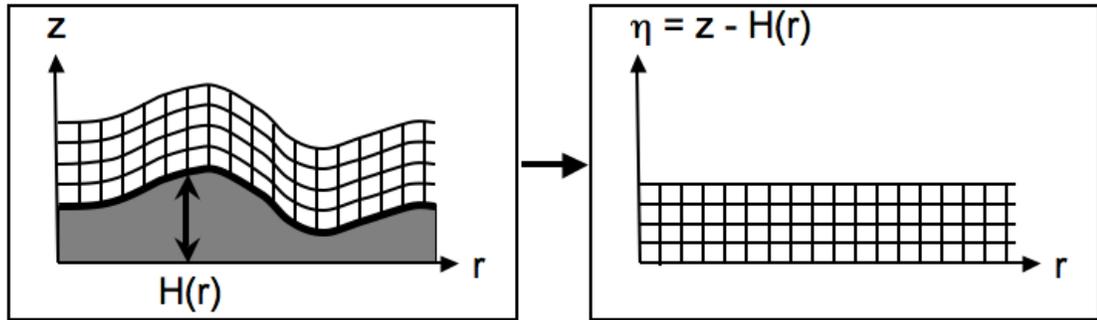


Figure 2.4. Coordinate transformation used in the GTPE, after [1] Figure M.4.

Beginning with the reduced Helmholtz equation, the generalized terrain method transforms the flat terrain coordinates (r, z) to coordinates that follow the arbitrary terrain profile (ξ, η) as:

$$\begin{aligned}\xi &= r \\ \eta &= z - H(r)\end{aligned}\tag{2.34}$$

where, in this section, $H(r)$ is the function of the terrain profile. It defines the height value of the terrain at each range value r and is restricted to terrain with local slopes of up to approximately 30° .

To transform the Helmholtz equation in the form of Equation 2.4 to the new

coordinate system, the derivative relationships

$$\begin{aligned}
\partial_z &= \partial_\eta \\
\partial_z^2 &= \partial_\eta^2 \\
\partial_r &= \partial_\xi - H' \partial_\eta \\
\partial_r^2 &= \partial_\xi^2 - 2H' \partial_{\xi\eta}^2 - H'' \partial_\eta + H'^2 \partial_\eta^2
\end{aligned} \tag{2.35}$$

are used, where primes denote derivatives with respect to the range variable ξ .

Substituting the transform derivatives into the Helmholtz equation gives

$$\partial_\xi^2 q - 2H' \partial_{\xi\eta}^2 q - H'' \partial_\eta q + (H'^2 + 1) \partial_\eta^2 q + k^2 q = 0. \tag{2.36}$$

The form of the solution used for the transformed coordinate system, corresponding to Equation 2.8 and for flat terrain, is

$$q(\xi, \eta) = \psi(\xi, \eta) e^{ik_a \xi} \tag{2.37}$$

and the equivalent differential equation for the envelope function, corresponding to Equation 2.14 for flat terrain, is

$$\partial_\xi^2 \psi + 2ik_a \partial_\xi \psi - 2H' (\partial_{\xi\eta}^2 \psi + ik_a \partial_\eta \psi) - H'' \partial_\eta \psi + (H'^2 + 1) \partial_\eta^2 \psi + (k^2 - k_a^2) \psi = 0. \tag{2.38}$$

The first-order approximation of Equation 2.38 is found by neglecting $\partial_\xi^2 \psi$ and $\partial_{\xi\eta}^2 \psi$ to get

$$\partial_\xi \psi = \frac{i}{2k_a} L_1(\psi), \tag{2.39}$$

where

$$L_1 = \alpha \partial_\eta^2 - \beta \partial_\eta + \gamma \tag{2.40}$$

and

$$\begin{aligned}
\alpha(\xi) &= H'^2 + 1 \\
\beta(\xi) &= 2ik_a H' + H'' \\
\gamma(\eta) &= k^2(\eta) - k_a^2.
\end{aligned} \tag{2.41}$$

The second order (wide-angle) generalized terrain parabolic equation is found by integrating Equation 2.38 over one range step from $\xi = a$ to $\xi = b = a + \Delta\xi$, and using the first-order approximation, Equation 2.39, to substitute for the integral

of $\partial_\xi^2 \psi$. Integrating Equation 2.38, therefore, gives

$$\left[\frac{i}{2k_a} L_1(\psi) + 2ik_a \psi - 2H' \partial_\eta \psi \right]_a^b + I_\alpha + I_\chi + I_\gamma = 0 \quad (2.42)$$

where the term $-2H' \partial_\eta^2 \psi$ has been integrated by parts as

$$\int_a^b -2H' \partial_\eta^2 \psi d\xi = -2H' \partial_\eta \psi + \int_a^b 2H'' \partial_\eta \psi d\xi, \quad (2.43)$$

and

$$\begin{aligned} I_\alpha &= \int_a^b \alpha(\xi) \partial_\eta^2 \psi d\xi \\ I_\chi &= \int_a^b \chi(\xi) \partial_\eta \psi d\xi \\ I_\gamma &= \int_a^b \gamma(\eta) \psi d\xi \end{aligned} \quad (2.44)$$

where

$$\chi(\xi) = H'' - 2ik_a H'. \quad (2.45)$$

The integrals of Equation 2.44 can be written in the general form

$$I_R = \int_a^b R(\xi) \partial_\eta^n \psi d\xi \quad (2.46)$$

where $n = 0, 1$, and 2 for $R = \gamma, \chi$, and α , respectively. The derivatives $\partial_\eta^n \psi$ in the integrals of Equation 2.44 are approximated assuming a linear variation between range steps $\xi = a$ and b as

$$\partial_\eta^n \psi(\xi) = \frac{b - \xi}{\Delta\xi} \partial_\eta^n \psi(a) + \frac{\xi - a}{\Delta\xi} \partial_\eta^n \psi(b). \quad (2.47)$$

Substituting Equation 2.47 into 2.46 yields an equation of the form

$$I_R = A_R \partial_\eta^n \psi(a) + B_R \partial_\eta^n \psi(b) \quad (2.48)$$

where

$$\begin{aligned} A_R &= \frac{1}{\Delta\xi} \int_a^b (b - \xi) R(\xi) d\xi \\ B_R &= \frac{1}{\Delta\xi} \int_a^b (\xi - a) R(\xi) d\xi \end{aligned} \quad (2.49)$$

Similarly, $R(\xi)$ in the integrals of Equation 2.44 is approximated assuming a linear

variation between range steps $\xi = a$ and b as

$$R(\xi) = \frac{b - \xi}{\Delta\xi} R(a) + \frac{\xi - a}{\Delta\xi} R(b). \quad (2.50)$$

Evaluating the integrals of Equation 2.49 with the approximations of Equation 2.50 gives

$$\begin{aligned} A_R &= \Delta\xi \left[\frac{1}{3} R(a) + \frac{1}{6} R(b) \right] \\ B_R &= \Delta\xi \left[\frac{1}{6} R(a) + \frac{1}{3} R(b) \right]. \end{aligned} \quad (2.51)$$

Just as in the flat terrain CNPE derivation, a finite-difference form of the GTPE can be reached by applying central difference approximations to the height variable derivatives as

$$\begin{aligned} (\partial_\eta \psi)_{\eta_j} &= \frac{\psi_{j+1} - \psi_{j-1}}{2\Delta\eta} \\ (\partial_\eta^2 \psi)_{\eta_j} &= \frac{\psi_{j+1} - 2\psi_j + \psi_{j-1}}{(\Delta\eta)^2} \end{aligned} \quad (2.52)$$

where $\eta_j = j\Delta\eta$ with $j = 1, 2, \dots, M$, and $\psi_j = \psi(\xi, \eta_j)$. Thus, the integrated equation, Equation 2.42, can be rewritten in the same form as Equation 2.24 for the flat terrain CNPE as

$$M_2 \vec{\psi}(b) = M_1 \vec{\psi}(a) \quad (2.53)$$

where M_1 and M_2 are the tridiagonal matrices

$$\begin{aligned} M_2 &= c_3 \delta^2 + c_2 \delta + c_1 \\ M_1 &= d_3 \delta^2 + d_2 \delta + d_1 \end{aligned} \quad (2.54)$$

with which the sound field at one range step can be extrapolated from the sound field at the previous range step. The coefficients are given as

$$\begin{aligned} c_3 &= \frac{1}{2ik_a(\Delta\eta)^2} \left(\frac{i\alpha(b)}{2k_a} + B_\alpha \right) \\ c_2 &= -\frac{1}{2ik_a\Delta\eta} \left(\frac{i\beta(b)}{2k_a} + 2H'(b) - B_\chi \right) \\ c_1 &= 1 + \frac{\gamma}{4k_a^2} + \frac{B_\gamma}{2ik_a} \end{aligned} \quad (2.55)$$

where $\alpha(b)$, $\beta(b)$, and $H'(b)$ refer to each function evaluated at the range point

$\xi = b$ and

$$\begin{aligned} d_3 &= \frac{1}{2ik_a(\Delta\eta)^2} \left(\frac{i\alpha(a)}{2k_a} - A_\alpha \right) \\ d_2 &= -\frac{1}{2ik_a\Delta\eta} \left(\frac{i\beta(a)}{2k_a} + 2H'(a) + A_\chi \right) \\ d_1 &= 1 + \frac{\gamma}{4k_a^2} - \frac{A_\gamma}{2ik_a}. \end{aligned} \tag{2.56}$$

where function evaluation is at point $\xi = a$, instead of b . The tridiagonal matrices δ^2 and δ are given as

$$\delta^2 = \begin{bmatrix} -2 & 1 & & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -2 \end{bmatrix} \tag{2.57}$$

and

$$\delta = \frac{1}{2} \begin{bmatrix} 0 & 1 & & & & & \\ -1 & 0 & 1 & & & & \\ & -1 & 0 & 1 & & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -1 & 0 & 1 \\ & & & & & -1 & 0 \end{bmatrix}. \tag{2.58}$$

2.2.2.1 Grid Boundary Conditions

This section discusses the treatment of the lower and upper boundaries of the grid to include the effects of a sloping ground surface, and the correspondingly sloping sky boundary. An upper absorbing layer is again used to attenuate sound before it reaches the upper grid boundary. Because the implementation of the absorbing layer is the same as described in Section 2.2.1.1.2, no further discussion is included in this section.

The lower boundary condition is applied in a similar way as in the flat terrain CNPE. The derivation begins with a boundary condition equation similar to

Equation 2.27

$$\left(\frac{p}{v_{\mathbf{n}}} \right)_{\eta=0} = Z\rho c. \quad (2.59)$$

However, the linearized Euler equation, corresponding to Equation 2.28 in the flat terrain CNPE derivation, which is used to relate the velocity normal to the surface to the pressure, becomes

$$v_{\mathbf{n}} = \frac{1}{i\omega\rho} \frac{\partial p}{\partial \mathbf{n}} \quad (2.60)$$

where \mathbf{n} , the unit normal vector to the ground, pointing downward, has replaced the vertical coordinate z in the derivative of pressure. An illustration of the grid geometry and associated variable parameters is shown in Figure 2.5.

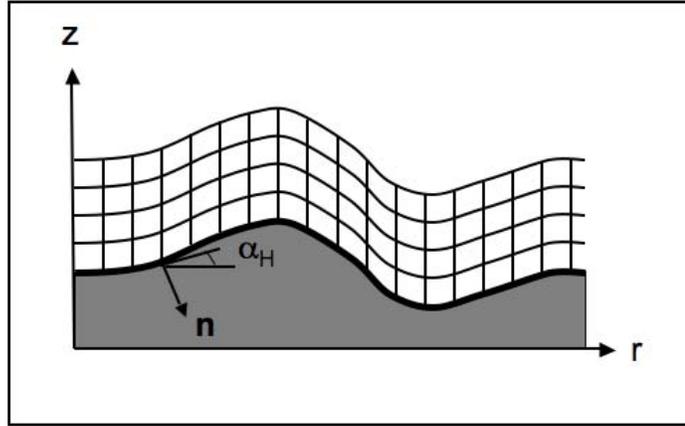


Figure 2.5. Diagram of the ground angle in the GTPE grid, after [1] Figure M.5.

The normal \mathbf{n} can be broken into horizontal and vertical components as $\mathbf{n} = (n_r, n_z)$, allowing the derivative of pressure to be written as

$$\frac{\partial p}{\partial \mathbf{n}} = n_r \frac{\partial p}{\partial r} + n_z \frac{\partial p}{\partial z}. \quad (2.61)$$

The unit normal vector can also be written as a function of the local elevation angle α_H as

$$\mathbf{n} = (\sin \alpha_H, -\cos \alpha_H) \quad (2.62)$$

where α_H is related to the derivative of the function of the terrain profile as $H'(r) = \tan \alpha_H$. Substituting Equation 2.62 and the derivative relationships of Equation

2.35 into Equation 2.61 gives

$$\frac{\partial p}{\partial \mathbf{n}} = \sin \alpha_H \frac{\partial p}{\partial \xi} - \frac{1}{\cos \alpha_H} \frac{\partial p}{\partial \eta}. \quad (2.63)$$

Substituting Equation 2.63 into Equation 2.59 gives

$$\frac{ik_0}{Z} p_0 = \sin \alpha_H \left(\frac{\partial p}{\partial \xi} \right)_{\eta=0} - \frac{1}{\cos \alpha_H} \left(\frac{\partial p}{\partial \eta} \right)_{\eta=0} \quad (2.64)$$

where p_0 and k_0 are pressure and wave number evaluated just above the ground surface.

Equation 2.64 can be written in terms of the envelope function ψ by substituting in the relationship $q = p\sqrt{r}$ and solution form, Equation 2.37 to get

$$\frac{ik_0}{Z} \psi_0 = \sin \alpha_H \left[\left(\frac{\partial \psi}{\partial \xi} \right)_{\eta=0} + ik_a \psi_0 \right] - \frac{1}{\cos \alpha_H} \left(\frac{\partial \psi}{\partial \eta} \right)_{\eta=0} \quad (2.65)$$

where the term $-(\frac{1}{2}\psi_0/\xi)\sin\alpha_H$ has been neglected. Using finite difference approximations, an equation for ψ_0 , analogous to Equation 2.17 for the flat terrain CNPE, can be obtained.

To maintain second-order accuracy, a second-order finite difference approximation is used in place of the derivatives $\left(\frac{\partial \psi}{\partial \xi} \right)_{\eta=0}$ and $\left(\frac{\partial \psi}{\partial \eta} \right)_{\eta=0}$. The approximations are found with a similar method as used in Equation 2.29:

- The derivative in ξ is found using three consecutive points in range at ground height: $\psi_0(\bar{a})$, $\psi_0(a)$, and $\psi_0(b)$, where $\bar{a} = a - \Delta\xi$, as

$$\left(\frac{\partial \psi}{\partial \xi} \right)_{\eta=0} = \frac{1}{2\Delta\xi} [-4\psi_0(a) + 3\psi_0(b) + \psi_0(\bar{a})]. \quad (2.66)$$

- The derivative in η is found using three consecutive points in height beginning at the ground, at the specified point in range, for example b : $\psi_0(b)$, $\psi_1(b)$, and $\psi_2(b)$, as

$$\left(\frac{\partial \psi}{\partial \eta} \right)_{\eta=0} = \frac{1}{2\Delta\eta} [4\psi_1(b) - 3\psi_0(b) - \psi_2(b)]. \quad (2.67)$$

Equations 2.66 and 2.67 are substituted into Equation 2.65 to get an expression for $\psi_0(b)$ in terms of neighboring grid points as

$$\psi_0(b) = u\psi_1(b) + v\psi_2(b) + w\psi_0(a) + y\psi_0(\bar{a}) \quad (2.68)$$

where

$$u = \frac{4}{d\epsilon}, \quad v = -\frac{u}{4}, \quad w = \frac{2 \sin \alpha_H}{d\Delta\xi}, \quad y = -\frac{w}{4}, \quad (2.69)$$

with

$$\epsilon = 2\Delta\eta \cos \alpha_H \quad (2.70)$$

and

$$d = -\frac{ik_0}{Z} + \frac{3}{\epsilon} + \left(\frac{3/2}{\Delta\xi} + ik_a \right) \sin \alpha_H. \quad (2.71)$$

Z and α_H in the previous equations are evaluated at range step b . (Salomons [1] identifies an error in [44] in the representation of the previous equation. Therefore, the correct equation from [1] is used.)

A similar equation can be written for $\psi_0(a)$ as

$$\psi_0(a) = u\psi_1(a) + v\psi_2(a) + w\psi_0(\bar{a}) + y\psi_0(\bar{\bar{a}}) \quad (2.72)$$

where $\bar{\bar{a}} = a - 2\Delta\xi$, and Z and α_H , as used in Equations 2.69, 2.70, and 2.71 are evaluated at $\xi = a$.

The boundary condition equations for the top of the grid are found analogously to those for the grid bottom, with adjustments to certain terms:

- $\sin \alpha_H$ and $\cos \alpha_H$ become $-\sin \alpha_H$ and $-\cos \alpha_H$, respectively because the unit normal is flipped to point upward as \mathbf{n} to $-\mathbf{n}$
- $\Delta\eta$ becomes $-\Delta\eta$,
- k_0 becomes k_{M+1} ,
- and Z is set to 1 for the normalized impedance of the air.

Equations 2.68 through 2.71, therefore, are replaced with

$$\psi_{M+1}(b) = u_t\psi_M(b) + v_t\psi_{M-1}(b) + w_t\psi_{M+1}(a) + y_t\psi_{M+1}(\bar{a}) \quad (2.73)$$

Still, these matrix modifications cannot incorporate the values of the last two terms of Equations 2.68 and 2.73, which are evaluated at $\xi = a$ and \bar{a} , respectively.

Therefore, vectors k_3 and k_2 :

$$k_3 = \begin{bmatrix} w\psi_0(a) + y\psi_0(\bar{a}) \\ 0 \\ 0 \\ \vdots \\ 0 \\ w_t\psi_{M+1}(a) + y_t\psi_{M+1}(\bar{a}) \end{bmatrix} \quad (2.80)$$

$$k_2 = \frac{1}{2} \begin{bmatrix} -w\psi_0(a) - y\psi_0(\bar{a}) \\ 0 \\ 0 \\ \vdots \\ 0 \\ w_t\psi_{M+1}(a) + y_t\psi_{M+1}(\bar{a}) \end{bmatrix} \quad (2.81)$$

are added in as well so that $\delta^2\vec{\psi}(b)$ is replaced with $T_3\vec{\psi}(b) + k_3$ and $\delta\vec{\psi}(b)$ is replaced with $T_2\vec{\psi}(b) + k_2$. The variables $u, v, w, y, u_t, v_t, w_t$, and y_t in the matrix representations of T_3, T_2, k_3 , and k_2 are evaluated at $\xi = b$.

Similarly, for evaluation at $\xi = a$, $\delta^2\vec{\psi}(a)$ is replaced with $S_3\vec{\psi}(a) + m_3$ and $\delta\vec{\psi}(a)$ is replaced with $S_2\vec{\psi}(a) + m_2$, where S_3 and S_2 are equal to T_3 and T_2 , respectively for u, v, u_t , and v_t evaluated at $\xi = a$. Vectors m_3 and m_2 become

$$m_3 = \begin{bmatrix} w\psi_0(\bar{a}) + y\psi_0(\bar{\bar{a}}) \\ 0 \\ 0 \\ \vdots \\ 0 \\ w_t\psi_{M+1}(\bar{a}) + y_t\psi_{M+1}(\bar{\bar{a}}) \end{bmatrix} \quad (2.82)$$

$$m_2 = \frac{1}{2} \begin{bmatrix} -w\psi_0(\bar{a}) - y\psi_0(\bar{a}) \\ 0 \\ 0 \\ \vdots \\ 0 \\ w_t\psi_{M+1}(\bar{a}) + y_t\psi_{M+1}(\bar{a}) \end{bmatrix} \quad (2.83)$$

where $w, y, w_t,$ and y_t also evaluated at $\xi = a$.

Equation 2.53 has, therefore, been modified to incorporate the boundary conditions. The final form of the equation is

$$(c_3T_3 + c_2T_2 + c_1)\vec{\psi}(b) + c_3k_3 + c_2k_2 = (d_3S_3 + d_2S_2 + d_1)\vec{\psi}(a) + d_3m_3 + d_2m_2 \quad (2.84)$$

where, in review, $T_3, T_2, c_1, S_3, S_2,$ and d_1 are matrices, $\vec{\psi}, k_3, k_2, m_3,$ and m_2 are vectors, and $c_3, c_2, d_3,$ and d_2 are scalars.

2.3 Numerical Implementation

The standard Crank-Nicholson and Generalized Terrain formulations of the parabolic equation method were derived in the earlier part of the chapter. In the remaining sections, numerical considerations and details of their implementation will be discussed. Most of the considerations are common to both the CNPE and the GTPE. However, the final section applies to the GTPE only.

2.3.1 Grid Parameters

The PE method places several constraints on the spatial grid of sound propagation calculations. One such constraint is on grid step size. By the Nyquist theorem, at least two points per wavelength are necessary to sufficiently represent a wave. However, using a finite difference approach as the Crank-Nicholson formulation does, sufficient numerical accuracy is obtained with a smaller computational step size of approximately one-tenth of the acoustic wavelength λ in both range and height [1]. These criterion are not applicable for the Green's Function PE in range, for which steps of many wavelengths can be used.

Use of a step size criterion based solely on wave number would yield different grid spacings for different frequencies, making comparison and combination difficult. It is desirable, instead, to both meet the minimum grid spacing criteria and have data for every frequency specified at common grid points. The range spacing is, therefore, chosen to be the smallest integer divisor of the user-specified spatial resolution of range that yields a step size less than $\frac{\lambda}{10}$. This allows each frequency to be defined at common points so that results of different frequencies can be easily added together, as shown for three example frequencies in Figure 2.6. For these three frequencies, the natural range step choice of $\frac{\lambda}{10}$ would not overlap. However, a step size slightly smaller than $\frac{\lambda}{10}$ for each frequency can be chosen to achieve overlapping grid points. If the user-defined spatial resolution is chosen to be too small as compared to a wavelength, however, numerical errors may result for uneven terrain cases at lower frequencies.

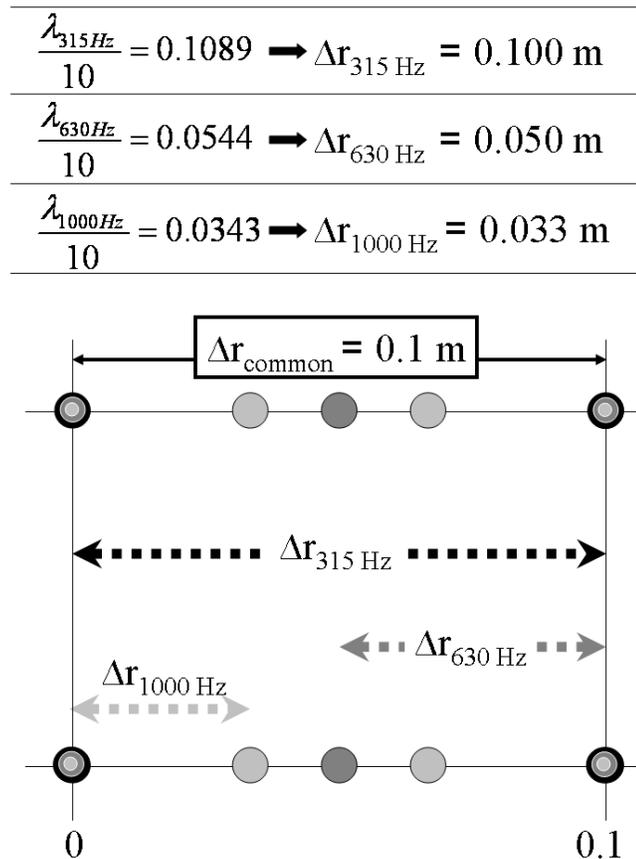


Figure 2.6. Diagram of grid spacing to common points for all frequencies.

The height spacing of the grid is chosen as the smallest integer divisor of the user-specified receiver height that yields a step size less than $\frac{\lambda}{10}$. This allows calculation at the exact height at which results are sought.

Two additional constraints are placed on the vertical axis of the grid.

1. There must be at least 1000 vertical grid points, including the absorbing layer (Section 2.2.1.1.2).
2. The absorbing layer must be at least 50 wavelengths high to sufficiently attenuate the sound before it reaches the upper boundary.

2.3.2 Starting Field

The distinctive form of the PE starting field stems from its propagation process. PE algorithms march results forward in range by extrapolating from vertical vectors of results at previous range steps. Therefore, to begin propagation, sound at the first range step must be fully defined, at all heights.

The PE method algorithms and assumptions also requires that the starting field be finite at all points and restrict energy emitted at large elevation angles from the source. This precludes the use of standard representations of some basic sources in the PE starting field. For example, the standard representation of a monopole diverges at the source and emits sound at large elevation angles. While this restriction prevents accurate representation of the near field for compact sources, far field predictions can be accurate if the starting field closely approximates the physical source at larger distances.

The traditional PE representation for a monopole takes the form of a Gaussian function with the same order as the square-root approximation (Equations 2.9 and 2.12). The Gaussian starting field can be derived from the exact expression for a monopole, applying far field and small elevation angle approximations [48, 52]. It is a smooth function, finite at all points, with limited energy in the larger vertical wave numbers. It, therefore, satisfies the requirements of the PE while serving as an acceptable substitute for the exact monopole expression.

2.3.2.1 Narrow-Angle Starting Field

The starting field for the narrow-angle PE was proposed by Tappert [48]. The following derivations for both the narrow- and wide-angle starting fields adheres closely to those in [1], based on work by Greene [52].

This derivation of the narrow-angle PE starting field assumes a free-field source located at the origin. It begins by considering the narrow angle PE equation, assuming a constant $k(z) = k_a$. The narrow-angle equation, found by substituting Equation 2.9 into Equation 2.7, is

$$\frac{\partial q}{\partial r} - ik_a q - \frac{i}{2k_a} \frac{\partial^2 q}{\partial z^2} = 0. \quad (2.85)$$

The solution to this equation can be written as a plane-wave expansion

$$q = \int_{-\infty}^{\infty} S(k_z) \exp(ik_z z + ik_r r) dk_z, \quad (2.86)$$

which, when substituted into Equation 2.85, yields the relationship between wave numbers:

$$k_r = k_a - \frac{k_z^2}{2k_a}. \quad (2.87)$$

The method of stationary phase can be used to approximate the plane-wave expansion integral. The stationary phase approximation is written as

$$I = \int_a^b g(t) e^{ikh(t)} dt \approx g(t_0) e^{ikh(t_0)} \left(\frac{2\pi}{k|h''(t_0)|} \right)^{1/2} e^{i\mu\pi/4} \quad (2.88)$$

for large k , where

$$\mu = \text{sign}[h''(t_0)], \quad (2.89)$$

$$\text{sign}(x) = \begin{cases} 1 & \text{for } x > 0 \\ -1 & \text{for } x < 0 \end{cases},$$

and t_0 is the point of stationary phase where $h'(t_0) = 0$ [1].

With this approximation, the plane-wave expansion integral is solved as

$$q \approx \sqrt{\frac{2\pi}{|F''(k_{z,0})|}} S(k_{z,0}) \exp \left[iF(k_{z,0}) - i\frac{1}{4}\pi \right] \quad (2.90)$$

where

$$F(k_z) = k_z z + \left(k_a - \frac{k_z^2}{2k_a} \right) r, \quad (2.91)$$

and the point of stationary phase where $F'(k_{z,0}) = 0$ (prime denotes derivatives with respect to k_z) is

$$k_{z,0} = \frac{k_a z}{r}. \quad (2.92)$$

Therefore, q can be rewritten as

$$q \approx \frac{1}{\sqrt{r}} \sqrt{\frac{2\pi k_a}{i}} S(k_{z,0}) \exp \left(i k_a r \left(1 + \frac{z^2}{2r^2} \right) \right). \quad (2.93)$$

Using the narrow-angle approximation $R = \sqrt{r^2 + z^2} \approx r(1 + \frac{1}{2}z^2/r^2)$, and comparing Equation 2.93 to the exact expression for a monopole $q = \sqrt{r} \exp(ik_a R)/R$, the coefficient $S(k_{z,0})$ is found to be

$$S(k_{z,0}) = \sqrt{\frac{i}{2\pi k_a} \frac{r}{R}}. \quad (2.94)$$

To eliminate range dependence, using Equation 2.92, $R = \sqrt{r^2 + z^2} = r \sqrt{1 + \frac{k_{z,0}^2}{k_a^2}}$ can be substituted into Equation 2.94 to get

$$S(k_{z,0}) = \sqrt{\frac{i}{2\pi k_a} \left(1 + \frac{k_{z,0}^2}{k_a^2} \right)^{-1/2}}, \quad (2.95)$$

which corresponds to the form of $S(k_z)$ to be substituted into equation 2.86

$$S(k_z) = \sqrt{\frac{i}{2\pi k_a} \left(1 + \frac{k_z^2}{k_a^2} \right)^{-1/2}}. \quad (2.96)$$

A final small angle approximation is made where $\left(1 + \frac{k_z^2}{k_a^2} \right)^{-1/2} \approx \exp \left(-\frac{k_z^2}{2k_a^2} \right)$ to obtain

$$S(k_z) = \sqrt{\frac{i}{2\pi k_a}} \exp \left(-\frac{k_z^2}{2k_a^2} \right), \quad (2.97)$$

which is substituted back into Equation 2.86 for $r = 0$. The integral is solved and

the standard Gaussian form of the starting field is found to be

$$q(0, z) = \sqrt{ik_a} \exp\left(-\frac{1}{2}k_a^2 z^2\right). \quad (2.98)$$

This PE starting field approximates a monopole source for narrow angles in the far-field.

2.3.2.2 Wide-Angle Starting Field

A similar process can be used to obtain a starting field with second-order accuracy for use in the wide-angle PE. The wide-angle PE equation, Equation 2.13, assuming a constant $k(z) = k_a$, becomes

$$\left(1 + \frac{1}{4k_a^2} \frac{\partial^2}{\partial z^2}\right) \frac{\partial q}{\partial r} - ik_a \left(1 + \frac{3}{4k_a^2} \frac{\partial^2}{\partial z^2}\right) q = 0. \quad (2.99)$$

The plane-wave expansion solution, Equation 2.86, is substituted into this equation to find the relationship between wave numbers

$$k_r = k_a \frac{1 - \frac{3}{4}k_z^2/k_a^2}{1 - \frac{1}{4}k_z^2/k_a^2}. \quad (2.100)$$

The function $F(k_z)$ and its first and second derivatives with respect to k_z are then

$$F(k_z) = k_z z + rk_a \frac{1 - \frac{3}{4}k_z^2/k_a^2}{1 - \frac{1}{4}k_z^2/k_a^2} \quad (2.101)$$

$$F'(k_z) = z - r \frac{k_z/k_a}{\left(1 - \frac{1}{4}k_z^2/k_a^2\right)^2} \quad (2.102)$$

$$F''(k_z) = -\frac{r}{k_a} \frac{1 + \frac{3}{4}k_z^2/k_a^2}{\left(1 - \frac{1}{4}k_z^2/k_a^2\right)^3}. \quad (2.103)$$

The stationary phase point, where $F'(k_{z,0}) = 0$, can be found to second order from Equation 2.102 as

$$k_{z,0} \approx \frac{k_a z}{r}. \quad (2.104)$$

Therefore, using the stationary phase approximation, Equation 2.90, yields

$$q \approx \frac{1}{\sqrt{r}} \sqrt{\frac{2\pi k_a}{i}} \frac{(1 - \frac{1}{4}z^2/r^2)^{3/2}}{(1 + \frac{3}{4}z^2/r^2)^{1/2}} S(k_{z,0}) \exp\left(ik_a r \frac{z^2}{r^2} + ik_a r \frac{1 - \frac{3}{4}\frac{z^2}{r^2}}{1 - \frac{1}{4}\frac{z^2}{r^2}}\right). \quad (2.105)$$

With a geometric series expansion of $\frac{1}{1 - \frac{1}{4}z^2/r^2} \approx 1 + \frac{1}{4}z^2/r^2 + \frac{1}{16}z^4/r^4$, the argument of the exponential, neglecting terms of order $r(z/r)^6$ and higher, becomes $ik_a r(1 + \frac{1}{2}z^2/r^2 - \frac{1}{8}z^4/r^4) \approx ik_a R$ for a second-order narrow-angle approximation of $R = \sqrt{r^2 + z^2} \approx r(1 + \frac{1}{2}z^2/r^2 - \frac{1}{8}z^4/r^4)$. This simplifies the previous equation to

$$q \approx \frac{1}{\sqrt{r}} \sqrt{\frac{2\pi k_a}{i}} \frac{(1 - \frac{1}{4}z^2/r^2)^{3/2}}{(1 + \frac{3}{4}z^2/r^2)^{1/2}} S(k_{z,0}) \exp(ik_a R). \quad (2.106)$$

Comparing this to the exact expression for a monopole $q = \sqrt{r} \exp(ik_a R)/R$, and using the substitution of $R = r\sqrt{1 + z^2/r^2} \approx r\sqrt{1 + k_z^2/k_a^2}$ in the denominator of the exact expression, the coefficient $S(k_z)$ is found to be

$$S(k_z) = \sqrt{\frac{i}{2\pi k_a}} \frac{1}{(1 + k_z^2/k_a^2)^{1/2}} \frac{(1 + \frac{3}{4}k_z^2/k_a^2)^{1/2}}{(1 - \frac{1}{4}k_z^2/k_a^2)^{3/2}}. \quad (2.107)$$

This equation can be written approximately as

$$S(k_z) = \sqrt{\frac{i}{2\pi k_a}} \left(1 + a_2 \frac{k_z^2}{k_a^2}\right) \exp\left(-b_2 \frac{k_z^2}{k_a^2}\right) \quad (2.108)$$

for adjustable parameters a_2 and b_2 . Good agreement is found with $a_2 = 1.02$ and $b_2 = 0.75$. For these parameter values, the starting field becomes

$$q(z, 0) = \sqrt{ik_a} (A_0 + A_2 k_a^2 z^2) \exp\left(-\frac{k_a^2 z^2}{B}\right) \quad (2.109)$$

where $A_0 = 1.3717$, $A_2 = -0.3701$, and $B = 3$. This second-order, wide-angle starting field accurately represents a monopole source to elevation angles up to $\pm 35^\circ$.

Other starting field implementations have been developed for the PE that present more intuitive forms of the source, including an ‘‘analytical start’’ field that uses an exact representation of a monopole and its image source, computed

a small distance in range from the source [53]. However, the standard Gaussian representation is used in the wide-angle PE model discussed in this dissertation.

2.3.2.3 Sources Near the Ground

For a low frequency source near a ground surface, the Gaussian function used to represent a monopole “runs into” the ground. Therefore, to accurately represent the source, the contribution of the image source must be included in the starting field. This section describes a starting field that satisfies the appropriate boundary conditions at the ground, based on the ground’s impedance [1, 45].

To include the image source contribution, the starting field is represented as

$$q(0, z) = q_0(z - z_s) + Cq_0(z + z_s). \quad (2.110)$$

The first term on the right-hand side corresponds to the source assumed to be Gaussian distributed, centered at height z_s . The second term corresponds to the contribution of the image source, representing the sound reflected off the ground surface. C is a general reflection coefficient [1].

The form of the reflection coefficient employed in this code is the spherical-wave reflection coefficient, Equation 1.3, with $r = 0$

$$C_s = 1 - 2 \frac{k_a R_2}{Z} \frac{1}{i} \int_0^\infty \frac{\exp[-qk_a(Z + 1)/Z]}{q + R_2/i} dq \quad (2.111)$$

where $R_2 = z + z_s$, k_1 —the wave number in the air above the ground—becomes k_a , and q is the integration variable, rather than the propagated PE quantity [1]. This equation can be re-written in an exponential integral function form

$$C_s = 1 - 2 \frac{k_a R_2}{Z} \frac{1}{i} \exp\left(k_a \frac{Z + 1 R_2}{Z} \frac{1}{i}\right) E_1\left(k_a \frac{Z + 1 R_2}{Z} \frac{1}{i}\right) \quad (2.112)$$

where E_1 is the exponential integral function [1, 54]. At lower frequencies, this form of the spherical-wave reflection coefficient is used. At higher frequencies, numerical errors arise, and an asymptotic expansion of the spherical wave number

is applied instead

$$C_s = C_p - \frac{2}{Z+1} \sum_{n=1}^{\infty} (-1)^n n! \left(\frac{i}{k_a R_2} \frac{Z}{Z+1} \right)^n \quad (2.113)$$

where C_p is the plane-wave reflection coefficient following Equation 1.2 with $\theta = 0$ as

$$C_p = \frac{Z-1}{Z+1}. \quad (2.114)$$

Comparison of the two spherical-wave reflection coefficient calculation methods suggest that the expansion form, with only two terms, can be used in place of the exponential integral form for larger frequencies with negligible decrease in accuracy.

An example of the contributions of the direct and image sources to the total start field for a 5 m high source over a finite impedance ground can be seen in Figure 2.7. The dashed line shows the source component, the dash-dot line shows the image component, and the solid line shows the total start field. Because the width of the Gaussian distribution is inversely proportional to the frequency, the source function intersects the ground for low frequency, low altitude sources, as indicated by the dashed line. The difference between the total field (solid line) and the source field (dashed line) indicates the significance of the contribution of the image source to the total start field.

2.3.3 Atmospheric Absorption

Atmospheric absorption is included in the propagation model by adding an imaginary component to the acoustic wave number as

$$k = \frac{\omega}{c(z)} + j\beta. \quad (2.115)$$

The atmospheric absorption is often given in the form of the absorption coefficient α , which specifies the attenuation in dB/meter. The absorption coefficient α can be related to β by recognizing that the contribution of β in the exponential adds a coefficient of $\exp(-\beta R)$. Therefore, the attenuation in dB/meter is found as $-\alpha = 20 \log(e^{-\beta R})/R$, or [1]

$$\beta = \alpha / (20 \log e) \quad (2.116)$$

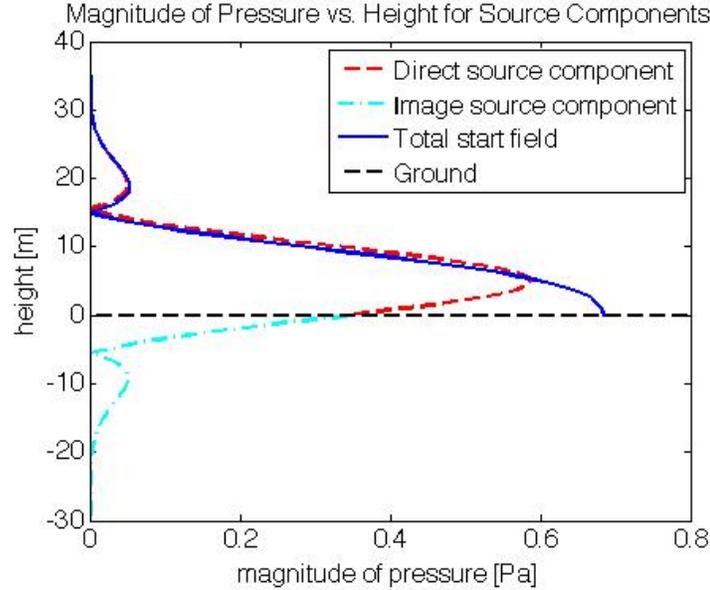


Figure 2.7. Magnitude of the pressure plotted against height for a start field representing a 10 Hz source at 5 m height above a ground with a normalized impedance of $Z = 38.79 + i38.41$. The dashed red line represents the source, the dash-dot cyan line represents the image source, and the solid dark blue line shows the combined contribution of both components.

where \log indicates logarithm base 10. The user may choose to manually assign the atmospheric absorption coefficients for each frequency, or allow the model to calculate it. If the user chooses to let the model calculate the coefficient, the model will calculate using a script based on equations in Bass, et al. [16] for the current frequency, and the specified relative humidity and absolute temperature in kelvins.

2.3.4 Refractive Atmosphere

To define the refractive state of the atmosphere, the code uses an effective sound speed profile, Equation 1.8, which accounts for wind and temperature gradients [1]. Standard linear and logarithmic profiles have been preprogrammed for the effective sound speed, the linear profile defined by Equation 1.6 and the logarithmic profile defined by Equation 1.7.

2.3.5 Turbulence

The turbulent fluctuations in the sound speed due to variations of wind velocity and temperature are included using a refractive-index fluctuation (Equation 1.9). The refractive-index fluctuations is applied through a phase factor in the PE propagation algorithms. This process is repeated for many realizations of turbulence, constituting an ensemble over which the average effect of turbulence on the sound field is obtained [1].

The phase factor is calculated from the refractive-index fluctuation (Section 1.2.4.2) assuming a certain spectral density function for the distribution of the eddy sizes. Gaussian, von Kármán, and Kolmogorov are popular functions used to model the spectrum. Turbulence is often described in terms of three spectrum sub-ranges: the energy-containing subrange, the inertial subrange, and the dissipative subrange. The energy-containing subrange contains the largest eddies; the inertial subrange, within which energy from large eddies gets passed to smaller eddies, which pass energy to even smaller eddies, contains mid-sized eddies; the dissipative subrange, in which energy from the eddies is converted into heat, contains the smallest eddies. The Kolmogorov spectral density function is valid in the inertial subrange. The von Kármán spectral density function follows the Kolmogorov spectral density function in this region and flattens out in the energy-containing subrange. Turbulent eddies in the energy-containing and inertial subranges most significantly impact acoustic propagation as their size is on the order of acoustic wavelengths, while the size of eddies in the dissipative subrange are small compared to acoustic wavelengths [1]. This research uses the two-dimensional the von Kármán spectral density function, calculated as

$$F(k_r, k_z) = \frac{A}{(k^2 + K_0^2)^{8/6}} \left(\frac{\Gamma(\frac{1}{2})\Gamma(\frac{8}{6})}{\Gamma(\frac{11}{6})} \frac{C_T^2}{4T_0^2} + \left[\frac{\Gamma(\frac{3}{2})\Gamma(\frac{8}{6})}{\Gamma(\frac{17}{6})} + \frac{k_z^2}{k^2 + K_0^2} \frac{\Gamma(\frac{1}{2})\Gamma(\frac{14}{6})}{\Gamma(\frac{17}{6})} \right] \frac{22C_v^2}{12c_0^2} \right) \quad (2.117)$$

where $A \approx 0.0330$, Γ is the gamma function, K_0 relates to the scales of the wave numbers used, and C_T^2 and C_v^2 are the structure parameters of turbulent temperature and wind velocity fluctuations, respectively. The two-dimensional von Kármán spectral density function F is proportional to $k^{-8/3}$ in the inertial sub-

range [1]. The turbulent fluctuations are assumed to be homogeneous and isotropic in this research. However, this is most often not the case in realistic atmospheric conditions. For a more complete description of inhomogeneity and anisotropy in turbulence, and their implications, the reader is directed to Wilson's dissertation, "Acoustic tomographic monitoring of the atmospheric boundary layer" [55].

To find the refractive-index fluctuation $\mu(r, z)$ from the spectral density function $F(k_r, k_z)$, an intermediate statistical function is used. The correlation function is defined for a random function $f(\mathbf{r})$ as

$$B(\mathbf{r}_1, \mathbf{r}_2) = \overline{f(\mathbf{r}_1)f(\mathbf{r}_2)} \quad (2.118)$$

where the overbar denotes a time average. If the function $f(\mathbf{r})$ depends only on the difference $\mathbf{r} = \mathbf{r}_1 - \mathbf{r}_2$ and is not a function of either \mathbf{r}_1 or \mathbf{r}_2 alone, it is referred to as homogenous and can be written $B(\mathbf{r}_1, \mathbf{r}_2) = B(\mathbf{r})$.

The correlation function can also be written as the inverse spatial Fourier transform of the spectral density of the random function $f(\mathbf{r})$. In two dimensions, they are related as

$$B(\mathbf{r}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(i\mathbf{k} \cdot \mathbf{r}) F(\mathbf{k}) d\mathbf{k} \quad (2.119)$$

$$F(\mathbf{r}) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(-i\mathbf{k} \cdot \mathbf{r}) B(\mathbf{r}) d\mathbf{r} \quad (2.120)$$

where $\mathbf{r} = (r, z)$ and $\mathbf{k} = (k_r, k_z)$.

The correlation function of the refractive-index fluctuation μ is an even function. For $\mathbf{r}_1 = \mathbf{r} + \mathbf{s}$ and $\mathbf{r}_2 = \mathbf{r}$ the correlation function of the refractive-index fluctuations is defined as

$$B(\mathbf{s}) \equiv \overline{\mu(\mathbf{r} + \mathbf{s})\mu(\mathbf{r})}. \quad (2.121)$$

Using the even characteristic of the function, the exponential function can be replaced by a cosine function in Equation 2.119 to give

$$B(\mathbf{s}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cos(\mathbf{k} \cdot \mathbf{s}) F(\mathbf{k}) d\mathbf{k} \quad (2.122)$$

where $F(\mathbf{k})$ is the two-dimensional spectral density of the refractive-index fluctuation, which was previously defined with Equation 2.117 for a von Kármán

spectrum.

With a relationship between the refractive-index fluctuation μ and its correlation function B (Equation 2.121), and a relationship between the correlation function B and spectral density function F (Equation 2.122), an equation for the refractive-index fluctuation as a function of the spectral density function can be found. To derive the equation, the correlation function is first transformed to a polar $k\theta$ coordinate system where $\mathbf{k} = (k \cos \theta, k \sin \theta)$

$$B(\mathbf{s}) = \int_0^{2\pi} \int_0^\infty \cos(\mathbf{k} \cdot \mathbf{s}) F(\mathbf{k}) k dk d\theta. \quad (2.123)$$

This can be further rewritten as

$$B(\mathbf{s}) = 2\pi \langle \int_0^\infty \cos(\mathbf{k} \cdot \mathbf{s}) F(\mathbf{k}) k dk \rangle_\theta \quad (2.124)$$

for which the integral over θ has been evaluated by defining the bracketed term $\langle . \rangle_\theta$ to be the average over the angle θ .

For numerical implementation, the integral is converted to a summation with a discretized wave number \mathbf{k}_n as

$$B(\mathbf{s}) = 2\pi \langle \Delta k \sum_n \cos(\mathbf{k}_n \cdot \mathbf{s}) F(\mathbf{k}_n) k_n \rangle_\theta. \quad (2.125)$$

It is proven in the following lines that a random realization of the random refractive-index fluctuations field $\mu(\mathbf{r})$ can be found from the spectral density function of the turbulent eddies $F(\mathbf{k})$ as

$$\mu(r, z_j) = \sum_n G(\mathbf{k}_n) \cos(rk_{nr} + z_j k_{nz} + \alpha_n) \quad (2.126)$$

where $G(\mathbf{k}_n) = \sqrt{4\pi\Delta k F(\mathbf{k}_n) k_n}$, $k_{nr} = k_n \cos \theta_n$, and $k_{nz} = k_n \sin \theta_n$. The angles θ_n and α_n are random angles assigned for different turbulence mode wave numbers where θ_n indicates the direction of the mode's turbulent eddy structure in space and α_n adds a random initial phase [1].

From the initial definition in Equation 2.121, the relationship between correla-

tion function and random function μ can be written as

$$B(\mathbf{s}) = \langle \mu(\mathbf{r} + \mathbf{s})\mu(\mathbf{r}) \rangle_{\theta, \alpha}. \quad (2.127)$$

Here the bracketed term is defined to be the average over the random angles θ and α , which represent the different random realizations of the refractive-index fluctuations.

Substituting Equation 2.126 into Equation 2.127 yields

$$B(\mathbf{s}) = 4\pi\Delta k \langle \sum_n \cos(\mathbf{k}_n \cdot \mathbf{r} + \alpha_n + \mathbf{k}_n \cdot \mathbf{s}) \sqrt{F(\mathbf{k}_n)k_n} \times \sum_m \cos(\mathbf{k}_m \cdot \mathbf{r} + \alpha_m) \sqrt{F(\mathbf{k}_m)k_m} \rangle_{\theta, \alpha}. \quad (2.128)$$

For $n \neq m$, the average of the term

$$\cos(\mathbf{k}_n \cdot \mathbf{r} + \alpha_n + \mathbf{k}_n \cdot \mathbf{s}) \sqrt{F(\mathbf{k}_n)k_n} \cos(\mathbf{k}_m \cdot \mathbf{r} + \alpha_m) \sqrt{F(\mathbf{k}_m)k_m} \quad (2.129)$$

is zero. After manipulating the equations using trigonometric relationships, the average this term for $n = m$ becomes

$$\begin{aligned} & \langle \cos^2(\mathbf{k}_n \cdot \mathbf{r} + \alpha_n) \cos(\mathbf{k}_n \cdot \mathbf{s}) F(\mathbf{k}_n) k_n \\ & \quad - \sin(\mathbf{k}_n \cdot \mathbf{r} + \alpha_n) \cos(\mathbf{k}_n \cdot \mathbf{r} + \alpha_n) \sin(\mathbf{k}_n \cdot \mathbf{s}) F(\mathbf{k}_n) k_n \rangle_{\theta, \alpha} \\ & = \frac{1}{2} \langle \cos(\mathbf{k}_n \cdot \mathbf{s}) F(\mathbf{k}_n) k_n \rangle_{\theta}. \end{aligned} \quad (2.130)$$

When Equation 2.130 is substituted back into Equation 2.128, Equation 2.125 is obtained, verifying the above-stated equation for $\mu(\mathbf{r})$ (Equation 2.126). The refractive-index can found with more computational efficiency by rewriting the cosine function of Equation 2.126 as

$$\cos(rk_{nr} + z_j k_{nz} + \alpha_n) = \mathbb{R} \left\{ \exp(ik_{nr}r + i\alpha_n) [\exp(ik_{nz}\Delta z)]^j \right\} \quad (2.131)$$

where $z_j = j\Delta z$ and \mathbb{R} indicates the real part.

The turbulent phase factor is applied in two factors to achieve a slight increase in accuracy. One factor is applied after the propagation algorithms from range step r to $r + \Delta r$, multiplying the field by $\exp[\frac{1}{2}ik_a\mu(r, z)\Delta r]$ and the other is applied before propagation from range step $r + \Delta r$ to $r + 2\Delta r$ multiplying by

$\exp[\frac{1}{2}ik_a\mu(r + \Delta r, z)\Delta r]$. The effective refractive-index fluctuation is, therefore, equivalent to

$$\mu_2 = \frac{1}{2}[\mu(r, z) + \mu(r + \Delta r, z)], \quad (2.132)$$

resulting in a phase factor of $\exp(ik_a\mu_2\Delta r)$.

Additional discussion of turbulence, including a more detailed description of user-input parameters can be found in Appendix C.

2.3.6 Ground Impedance

Ground effects are included by defining variables σ_1 and σ_2 based on the ground impedance, as calculated in the CNPE with Equation 2.31, or variable d , as calculated in the GTPE with Equation 2.71.

The user may choose to manually assign the complex, normalized impedance of the ground for a particular frequency, or allow the model to calculate it by specifying the flow resistivity parameter that corresponds to the desired ground type as outlined in the work of Delany and Bazley [19], and extended by Embleton, et al. [8] and Chessell [20]. If the user chooses to let the model calculate the impedance, the model will calculate using Equation 1.1.

To include a transition from one type of ground to another, the value of the impedance is changed between range steps, so that the impedance value remains constant within each step [1]. Diffraction theory applied at an impedance discontinuity shows that some sound is diffracted in the negative r direction. However, the parabolic equation method cannot account for backscattering and is, therefore, unable to include this effect [28].

2.3.7 GTPE-Specific Numerical Implementation

Because of its added terrain capability, the GTPE requires additional implementation description.

2.3.7.1 Terrain

To include terrain effects, an arbitrary terrain profile can be specified by defining the height of the ground at each range point [1, 44]. The local slope of the ground

profile input should not exceed 30° . The derivatives of the terrain profile function H' and H'' are found using a finite difference approach.

2.3.7.2 First Three Range Steps

The GTPE requires extrapolation from not just the previous range step, but the previous 3 range steps. Therefore, the first 3 range steps are calculated with the simpler, flat terrain CNPE formulation that extrapolates from only the previous range step. Once results for the first 3 columns are calculated, the GTPE has enough information about the previous range steps to be implemented.

2.4 Numerical Examples

Several numerical examples, shown in Figure 2.8, are presented to illustrate the effects of different conditions on the interference patterns of propagated, single-frequency (100 Hz) sound; they are plotted as the sound pressure level relative to free field:

Figure 2.8a shows sound propagated in a homogeneous atmosphere over a flat, rigid ground. Because of the limitations of the PE formulation, the accuracy of the PE model is degraded at elevation angles larger than approximately 35° . Therefore, these regions in Figures 2.8a-e have been marked by dash-dot lines.

Figure 2.8b shows the effect of uneven terrain, where the terrain feature is filled in with white. A comparison of Figures 2.8a and 2.8b illustrates the distortion of the interference pattern as sound passes over a hill, and the shielding effect on the region behind the hill, shown as a marked decrease in sound level.

Figure 2.8c shows the effect of a finite ground impedance discontinuity where there is a transition from rigid to soft ground at 300 m in range. A comparison of Figure 2.8a and 2.8c illustrates a broadening of the first destructive interference dip and a decrease in the level of the constructive interference peak near the ground, beginning after sound passes over the impedance discontinuity at 300 m.

Figure 2.8d shows the effect of an upward-refracting atmosphere. A comparison of Figure 2.8a and 2.8d illustrates that the shapes, locations, and levels of the peaks and dips have shifted in the refractive atmosphere, especially at large ranges and low altitudes where a shadow zone has formed.

Figure 2.8e shows the effect of turbulence. It is a composite figure obtained by averaging over 50 individual realizations of sound speed fluctuations caused by strong turbulence. A comparison of Figure 2.8d and 2.8e (both with upward refracting atmospheres) illustrates an increase in the sound levels within the interference pattern minima and in shadow regions when turbulence is present.

By comparing Figures 2.8a–e, it can be seen that including each propagation effect has consequences for the resulting sound field. If these effects are neglected by a noise propagation model when physical conditions dictate that they be included, inaccurate results will be obtained.

2.5 Conclusions

In this chapter, the parabolic equation method formulation included in the hybrid propagation model was derived. The details of the model's implementation were presented, along with numerical examples to demonstrate the strengths of the model and justification for its use. While the PE model is used as the foundation of the hybrid model because it can include a wide range of range-dependent propagation effects, limitations call for a method to compensate for its degraded accuracy at large elevation angles. Therefore, the fast field program (FFP) method, described in the following chapter, is used in combination with the PE in the hybrid model.

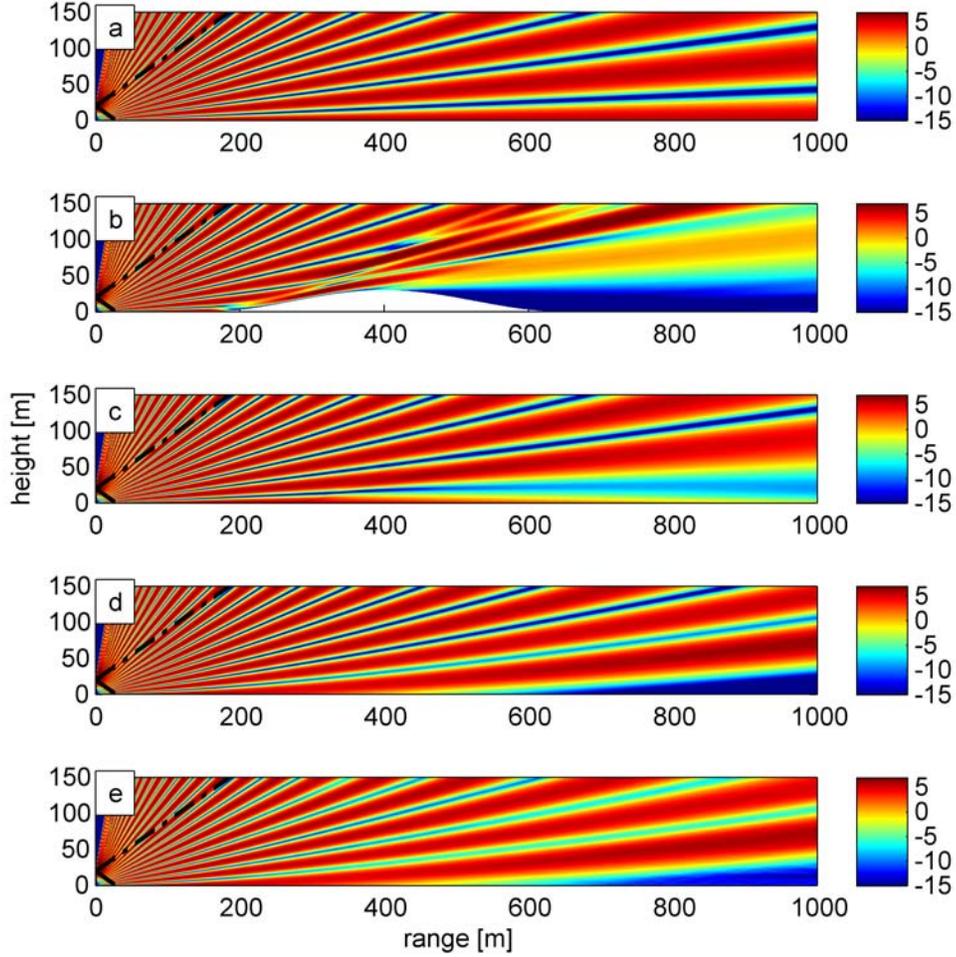


Figure 2.8. Sound pressure level relative to free field [dB] for 100 Hz source at 20 m height calculated with PE. (a) Homogeneous atmosphere, flat, rigid ground. (b) Homogeneous atmosphere, rigid ground with hill-peak at range of 400 m, height of 30 m, and width of 500 m. (c) Homogeneous atmosphere, flat ground with $\sigma = \infty$ for range 0–300 m and $\sigma = 150$ cgs Rayls for range 300–1000 m. σ is the effective flow resistivity. (d) Refractive atmosphere: $b = -1$ m/s, flat, rigid ground. b is the parameter of logarithmic sound speed profile. (e) Refractive atmosphere: $b = -1$ m/s, flat, rigid ground, 50 realizations of turbulence, $C_T^2/T_0^2 = 6 * 10^{-7} m^{-2/3}$, $C_v^2/c_0^2 = 2 * 10^{-6} m^{-2/3}$. C_T^2 and C_v^2 are the structure parameters of turbulent temperature and wind velocity fluctuations, respectively.

The Two-Dimensional Fast Field Program Model

3.1 Introduction

Because of the elevation angle limitation of the PE method formulation, the fast field program (FFP) method [56, 57] is used in the hybrid model as a supplement to fill in the areas where the PE's accuracy is diminished. In exchange for the FFP's large elevation angle accuracy, the user endures the limitation of a homogeneous ground surface and a layered (non-range-dependent) atmosphere. However, over short ranges, range-dependent effects are assumed to be small.

Like the PE, the FFP gained acceptance in the underwater acoustics community before being applied to atmospheric acoustic propagation. It was developed in the early 1970s [58, 59, 60] for underwater applications, and extended to atmospheric propagation in the 1980s by Raspert et al. [61] and Lee et al. [62].

The FFP used in this research calculates the sound field in two dimensions, although the FFP has also been extended to three-dimensional formulations [63, 64]. The FFP is based on an approximate form of the wave equation that employs a transform from the horizontal spatial domain to the horizontal wave number domain. The equation is solved numerically within each stratified layer of the atmosphere, and transformed back to the spatial domain. This chapter outlines the FFP model derivation, following [1] and [57] closely, and details the model

implementation.

3.2 FFP Derivation

The FFP is derived from the inhomogeneous Helmholtz equation for a monopole source. The pressure radiated by a monopole source with unit amplitude, in the free-field is $p(R) = \frac{\exp(ikR)}{R}$, where R is radial distance from the source. The corresponding inhomogeneous Helmholtz equation is written as

$$\nabla^2 p + k^2 p = -4\pi\delta(\mathbf{r} - \mathbf{r}_s) \quad (3.1)$$

where k is the acoustic wave number and, in Cartesian coordinates, $\mathbf{r}_s = (x_s, y_s, z_s)$ and $\delta(\mathbf{r} - \mathbf{r}_s) = \delta(x - x_s)\delta(y - y_s)\delta(z - z_s)$. The wave number k is replaced by the effective wave number k_{eff} , based on the effective sound speed profile approximation (Section 1.2.4.1). To find the inhomogeneous Helmholtz equation in the horizontal wave number domain, the two-dimensional Fourier transform

$$P(k_x, k_y, z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(-ik_x x - ik_y y) p(\mathbf{r}) dx dy \quad (3.2)$$

is evaluated over x and y , where k_x and k_y are the x and y components of the effective wave number, respectively. Applying this transform to the Helmholtz equation and assuming that the effective wave number $k_{\text{eff}}(z)$ is a function of height only yields [1]

$$k_{\text{eff}}^2 \frac{\partial}{\partial z} \left(k_{\text{eff}}^{-2} \frac{\partial P}{\partial z} \right) + (k_{\text{eff}}^2 - k_x^2 - k_y^2) P = -4\pi\delta(z - z_s). \quad (3.3)$$

In the two-dimensional formulation of the FFP, an axisymmetry approximation is applied to reduce the problem from three dimensions to two (as is done in the PE). Therefore, a transformation from the Cartesian coordinate system to a cylindrical coordinate representation of the Fourier transform is made for convenience. Substituting $x = r \cos \phi$ and $y = r \sin \phi$, the Fourier transform becomes

$$P(k_x, k_y, z) = \int_0^{2\pi} \int_0^{\infty} \exp(-ik_x r \cos \phi - ik_y r \sin \phi) p(\mathbf{r}) r dr d\phi. \quad (3.4)$$

This form of the transform can be substituted into Equation 3.3 as

$$\begin{aligned}
& k_{\text{eff}}^2 \frac{\partial}{\partial z} \left(k_{\text{eff}}^{-2} \frac{\partial}{\partial z} \left[\int_0^{2\pi} \int_0^\infty \exp(-ik_x r \cos \phi - ik_y r \sin \phi) p(\mathbf{r}) r dr d\phi \right] \right) \\
& + (k_{\text{eff}}^2 - k_x^2 - k_y^2) \left[\int_0^{2\pi} \int_0^\infty \exp(-ik_x r \cos \phi - ik_y r \sin \phi) p(\mathbf{r}) r dr d\phi \right] \\
& = -4\pi \delta(z - z_s).
\end{aligned} \tag{3.5}$$

With the assumption of a homogeneous atmosphere within each horizontal stratum, and the axisymmetry approximation, k_{eff} , k_x , and k_y are functions only of z and can be brought inside the integrals. In addition, as in the parabolic equation method, the quantity $q = p\sqrt{r}$ replaces pressure and the Helmholtz equation in the horizontal wave number domain becomes

$$\begin{aligned}
& \int_0^{2\pi} \int_0^\infty \exp(-ik_x r \cos \phi - ik_y r \sin \phi) \left[k_{\text{eff}}^2 \frac{\partial}{\partial z} \left(k_{\text{eff}}^{-2} \frac{\partial q}{\partial z} \right) + (k_{\text{eff}}^2 - k_x^2 - k_y^2) q \right] \sqrt{r} dr d\phi \\
& = -4\pi \delta(z - z_s).
\end{aligned} \tag{3.6}$$

Next, the Cartesian wave number components are replaced with their cylindrical counterparts $k_x = k_r \cos \psi$ and $k_y = k_r \sin \psi$, and the trigonometric relationships $\cos(u - v) = \cos u \cos v + \sin u \sin v$ and $\cos^2 u + \sin^2 u = 1$ are used to obtain:

$$\begin{aligned}
& \int_0^\infty \left[k_{\text{eff}}^2 \frac{\partial}{\partial z} \left(k_{\text{eff}}^{-2} \frac{\partial q}{\partial z} \right) + (k_{\text{eff}}^2 - k_r^2) q \right] \sqrt{r} \int_0^{2\pi} \exp(-ik_r r \cos(\phi - \psi)) d\phi dr \\
& = -4\pi \delta(z - z_s)
\end{aligned} \tag{3.7}$$

The integral over ϕ resembles the integral representation of the zero-order Bessel function of the first kind [54]:

$$\int_0^{2\pi} \exp(-ik_r r \cos \alpha) d\alpha = 2\pi J_0(k_r r). \tag{3.8}$$

Therefore, the Bessel function representation is substituted in for the integral over angle ϕ . With a large argument asymptotic approximation, the Bessel function

can be simplified to:

$$J_0(k_r r) = \sqrt{\frac{2}{\pi k_r r}} \cos(k_r r - \frac{1}{4}\pi). \quad (3.9)$$

Substituting Equation 3.9 into Equation 3.8, and the resulting equation into Equation 3.7 yields

$$\int_0^\infty \left[k_{\text{eff}}^2 \frac{\partial}{\partial z} \left(k_{\text{eff}}^{-2} \frac{\partial q}{\partial z} \right) + (k_{\text{eff}}^2 - k_r^2) q \right] 2\sqrt{\frac{2\pi}{k_r}} \cos(k_r r - \frac{1}{4}\pi) dr = -4\pi \delta(z - z_s). \quad (3.10)$$

If a transform is defined as

$$Q(k_r, z) = \int_0^\infty q \cos(k_r r - \frac{1}{4}\pi) dr = \int_0^\infty \frac{q}{\sqrt{2}} \cos(k_r r) + \frac{q}{\sqrt{2}} \sin(k_r r) dr, \quad (3.11)$$

the horizontal wave number domain, inhomogeneous Helmholtz equation can be written

$$k_{\text{eff}}^2 \frac{\partial}{\partial z} \left(k_{\text{eff}}^{-2} \frac{\partial Q}{\partial z} \right) + (k_{\text{eff}}^2 - k_r^2) Q = -\sqrt{2\pi k_r} \delta(z - z_s). \quad (3.12)$$

The FFP method begins with this equation, employs numerical methods to solve it, and uses an inverse transform to return to the spatial domain. The corresponding inverse transformation is [1]:

$$\frac{q}{\sqrt{2}} = \frac{1}{\pi} \int_{-\infty}^\infty Q(k_r, z) \cos(k_r r) dk_r = \frac{1}{\pi} \int_{-\infty}^\infty Q(k_r, z) \sin(k_r r) dk_r. \quad (3.13)$$

3.2.1 FFP Method Solution of the Horizontal Wave Number Domain Helmholtz Equation

In deriving Equation 3.12, assumptions of axisymmetry and range-independence were made while variation in the vertical atmospheric profile were permitted. Accordingly, the FFP method makes use of a layered representation of the atmosphere. The atmosphere consists of a stack of horizontal homogeneous layers with effective wave numbers that remain constant within each layer. An illustration of the stratified atmosphere model is shown in Figure 3.1.

In the stratified atmosphere representation, care must be taken to ensure that solutions are accurate both within each layer and between layers. This is accom-

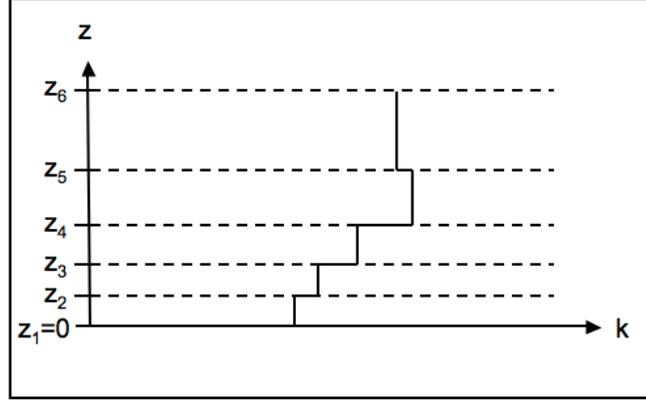


Figure 3.1. The atmosphere in the FFP method is modeled as a combination of horizontally homogeneous layers within which the wave number is constant, after [1] Figure F.1.

plished by enforcing continuity of pressure at all layer interfaces, and continuity of normal particle velocity at all layer interfaces that do not fall at the source height.

3.2.1.1 Inter-Layer Boundary Conditions

The boundary condition for continuity of pressure between layers is written as

$$p_j(z_j) = p_{j-1}(z_j) \quad (3.14)$$

where z_j is the height at the bottom of layer j and p_j is the acoustic pressure solution within layer j , as shown in Figure 3.1. After employing the substitution $p_j = q_j/\sqrt{r}$ and canceling the like-factor $1/\sqrt{r}$, the corresponding wave number domain boundary condition is

$$Q_j(z_j) = Q_{j-1}(z_j) \quad \text{for } j = 1, 2, \dots, N. \quad (3.15)$$

Assuming a time-harmonic sound field, the particle velocity in layer j normal to the layer boundary, $u_{zj}(r, z)$, is given by the linearized Euler's equation

$$u_{zj}(r, z_j) = \frac{i}{\omega \rho_{j,av}} \frac{\partial p_j(r, z_j)}{\partial z} \quad (3.16)$$

where $\rho_{j,av}$ is the average density of the atmosphere within layer j for $j =$

$1, 2, \dots, N$ and $\rho_{0,av}$ is the density of the ground material. Continuity of normal velocity, with $p_j = q_j/\sqrt{r}$ and like terms cancelled, can be written

$$\frac{1}{\rho_{j,av}} \frac{\partial q_j(r, z_j)}{\partial z} = \frac{1}{\rho_{j-1,av}} \frac{\partial q_{j-1}(r, z_j)}{\partial z}. \quad (3.17)$$

The corresponding wave number domain equation is

$$\rho_{j,av}^{-1} \frac{\partial Q_j(z_j)}{\partial z} = \rho_{j-1,av}^{-1} \frac{\partial Q_{j-1}(z_j)}{\partial z} \quad \text{for } j = 1, 2, \dots, N \quad (j \neq m). \quad (3.18)$$

The final boundary condition defines the discontinuity in the normal particle velocity at the source height. To derive this condition, Equation 3.12 is integrated over z from slightly below the source height to slightly above

$$\int_{z_s-\epsilon}^{z_s+\epsilon} \frac{\partial}{\partial z} \left(\frac{\partial Q_m}{\partial z} \right) dz + k_z^2 \int_{z_s-\epsilon}^{z_s+\epsilon} Q_m dz = -\sqrt{2\pi k_r}. \quad (3.19)$$

As ϵ goes to zero, the second term on the left-hand side goes to zero, assuming that the wave number is continuous across the source height, as would be true in a real profile. The boundary condition becomes

$$\frac{\partial Q_j(z_j)}{\partial z} = \frac{\partial Q_{j-1}(z_j)}{\partial z} - \sqrt{2\pi k_r} \quad \text{for } j = m. \quad (3.20)$$

Equations 3.15, 3.18, and 3.20 constitute the boundary conditions that define the propagation scheme. In practice, the factors $\rho_{j,av}^{-1}$ and $\rho_{j-1,av}^{-1}$ have negligible effect for the layer interfaces $j = 1, 2, \dots, N$. They are, however, significant for $j = 0$, where they account for the effect of the ground surface.

3.2.1.2 Intra-Layer Solution

Now that the rules for layer boundaries are established, the method of propagating between the reaches of each layer must be derived. If it is assumed that the effective wave number is constant within each stratum, Equation 3.12 simplifies to

$$\frac{\partial^2 Q}{\partial z^2} + k_z^2 Q = -\sqrt{2\pi k_r} \delta(z - z_s) \quad (3.21)$$

where $k_z^2 = k_{\text{eff}}^2 - k_r^2$. Therefore, the solution to Equation 3.21 is written within each stratum layer in the following form

$$Q_j(z) = A_j e^{ik_{zj}z} + B_j e^{-ik_{zj}z} \quad \text{for } z_j \leq z \leq z_{j+1} \quad (3.22)$$

where A_j and B_j are constants, and k_{zj} is the value of k_z in layer j .

The goal of the rest of this section is to determine an equation from which the solution at $z + \Delta z$ can be calculated from the solution and its derivative with respect to z , at z . The reflection coefficient is also derived for use in the next section, where the effect of ground is included.

The derivative of the solution with respect to z , from Equation 3.22, is:

$$Q'_j(z) = ik_{zj}(A_j e^{ik_{zj}z} - B_j e^{-ik_{zj}z}) \quad (3.23)$$

The corresponding equations at height $z + \Delta z$, also found from Equation 3.22, are

$$Q_j(z + \Delta z) = A_j e^{ik_{zj}z} e^{ik_{zj}\Delta z} + B_j e^{-ik_{zj}z} e^{ik_{zj}\Delta z} \quad (3.24)$$

$$Q'_j(z + \Delta z) = ik_{zj}(A_j e^{ik_{zj}z} e^{ik_{zj}\Delta z} - B_j e^{-ik_{zj}z} e^{ik_{zj}\Delta z}). \quad (3.25)$$

Within the uppermost layer, there should be no sound traveling downward, so B_{N-1} is set to 0. Also, below the ground surface, there should only be downward traveling sound, so A_0 is set to 0, yielding

$$Q_0 = B_0 e^{-ik_0 z} \quad \text{for } z \leq 0 \quad (3.26)$$

where k_0 is the complex wave number in the ground.

From the first boundary condition, Equation 3.15, evaluated at the ground surface $z_1 = 0$, the relationship between coefficients is found to be:

$$A_1 + B_1 = B_0. \quad (3.27)$$

The second boundary condition, Equation 3.18, evaluated at $z_1 = 0$ is:

$$\rho_1^{-1} \frac{\partial Q_1(z_1)}{\partial z} = \rho_0^{-1} \frac{\partial Q_0(z_1)}{\partial z}. \quad (3.28)$$

From this boundary condition, the relationship between coefficients is found to be:

$$\frac{ik_{z1}}{\rho_1} [A_1 - B_1] = -\frac{ik_{z0}}{\rho_0} B_0. \quad (3.29)$$

Substituting Equation 3.27 into Equation 3.29 gives the relationship between A_1 and B_1

$$A_1 = R(k_{z1})B_1 \quad (3.30)$$

where

$$R(k_{z1}) = \frac{\rho_0 k_{z1} - \rho_1 k_{z0}}{\rho_0 k_{z1} + \rho_1 k_{z0}}. \quad (3.31)$$

Using the local reaction approximation (Section 1.2.3.1), $k_{z0} \approx k_0$, and the normalized impedance of the ground surface is approximately equal to the normalized ground impedance, $Z_s \approx Z = \rho_0 c_0 / \rho_1 c_1$. With these substitutions, the reflection coefficient $R(k_{z1})$ is written

$$R(k_{z1}) = \frac{k_{z1} - k(z_1)/Z_s}{k_{z1} + k(z_1)/Z_s} \quad (3.32)$$

where $k(z_1)$ is the acoustic wave number in the atmosphere at the height $z_1 = 0$.

Seeking a solution at height $z + \Delta z$ that can be calculated from results at z , the solution takes the form

$$Q_j(z + \Delta z) = C_j Q_j(z) + D_j Q'_j(z) \quad (3.33)$$

or

$$\begin{aligned} & A_j e^{ik_{zj}z} e^{ik_{zj}\Delta z} + B_j e^{-ik_{zj}z} e^{ik_{zj}\Delta z} \\ &= C_j [A_j e^{ik_{zj}z} + B_j e^{-ik_{zj}z}] + D_j [ik_{zj} (A_j e^{ik_{zj}z} - B_j e^{-ik_{zj}z})]. \end{aligned} \quad (3.34)$$

Equating coefficients of like terms, C_j and D_j are found to be:

$$C_j = \cos(k_{zj}z) \quad (3.35)$$

$$D_j = k_{zj}^{-1} \sin(k_{zj}z). \quad (3.36)$$

Following a similar process to calculate $Q'_j(z + \Delta z)$, Q is found in each mid-grid

horizontal stratum as:

$$\begin{aligned} Q_j(z + \Delta z) &= \cos(k_{zj}\Delta z)Q_j(z) + k_{zj}^{-1} \sin(k_{zj}\Delta z)Q'_j(z) \\ Q'_j(z + \Delta z) &= -k_{zj} \sin(k_{zj}\Delta z)Q_j(z) + \cos(k_{zj}\Delta z)Q'_j(z). \end{aligned} \quad (3.37)$$

As was the goal, with these equations, results can be extrapolated in height from one vertical grid step to the next within layer j .

3.2.1.3 Calculating the Sound Field with the FFP

In the two preceding sections, methods for vertically extrapolating results within each layer, and maintaining accurate relationships between layers were described. This section outlines how these relationships can be used to arrive at correct results for sound levels in the propagation grid. Calculation of the sound field using the FFP proceeds with the following steps:

1. An arbitrary solution field at the ground is assumed.
2. The extrapolation methods previously discussed are applied to the arbitrary solution field at the ground, extrapolating up the grid, to find relative solutions for each layer until the source height is reached.
3. An arbitrary solution field at the top of the grid is assumed.
4. The extrapolation methods previously discussed are applied to the arbitrary solution field at the grid top, extrapolating down the grid, to find relative solutions for each layer until the source height is reached.
5. The difference in the solutions found at the source height from extrapolating up from the ground and down from the grid top are used to find the correct sound field results.

3.2.1.3.1 Assuming an Arbitrary Solution at the Ground In the first step, the value of the coefficient representing sound traveling in the downward direction B_1 is arbitrarily set to 1, yielding, from Equations 3.22 and 3.30:

$$Q_1(z_1) = R(k_{z1}) + 1 \quad (3.38)$$

$$Q'_1(z_1) = ik_{z1}[R(k_{z1}) - 1]. \quad (3.39)$$

These definitions are equivalent to dividing the full solutions for Q_1 , Equation 3.22, and Q'_1 , Equation 3.23, by B_1 for $j = 1$ and $z = 0$:

$$\frac{Q_1}{B_1} = \frac{A_1}{B_1} + 1 = R(k_{z1}) + 1 \quad (3.40)$$

$$\frac{Q'_1}{B_1} = ik_{z1}\left[\frac{A_1}{B_1} - 1\right] = ik_{z1}[R(k_{z1}) - 1]. \quad (3.41)$$

The ratio of these equations for arbitrarily set coefficients is equivalent to the ratio of the general equations:

$$\frac{Q'_1/B_1}{Q_1/B_1} = \frac{Q'_1}{Q_1}. \quad (3.42)$$

However, the values are not correct for Q_1 or Q'_1 alone.

3.2.1.3.2 Extrapolating Up From the Ground The second step is completed using Equations 3.37 with $z = z_j$ and $\Delta z = z_{j+1} - z_j$ and the inter-layer boundary conditions, Equations 3.15 and 3.18, to extrapolate from the arbitrary field defined at the ground. The solution is determined from successive calculations for $j = 2, 3, \dots, m$, where $z_m = z_s$ is the source height. Once the source height is reached in the layer directly below the source, the values of $Q_{m-1}(z_m)$ and $Q'_{m-1}(z_m)$ are stored as Q_{ml} and Q'_{ml} , designating that the values came from the lower region of the grid.

3.2.1.3.3 Assuming an Arbitrary Solution at the Top of the Grid Similar to Step 1, in the Step 3, the solution is set to $Q_{N-1}(z_N) = 1$ at the top of the grid. Because it is assumed that no sound is traveling downward and, thus, $B_{N-1} = 0$, the equations for the top of the grid become

$$Q_{N-1}(z_N) = 1 \quad (3.43)$$

$$Q'_{N-1}(z_N) = ik_{zN}. \quad (3.44)$$

Once again, the ratio of these two equations is correct, but neither is correct alone.

3.2.1.3.4 Extrapolating Down From the Top of the Ground To perform the fourth step, Equations 3.37 are used with $z = z_{j+1}$ and $\Delta z = z_j - z_{j+1}$ and the inter-layer boundary conditions, Equations 3.15 and 3.18 are applied, to extrapolate from the arbitrary field defined at the grid top. The solution is determined from successive calculations for $j = N - 1, N - 2, \dots, m$. Once the source height z_m is reached in the layer directly above the source, the values of $Q_m(z_m)$ and $Q'_m(z_m)$ are stored as Q_{mu} and Q'_{mu} , designating that the values come from the upper region of the grid.

3.2.1.3.5 Calculating Correct Sound Field Results Finally, in the fifth step, the correct numerical values of Q_j and Q'_j are determined by using the difference between the ratios Q'_{mu}/Q_{mu} and Q'_{ml}/Q_{ml} calculated just above and below the source height. The boundary condition at the source height defined by Equation 3.20 specifies the relationship that should exist between the derivative of the solution calculated in the layers directly below and above the source. This boundary condition can be written

$$\left(\frac{Q'_{mu}}{Q_{mu}}\right)Q_m - \left(\frac{Q'_{ml}}{Q_{ml}}\right)Q_m = -\sqrt{2\pi k_r} \quad (3.45)$$

where Q_m is the correct solution at the source height z_m . Rearranging the equation, Q_m is found in terms of calculated quantities:

$$Q_m = \frac{-\sqrt{2\pi k_r}}{(Q'_{mu}/Q_{mu}) - (Q'_{ml}/Q_{ml})}. \quad (3.46)$$

Thus, to find the correct values of the solution in the lower and upper grid regions, the extrapolated relative solutions are multiplied by the ratio of the correct result to the calculated result at the source height, from the lower and upper regions:

$$\begin{aligned} (Q_m/Q_{mu}) & \text{ for } z_j > z_m \\ (Q_m/Q_{ml}) & \text{ for } z_j < z_m. \end{aligned} \quad (3.47)$$

The value of q is found by performing an inverse Fourier transformation of the

new, correct Q in each layer of atmosphere, from 3.13 [1]:

$$q(r, z) = \frac{1}{\pi\sqrt{2}} \int_{-\infty}^{\infty} [e^{ik_r r} + e^{-ik_r r}] Q(k_r, z) dk_r. \quad (3.48)$$

Thus, correct solutions are determined by transforming the wave number domain solutions back into the spatial domain.

3.2.2 Numerical Considerations of the Inverse Transform Evaluation

Certain numerical aspects of the inverse transform integral, Equation 3.48, must be taken into consideration to represent the integral accurately.

First, the integrand in Equation 3.48 contains poles near $k_r = \pm k$. So that these regions will not dominate the integral, a small imaginary term ik_t is added to or subtracted from the integration path for negative and positive values of the real part of k_r , respectively. This serves to deform the path and avoid the poles, as shown in Figure 3.2. The added factor $\exp(-k_t r)$ can be taken out of the integral so that Fourier techniques may still be used to numerically solve the integration.

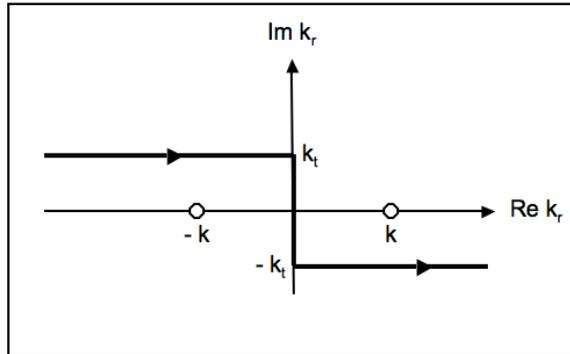


Figure 3.2. Integration path for the inverse transform integral that avoids poles at $k_r = \pm k$, after [1] Figure F.2.

To simplify the integral further, it is recognized from Equation 3.12 that $Q(-k_r, z) = \pm iQ(k_r, z)$. With this substitution, the limits of the integral can be adjusted from $[-\infty, \infty]$ to $[0, \infty]$:

$$q(r, z) = \frac{1 - i}{\pi\sqrt{2}} \int_0^{\infty} [\exp(ik_r r) + \exp(-ik_r r)] Q(k_r, z) dk_r. \quad (3.49)$$

To allow for numerical implementation, the horizontal wave number is discretized, including the small imaginary term as:

$$k_{r,n} = k_{s,n} - ik_t \quad (n = 1, 2, \dots, M) \quad (3.50)$$

where

$$k_{s,n} = \frac{1}{2}\Delta k, \frac{3}{2}\Delta k, \dots, k_{s,M}. \quad (3.51)$$

Assignment of the wave number spacing Δk and the largest wave number $k_{s,M}$ require examination of the desired geometry of the results grid and consideration of the frequency of sound. The wave number spacing Δk affects the range over which the result field will be periodic. Thus, the spacing must be chosen small enough that the periodicity does not have a significant effect within the range to which results are sought. The guideline of $2\pi/\Delta k \geq 3r$ can be applied, where r is the range variable. In addition, the largest wave number used in the Fourier transform depends on the frequency of sound and should be chosen much larger than the location of the pole at $k_r = k = \omega/c_{\text{eff}}$. For this, the guideline $k_{s,M} \approx 3\omega/c_0$ can be applied, where c_0 is the speed of sound at the ground. Finally, for the small imaginary component ik_t subtracted from the integration variable, a value of $i\Delta k$ can be used.

Another concern in truncating the range of wave numbers in the integration is the introduction of small, rapid oscillations in q caused by a discontinuity between the beginning and ending values of the wave number spectrum, which leads to a discontinuity in the assumed periodic spectrum. A window function that consists of ones through the middle of the spectrum and goes smoothly to zero at both the lower and upper integration limits can be applied as a factor to the integrand in Equation 3.49 to avoid these oscillations.

A final method of increasing numerical accuracy addresses errors that result from including a large number of vertical layers in the FFP calculations. In such conditions, when Equations 3.37 are applied numerous times, the imaginary component of k_{zj} may cause the values of Q_j to grow large with increasing j in the region below the source (from the factor $\exp(-ik_{zj}\Delta z)$) and with decreasing j in the region above the source (from the factor $\exp(+ik_{zj}\Delta z)$).

To avoid the exponential growth, the quantities Q_j and Q'_j can be multiplied

by the factors

$$\begin{aligned} \exp(ik_{z0}\Delta z) & \quad \text{for } \Delta z > 0 \\ \exp(-ik_{z0}\Delta z) & \quad \text{for } \Delta z < 0 \end{aligned} \tag{3.52}$$

after each step from height z to $z + \Delta z$, where k_{z0} can be set as the vertical wave number at the ground. Applying the factor to both Q_j and Q'_j does not change the ratio between the two. Thus, no change to the extrapolation procedure is necessary. However, a correction factor of $\exp(ik_{z0}|z - z_s|)$ must be applied when calculating the correct values of Q_j using the ratios factors in Equation 3.47.

3.3 Numerical Implementation

In the remaining sections, numerical considerations and implementations in this research are discussed. A number of the implementation techniques are common to both the parabolic equation and the fast field program, for example ground impedance calculation and refractive atmosphere definition. Where implementation is the same, the reader is referred to the corresponding descriptions in Chapter 2.

3.3.1 Grid Parameters

While the derivation of the FFP method includes the use of integral transforms, in practice, numerical implementation uses discrete points and finite limits. Consequently, as specified in Section 3.2.2, certain guidelines are applied to the integration variable k_r used in the inverse transform of the FFP, Equation 3.49, to avoid numerical inaccuracies.

As suggested by Salomons, the guideline of $2\pi/\Delta k = 3r$ is followed to avoid the influence of the periodicity of the result field. Therefore, to ensure that results at all ranges are calculated accurately, the wave number spacing Δk is set to

$$\Delta k = \frac{2\pi}{3r_f} \tag{3.53}$$

where r_f is the range to which the user desires accurate results. By sampling theory, the relationship $\Delta k \Delta r = 2\pi/N$ must be observed and the range used in

the FFP calculations becomes

$$N\Delta r = r_N = 3r_f. \quad (3.54)$$

Thus, the range to which calculations are made is much larger (by a factor of three) than the range for which the results are used [65].

This research also complies with the other guidelines suggested by Salomons regarding discretization of the integration variable k_r : Δk is used for the value of the small imaginary term k_t introduced to the wave number to avoid poles in the integrand and $k_{s,M} = 3\omega/c_0$ is used for the maximum wave number included in the transform spectrum.

3.3.1.1 Grid Point Spacing

For ease of combining PE and FFP results in the hybrid model, the spacing of grid points in the FFP follows similar procedures as the PE (Section 2.3.1), with one exception:

The FFP does not require the small resolution of range steps necessary in the PE. Therefore, so long as the range step resolution the user desires is not exceeded, the range step of the FFP is set to twice that of the PE. With this spacing, the horizontal grid spacing may be as large as $\lambda/5$, increasing computational efficiency.

The PE does not use the stratified atmosphere representation. However, the spacing of the vertical layers of the atmosphere used in the FFP is set equal to the vertical grid point spacing of the PE in order to maintain common vertical grid point spacing between models.

Finally, because it is required that the source height used in the FFP lie between layers, the source height is rounded to the height of the nearest layer boundary. While this likely changes the source height slightly, the difference between the real and implemented height will be no more than $\lambda/20$.

3.3.2 Atmospheric Absorption, Refractive Atmosphere, and Ground Impedance

A number of propagation mechanisms including atmospheric absorption, refractive atmosphere, and ground impedance, are implemented in a similar manner in the FFP as in the PE. Atmospheric absorption is included with the same imaginary component added to the acoustic wave number (see Section 2.3.3). The effective sound speed profile definition of the PE model implementation using specified profiles (see Section 2.3.4) is also used in the FFP.

The method of including the effect of the ground with the reflection coefficient $R(k_{z1})$ was described in Section 3.2.1.2. The normalized ground impedance is used as an input into the reflection coefficient equation, Equation 3.32. Just as in the PE (see Section 2.3.6), the user may choose to manually assign the complex, normalized impedance. However, if the user chooses to let the model calculate the impedance from the flow resistivity parameter, Equation 1.1 will be used.

Turbulence is not included in this implementation of the FFP method. However, the effects of turbulent scattering have been included in the FFP in other work by accounting for the amplitude and phase decorrelation introduced between direct and reflected sound [66].

The FFP cannot accommodate range-dependent effects. Therefore, neither discontinuities in ground impedance, nor terrain features can be included.

3.4 Conclusions

In this chapter, the fast field program method formulation included in the hybrid propagation model was derived and the details of the model's implementation were presented. Although the FFP is unable to incorporate range-dependent effects, it is a good complement to the PE model, providing accurate results at large elevation angles where the PE's accuracy is limited. Thus, the FFP is used to support the PE in the hybrid model. The two models combined, as described in the following chapter, achieve an accurate sound field at all points in a vertical two-dimensional grid.

Hybrid PE-FFP Model

4.1 Introduction

The hybrid propagation model developed in this research is a composite model of two different propagation methods: the parabolic equation method and the fast field program. Utilizing the strengths of each model, the full two-dimensional vertical sound field produced for a stationary point source is represented accurately at each grid point. The two hybrid model components were described in detail in the previous chapters. The method for combining them, and the performance of the resulting hybrid model, is detailed in this chapter.

4.2 Procedure for Model Combination

As discussed in Chapter 2, the wide-angle formulation of the parabolic equation method used in this research is valid for elevation angles of up to approximately 35° . The hybrid model is, thus, formed by running both PE and FFP models for the desired results grid, retaining the PE results for elevation angles within 35° and using FFP results at all other grid points. Although only PE results at low elevation angles are used in the hybrid model, these results are calculated purely with the PE, extrapolated in range from a starting field at the first range step. Similarly, FFP results are calculated purely with the FFP method. Transitions between models is implemented only after all propagation calculations are complete

and, thus, the PE and FFP results are entirely uncoupled. Figure 4.1 shows an illustration of the hybrid model formation process. In the upper left figure, the FFP results are calculated and the values at grid points inside of 35° are set to 0. In the upper right figure, the PE results are calculated and the values at grid points outside of 35° are set to 0. The FFP and PE results grids are then added together in the lower left figure to produce a results grid accurate at all points. The lower right figure shows a slice of the sound field from the figure on the lower left, at a height of 2 m. The values of the curve to the left of the black arrow were calculated with the FFP. The values to the right were calculated with the PE. Their rather seamless union confirms the validity of combining the two models.

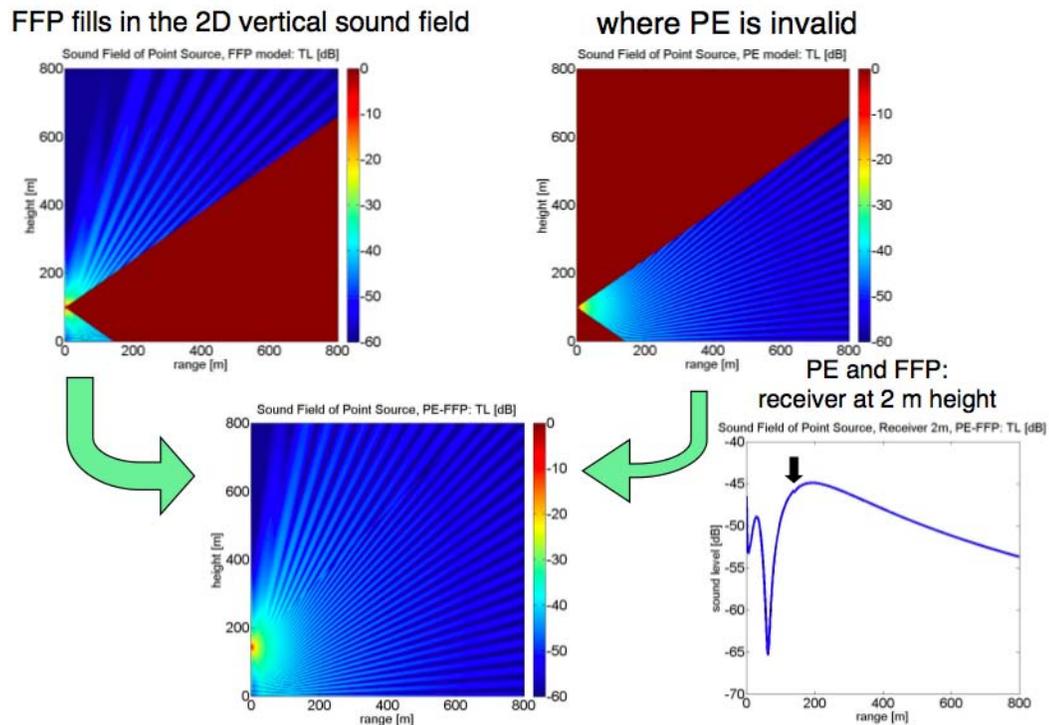


Figure 4.1. Method of combining the PE and FFP methods into the hybrid model. Upper Left Figure: contribution of the FFP to the vertical two-dimensional sound field, retaining data at elevation angles greater than 35° from the source. Upper Right Figure: contribution of the PE, retaining data at elevation angles of 35° and smaller. Bottom Left Figure: PE-FFP union. Bottom Right Figure: PE-FFP union at a height of 2 m over the ground. The arrow marks the transition from FFP to PE data.

The process of combining the PE and FFP models is used through the remainder of this chapter and is the foundation of the hybrid model. Substantiation of

the model is achieved by verifying both individual FFP and PE components and validating their combination into the hybrid model.

4.3 Validation and Verification

The PE and FFP methods have been extensively used in the field of sound propagation. There are, therefore, established benchmark results available for verification of the models. Comparisons were made between model-produced and published results for the both the PE and FFP. To validate that the models were producing accurate results, model results collected under varied test cases were evaluated, to confirm the diverse capabilities of the model were working correctly.

4.3.1 Verification: Individual Model Comparisons with Published and Accepted Results

To verify both the PE and FFP models, results have been compared to three sets of computations: (1) the published results of Salomons in “Computational Atmospheric Acoustics” [1], (2) an analytical model that adds the contributions of direct and reflected sound and uses a homogeneous atmosphere, and (3) the published results of Attenborough, et al. “Benchmark cases for outdoor sound propagation models” [7]. The models’ agreement with these results, spanning a large range of frequencies, ground impedances, and atmospheric conditions suggest that the models have been coded correctly. Comparisons with [7] for different sound speed profiles and a comparison with [1] for a condition of turbulence are included in Appendix F.

4.3.2 Validation: Near Ground Operation Test Cases

In order to explore the performance capabilities of the hybrid PE-FFP propagation model, results for five test cases were calculated. The test cases were designed by the U.S. Department of Transportation Volpe National Transportation Systems Center and the Federal Aviation Administration (FAA) to be simple enough for analytical study, and yet representative of certain aspects of real world applications. Penn State, in conjunction with Volpe and the FAA, then picked specific

quantitative parameters to apply to the general test case designs. The test cases serve as reference benchmarks for comparison between the hybrid model and other analytical and numerical models.

4.3.2.1 Description of Test Cases

The five test cases explore different combinations of terrain and ground impedance features, with increasing complexity. The first case uses a flat, soft ground geometry; the second, a flat, hard ground, with a patch of soft ground in the middle; the third, a soft ground with an upward slope terrain feature; the fourth, a soft ground with a downward slope terrain feature; and the fifth, a hard ground that transitions to soft and includes a hill feature.

All five test cases were run for 50, 500, and 2500 ft source heights. The 50 and 500 ft sources were run for 1/3-octave bands from 50-1000 Hz and octave bands from 2000-8000 Hz in a homogeneous atmosphere. Calculations were performed for a large frequency range because aviation noise is not well represented by single tones, but rather is broadband in nature. The 2500 ft source was run at 100, 800, and 4000 Hz in a homogeneous atmosphere. A 50 ft source was run at 100 and 800 Hz in a downward refracting atmosphere. The two lower altitude sources in a homogeneous atmosphere were run for the full range of frequencies with the aim of achieving meaningful broadband results to use in validation with other models. The high altitude source in a homogeneous atmosphere was run at low, middle, and high frequencies, and the two low altitude sources in a refractive atmosphere were run at low and middle frequencies to get a sense of performance in more extreme and complex conditions while maintaining moderate runtimes. Other parameters used in the test cases are as follows:

- Omni-directional point sources, 94 dB re 1 m
- Receivers: 4 ft height
- Soft ground: 150 cgs rayls flow resistivity, hard ground: 20,000 cgs rayls flow resistivity

The test case geometries for the 50 and 500 ft sources are illustrated in Table 4.1. The tables and figures have been grouped together for ease of comparison.

The ranges at which transitions in impedance and terrain occur are increased by 2,953 ft (900 m) for the 2500 ft source height condition. Otherwise, these test case geometries for the 2500 ft source are equivalent to those of the 50 and 500 ft sources.

4.3.2.2 Numerical Implementation of the Test Cases in the Hybrid Model

General numerical implementation of propagation effects was described in Sections 2.3 and 3.3, for the PE and FFP, respectively. This section describes the specific numerical implementation of the test cases, referencing Sections 2.3 and 3.3 when needed.

4.3.2.2.1 PE Start Field To avoid numerical errors resulting when the exponential integral form is used at high frequencies, the series representation of the spherical-wave reflection coefficient, introduced in Section 2.3.2.3 is utilized at and above 500 Hz, with two summation terms. The exponential integral and series expansion methods were found to be nearly equivalent at higher frequencies.

4.3.2.2.2 Grid Step Spacing Grid spacings that both meet the minimum PE spacing criteria and have data for every frequency specified at common grid points, as discussed in Sections 2.3.1 and 3.3.1.1, were utilized in the test cases. However, larger maximum step sizes were used for smaller frequencies in cases with sloping ground to avoid numerical errors. The maximum horizontal grid step size for the various frequency ranges and test cases was specified as shown in Table 4.2. If this step size is less than $\frac{\lambda}{10}$, this step size is used. If this spacing is greater than $\frac{\lambda}{10}$, the maximum step size in Table 4.2 is divided by the smallest integer divisor that yields a step size less than $\frac{\lambda}{10}$.

4.3.2.2.3 Atmospheric Absorption Utilizing the option to manually assign the atmospheric absorption coefficients for each frequency, discussed in Section 2.3.3, atmospheric absorption is included using the standard SAE 866A [9] atmosphere absorption coefficients. These are the coefficients used in INM/AEDT.

4.3.2.2.4 Refractive Atmosphere For the cases run with a downward refracting atmosphere, a logarithmic effective sound speed profile, defined with Equation 1.7, is used. The speed of sound at the ground c_0 is set to 343 m/s, the parameter of logarithmic sound speed b is set to 1 m/s, and the roughness length z_0 is set to 0.1 m.

4.3.2.2.5 Terrain To avoid discontinuities in the slope of the terrain, a sinusoid-taper smoothing algorithm was applied for 50 meters after the start and 50 meters before the end of the terrain slopes. The smoothed slopes can be seen for the Test Case 5 example in Figure 4.2. The resulting change to the terrain is small.

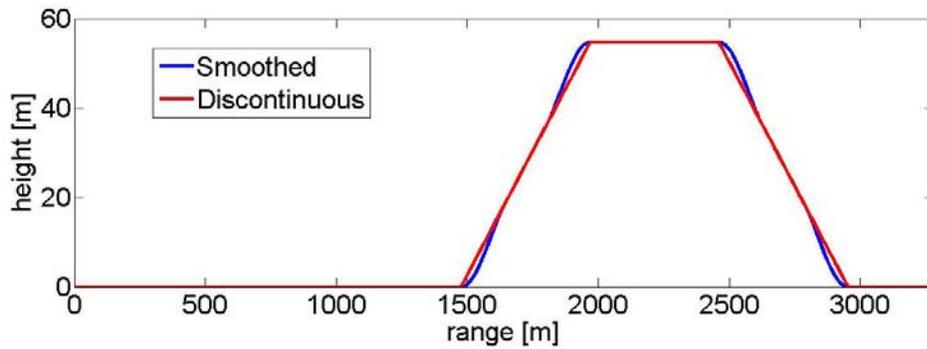


Figure 4.2. Smoothed terrain geometry plotted against discontinuous-slope terrain, Test Case 5.

4.3.2.2.6 Propagation Range The range at which propagation calculation was terminated was determined by the Detectability Level, as defined in the INM 7.0 Technical Manual [4], using a 94 dB re 1 m source. As their geometries were specified in English units, test case result distances are presented in feet. A minimum propagation distance was set to 3281 ft (1 km) for source heights of 50 and 500 ft, and 6562 ft (2 km) for the source height of 2500 ft (where the ranges at which transitions in impedance and terrain occur are larger by 2,953 ft (900 m)). The maximum allowable propagation range was 75,130 ft (22.9 km).

4.3.2.3 Test Case Results

Results generated by the hybrid model for the five test cases are presented in Figures 4.3 through 4.8. Figures 4.3 through 4.7 show results for low, middle and high frequencies: 100, 800, and 4000 Hz, and low, middle, and high source heights: 50, 500, and 2500 feet, in a homogeneous atmosphere. A dashed curve representing spherical spreading and atmospheric absorption is shown for comparison. The figures also include averaged results for frequencies of 50-8000 Hz with source heights of 50 and 500 ft. Figure 4.8 shows results for low and middle frequencies: 100 and 800 Hz, and a low source height: 50 ft, in a downward refracting atmosphere. A curve representing results for a homogeneous atmosphere is included, for comparison. The geometry for each test case is shown below the figures. Distances in the geometry illustrations correspond to those in the results figures.

The test case results conform well to expectations for the different geometries. Test Case 1 results (Figure 4.3) show standard interference patterns with interference dip locations varying for different frequencies. Discontinuities in levels can be seen for the higher source heights at the transition from the FFP model calculations to the PE model calculations but are mostly small, especially for broadband calculations. (Transitions occur at 72, 712, and 3570 ft for 50, 500, and 2500 ft source heights, respectively.) Averaged broadband results for frequencies 50-8000 Hz were determined by averaging results from runs at each 1/3-octave band center frequency, weighted by the bandwidth of the 1/3-octave band. The averaged results follow the spherical spreading and atmospheric absorption curve closely for both 50 and 500 ft source heights. (Averaged results were not computed for the 2500 ft source because only 3 frequencies were run.)

In Test Case 2 (Figure 4.4) levels trend slightly higher than in Test Case 1 for the first 984 ft (300 m) of propagation, where the ground is now hard, rather than soft. Results show minimal effect of the impedance change for lower frequencies and a more significant effect for higher frequencies. In the soft segment, the level drops slightly, as compared to the hard segment, and interference dips are less severe. The averaged results for frequencies 50-8000 Hz again follow the spherical spreading and atmospheric absorption curve closely. However, a slight effect of the impedance changes can be seen for the 50 ft source, while the effect is negligible for the 500 ft source.

Test Case 3 results (Figure 4.5) show the shadowing effect of an upward slope. The shadow zone begins after the transition from sloping ground to flat ground. The effects of the upward slope on the sound levels on the higher flat ground are smaller for low frequency sources, for which diffraction causes higher levels in the shadow zone. The effects are significant for higher frequency sources at lower altitudes, for which diffraction into the shadow zone is not as substantial. The effects of the upslope are small for high altitude sources where a more direct path exists from source to receiver. Interference-pattern behavior can also be seen over the sloping ground for most Test Case 3 runs. However, because of the one-way limitations of the PE, sound cannot be reflected backward from the upward slope. Therefore, sound levels before the slope are the same as in Test Case 1. The averaged results for frequencies 50-8000 Hz show a significant effect of the shadow zone for the 50 ft high source, but follow the spherical spreading and atmospheric absorption curve closely for the 500 ft source.

Test Case 4 results (Figure 4.6) show the shadowing effect of a downward slope. Here, the shadow zone begins after the transition from the flat to the sloping ground. The effects of the downward slope—a sharp decrease in sound level over the sloping ground, followed by a moderate increase over the lower flat ground, due to diffraction—are significant for both high and low frequency sound from low sources, though they remain larger for higher frequencies. Effects are, again, smaller for high altitude sources where a more direct path exists from source to receiver. Unlike in Test Case 3, the interference-pattern behavior is not as prominent over the sloping ground because there are no direct reflections off the ground. The small notches in level that are seen after the beginning and before the end of the sloping ground are results of the terrain smoothing. To make a more gradual transition between the flat and slanted ground segments, the slope of the terrain is increased in part of the smoothed region (see Figure 4.2). This temporarily increases the severity of the shadow zone, and decreases sound level. The averaged results for frequencies 50-8000 Hz again show a significant effect of the shadow zone for the 50 ft high source and follow the spherical spreading and atmospheric absorption curve closely for the 500 ft source (though the effect of the terrain smoothing can still be seen).

Test Case 5 results (Figure 4.7) show the effect of both a hill and an impedance

change from hard to soft ground. Here, again, the effect of the impedance change is minimal for low frequencies, but significant for higher frequencies. The shadowing effect of the upward slope on the flat top of the hill is small for low frequencies but significant for high frequencies, with the low altitude (50 ft) source. The shadowing effect on the downward slope of the hill is present for both high and low frequency sound. However, in this case, the direct path behind the hill is blocked even for the 500 ft source height, so the shadow zone is present for both 50 and 500 ft sources and a sharp decrease in level is seen. The 2500 ft source is high enough that a direct path exists from the source to the receivers behind the hill and the extreme decrease in level is not observed. In the averaged results for frequencies 50-8000 Hz and source height of 50 ft, the figure shows a slight effect of the impedance change at the transition from hard to soft ground, interference-pattern behavior over the upward slope, a small shadow zone over the flat top of the hill, a large shadow zone over the downward slope, and diffraction over the flat ground on the other side of the hill. In the averaged results for frequencies 50-8000 Hz and source height of 500 ft, the figure shows close adherence to the spherical spreading and atmospheric absorption curve until the downward slope, over which there is a shadow zone, and diffraction over the flat ground on the other side of the hill. The most significant deviation from spherical spreading and atmospheric absorption occurs where a shadow zone is present.

Results for the five test cases run with a downward refracting atmosphere, as described above, for 100 and 800 Hz sources at 50 ft height, are shown in Figure 4.8. Results obtained using a homogeneous atmosphere are included for comparison. For all test cases, results using the refractive atmosphere follow those of the homogeneous atmosphere at small ranges. The levels deviate as the range increases, and a trend toward slightly higher levels can be seen in Test Cases 1 and 2. However, the effect of the refractive atmosphere shows the largest significance in cases where a shadow zone forms. In the shadow zones, the level is raised appreciably. The effect is larger for the 800 Hz source, for which increases reach more than 20 dB.

The sensitivity of the results to the precise shape of terrain profile was investigated further by examining the effect of the terrain smoothing implemented in the test cases. Terrain smoothing was initially used because of indications in the

literature that changes in terrain slope should not be discontinuous. However, in our test case runs, the model appears to handle discontinuous changes in slope without issue. Therefore, additional hybrid model runs were performed for three different conditions of Test Case 4, where notches in level, evident after the beginning and before the end of the sloping ground, are attributed to the effects of terrain smoothing.

Figure 4.9 shows results for Test Case 4 for a 100 Hz source at heights of 50 and 500 ft, and for a 800 Hz source at a height of 50 ft. These conditions showed the clearest appearance of the notches in sound level. In this figure, comparisons are made between results with and without terrain smoothing. In all conditions investigated, the notches disappear when the terrain is not smoothed and the results over a portion of the middle of the downward slope agree for the two treatments of terrain. For the 100 Hz, 50 ft high source, the results at ranges beyond the downward slope agree well, with slightly larger levels shown for the terrain that has not been smoothed. Results for the 100 Hz, 500 ft high source condition at ranges beyond the downward slope show some small differences between the peaks and dips in level, as well as some difference in the pattern at the larger ranges. The condition of the 800 Hz, 50 ft high source shows larger differences between the two terrain treatments for ranges beyond the downward slope. In this condition, levels calculated for the unsmoothed terrain significantly exceed the levels calculated for the smoothed terrain. These preliminary results indicate that results beyond a terrain feature may be sensitive to the terrain shape that is used. However, these difference have not been exhaustively investigated. Therefore, if the user is particularly interested in a region either near transitions in the slopes of terrain features, or even within the region beyond the terrain feature, the user should exercise caution and avoid inattentively applying the model. Furthermore, care should be taken to accurately define the terrain profile because a greatly approximated combination of straight line segments may not be an adequate representation of the real profile.

All test cases were run using Matlab on a 64-bit computer workstation with an Intel Xeon Quad Core processor. Runtimes were strongly dependent on frequency and vertical grid size. For the 50 and 500 ft source heights, low frequency runs (up to the low to mid hundred Hz) took on the order of seconds and minutes,

mid frequency runs (upper hundred Hz to lower thousand) took on the order of hours, and high frequency runs (upper thousand Hz) took on the order of tens of hours. To run one test case for a source height of 50 or 500 ft at all frequencies included in this report would take approximately 24 to 34 hours. To run one test case for a source height of 2500 ft for frequencies 100, 800, and 4000 Hz would take approximately 36 to 40 hours.

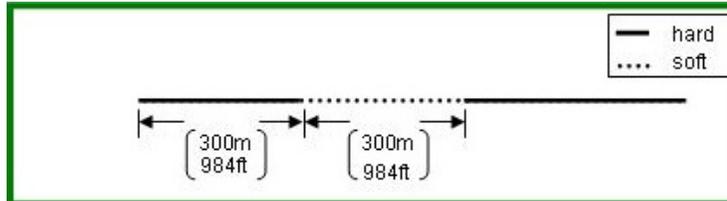
The model, as a prototype, was programmed in Matlab for ease of development. However, Matlab is not a compiled language and suffers from long runtimes. The model is, therefore, being reprogrammed in C#.net by the Volpe Center to achieve gains in computational speed. Additionally, because runs at each frequency are independent, the recoded model could make use of parallel processing for huge computational gains.

Table 4.1. Description of test case geometries for the 50 and 500 ft source heights.

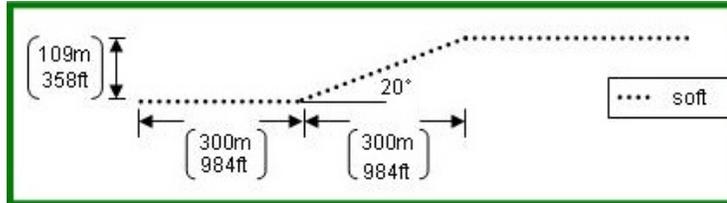
Test Case 1: Flat, Soft Ground



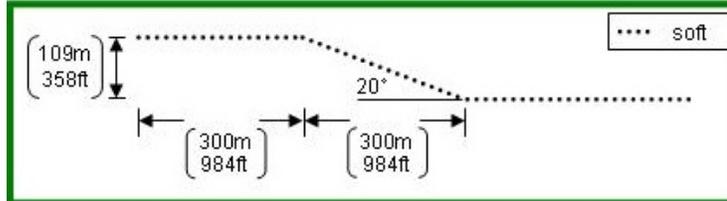
Test Case 2: Flat, Hard-Soft-Hard Ground



Test Case 3: Upward Slope, Soft Ground



Test Case 4: Downward Slope, Soft Ground



Test Case 5: Hill, Hard-Soft, Ground

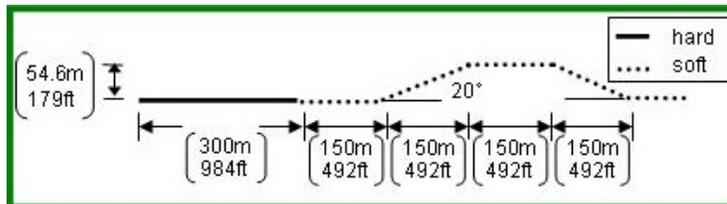


Table 4.2. Maximum horizontal grid step spacing, by case number and source frequency.

Case No.	Frequency [Hz]		
	10-20	25-40	50+
1-2	0.5 m	0.5 m	0.5 m
3-5	2.0 m	1.0 m	0.5 m

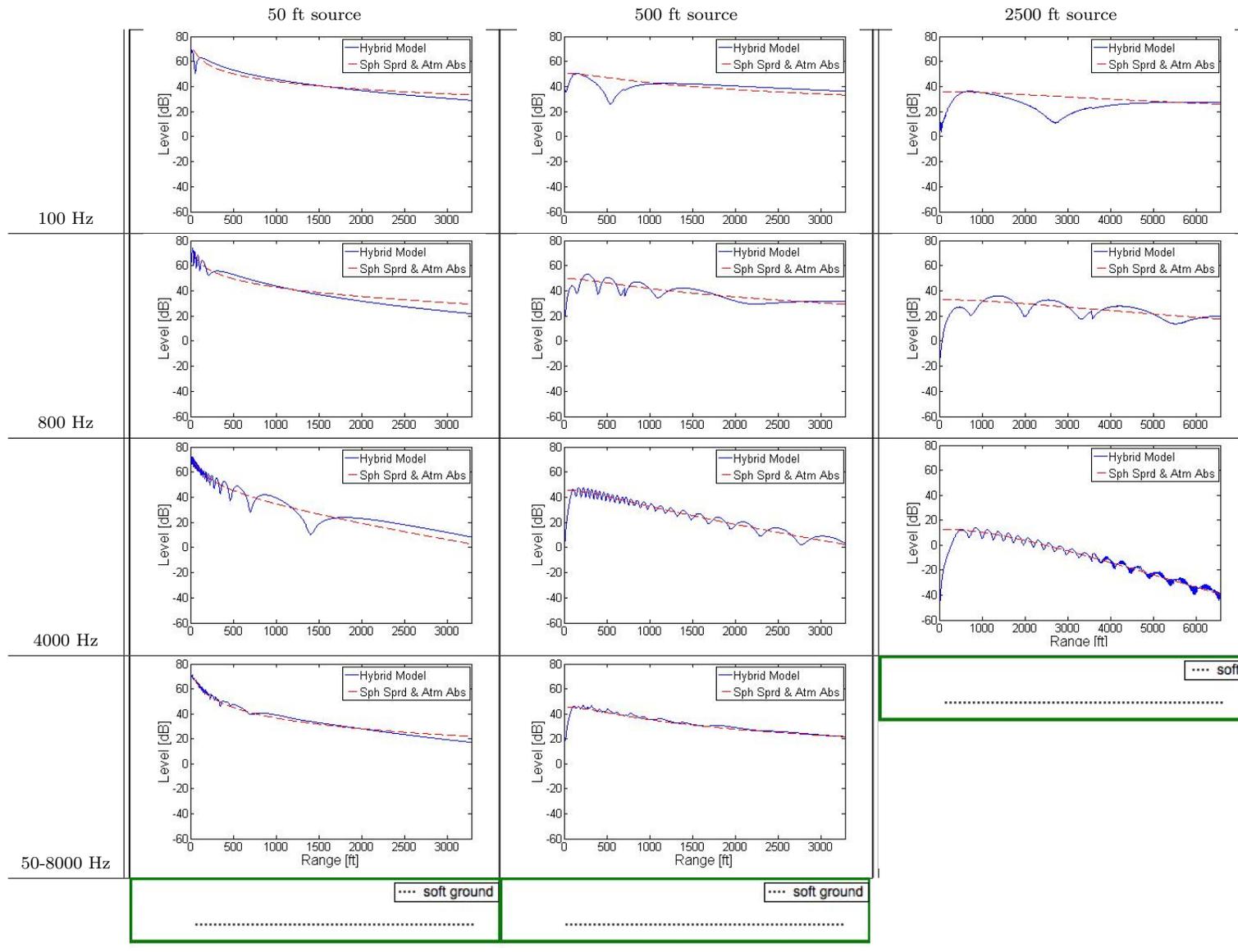


Figure 4.3. Results for Test Case 1, flat, soft ground. The first, second, and third columns show results for 50, 500, and 2500 ft source heights, respectively. The first, second, and third row show results for 100, 800, and 4000 Hz source, respectively. The fourth row shows averaged results for frequencies 50-1000 Hz in 1/3-octave bands and 2000-8000 Hz in octave bands for the 50 and 500 ft source heights. Geometries of the test cases are illustrated below the result figures. Note the different range used for the 2500 ft source.

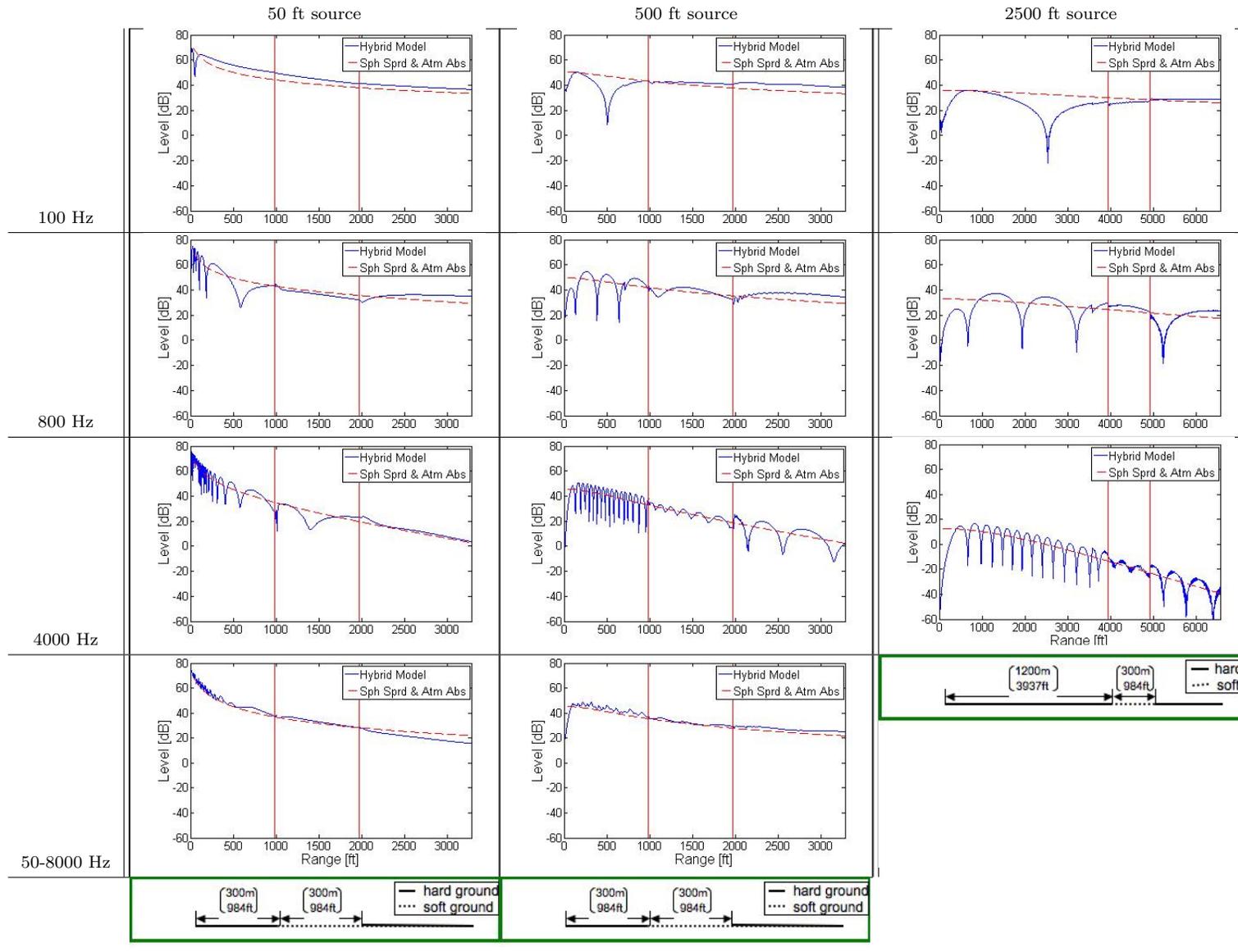


Figure 4.4. Results for Test Case 2, flat, hard-soft-hard ground. The first, second, and third columns show results for 50, 500, and 2500 ft source heights, respectively. The first, second, and third row show results for 100, 800, and 4000 Hz source, respectively. The fourth row shows averaged results for frequencies 50-1000 Hz in 1/3-octave bands and 2000-8000 Hz in octave bands for the 50 and 500 ft source heights. Geometries of the test cases are illustrated below the result figures. Note the different range used for the 2500 ft source.

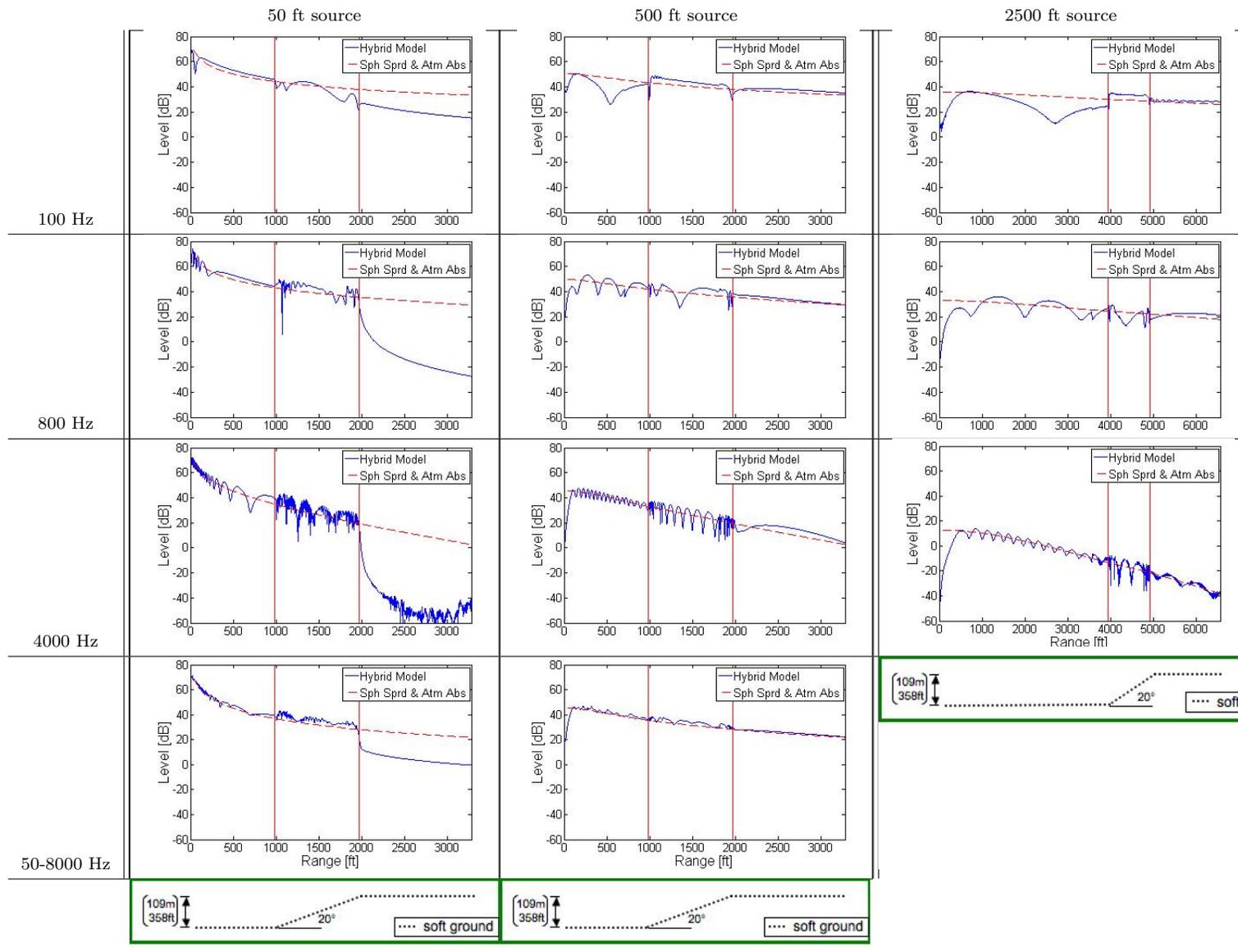


Figure 4.5. Results for Test Case 3, upward sloping, soft ground. The first, second, and third columns show results for 50, 500, and 2500 ft source heights, respectively. The first, second, and third row show results for 100, 800, and 4000 Hz source, respectively. The fourth row shows averaged results for frequencies 50-1000 Hz in 1/3-octave bands and 2000-8000 Hz in octave bands for the 50 and 500 ft source heights. Geometries of the test cases are illustrated below the result figures. Note the different range used for the 2500 ft source.

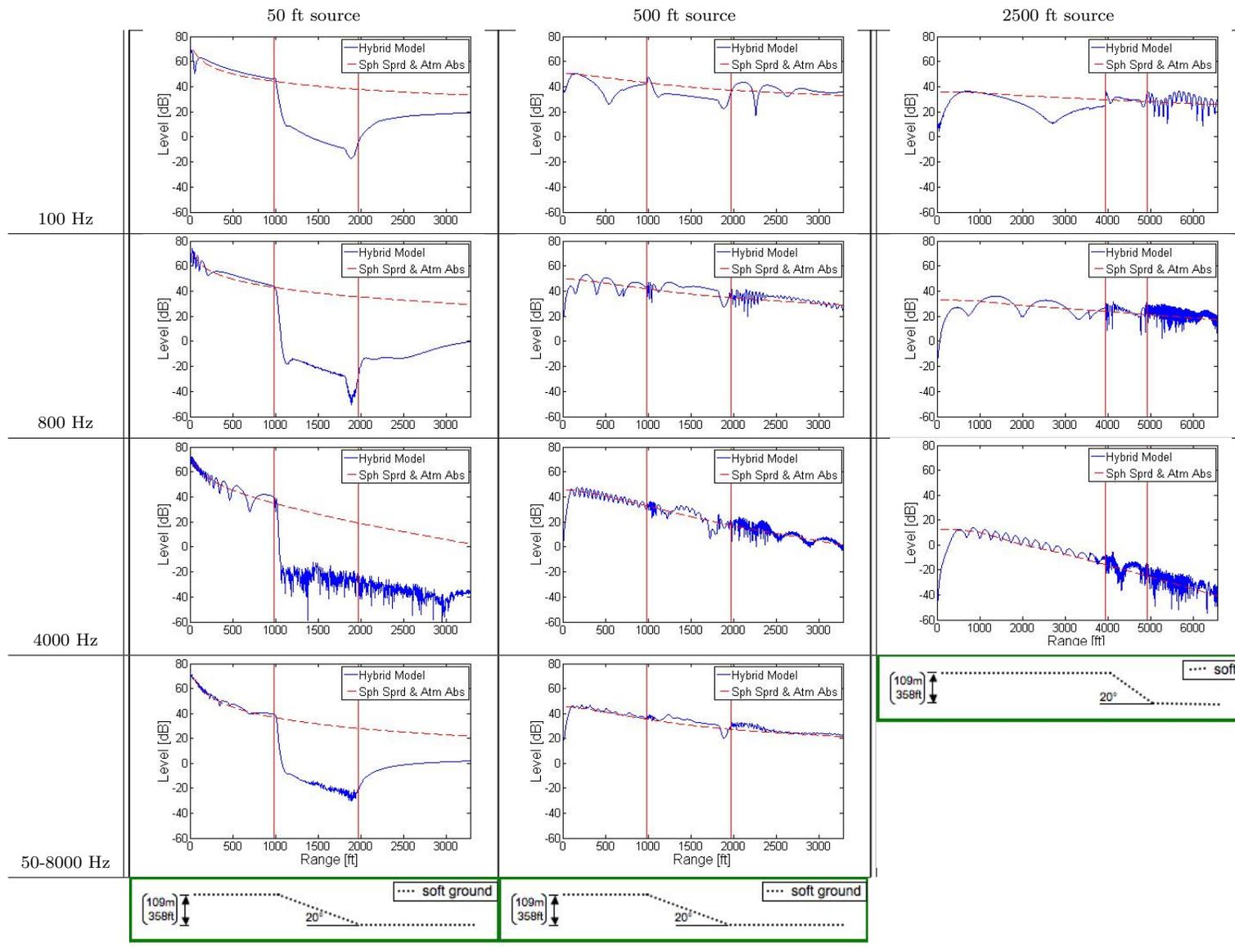


Figure 4.6. Results for Test Case 4, downward sloping, soft ground. The first, second, and third columns show results for 50, 500, and 2500 ft source heights, respectively. The first, second, and third row show results for 100, 800, and 4000 Hz source, respectively. The fourth row shows averaged results for frequencies 50-1000 Hz in 1/3-octave bands and 2000-8000 Hz in octave bands for the 50 and 500 ft source heights. Geometries of the test cases are illustrated below the result figures. Note the different range used for the 2500 ft source.

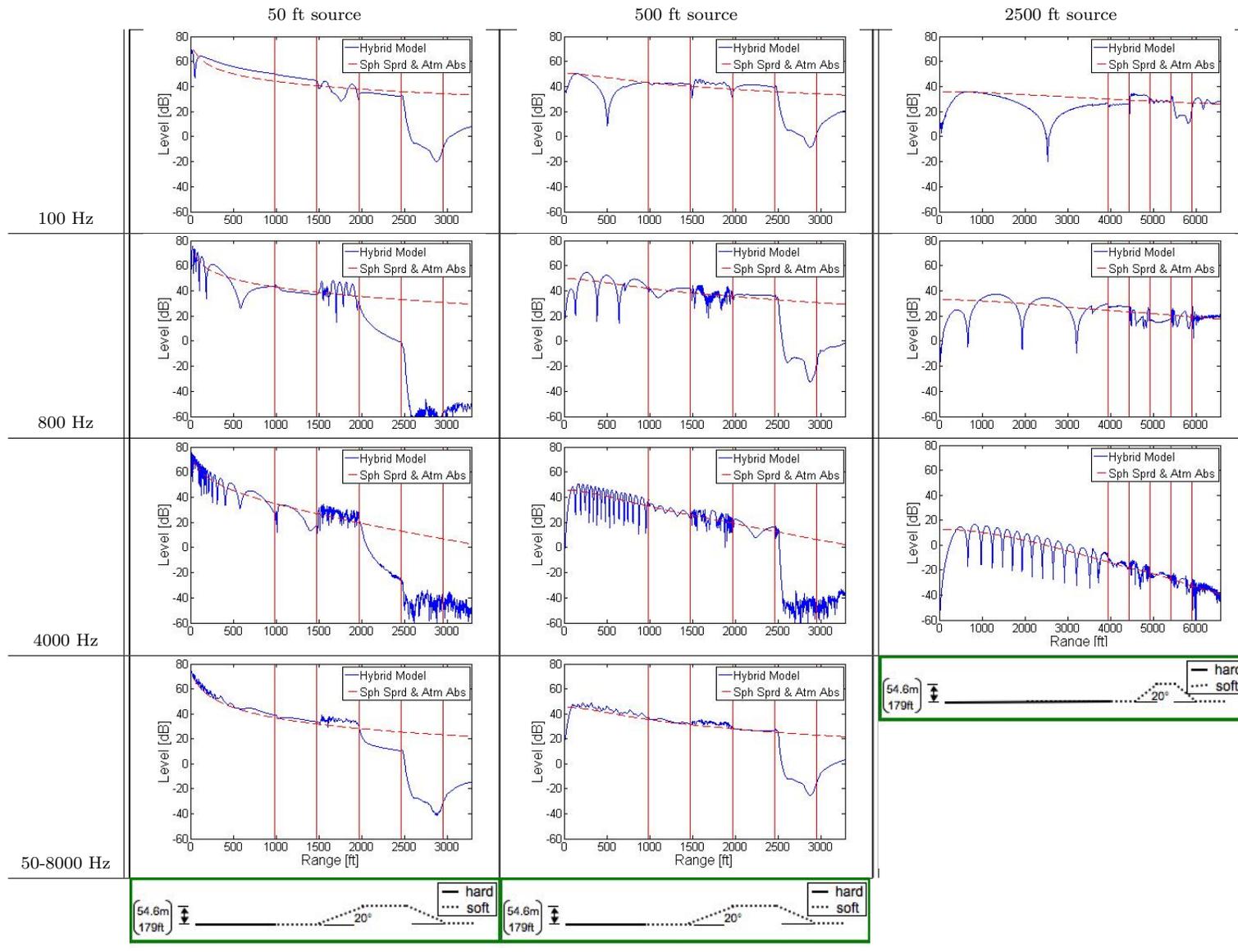


Figure 4.7. Results for Test Case 5, hill, hard-soft ground. The first, second, and third columns show results for 50, 500, and 2500 ft source heights, respectively. The first, second, and third row show results for 100, 800, and 4000 Hz source, respectively. The fourth row shows averaged results for frequencies 50-1000 Hz in 1/3-octave bands and 2000-8000 Hz in octave bands for the 50 and 500 ft source heights. Geometries of the test cases are illustrated below the result figures. Note the different range used for the 2500 ft source.

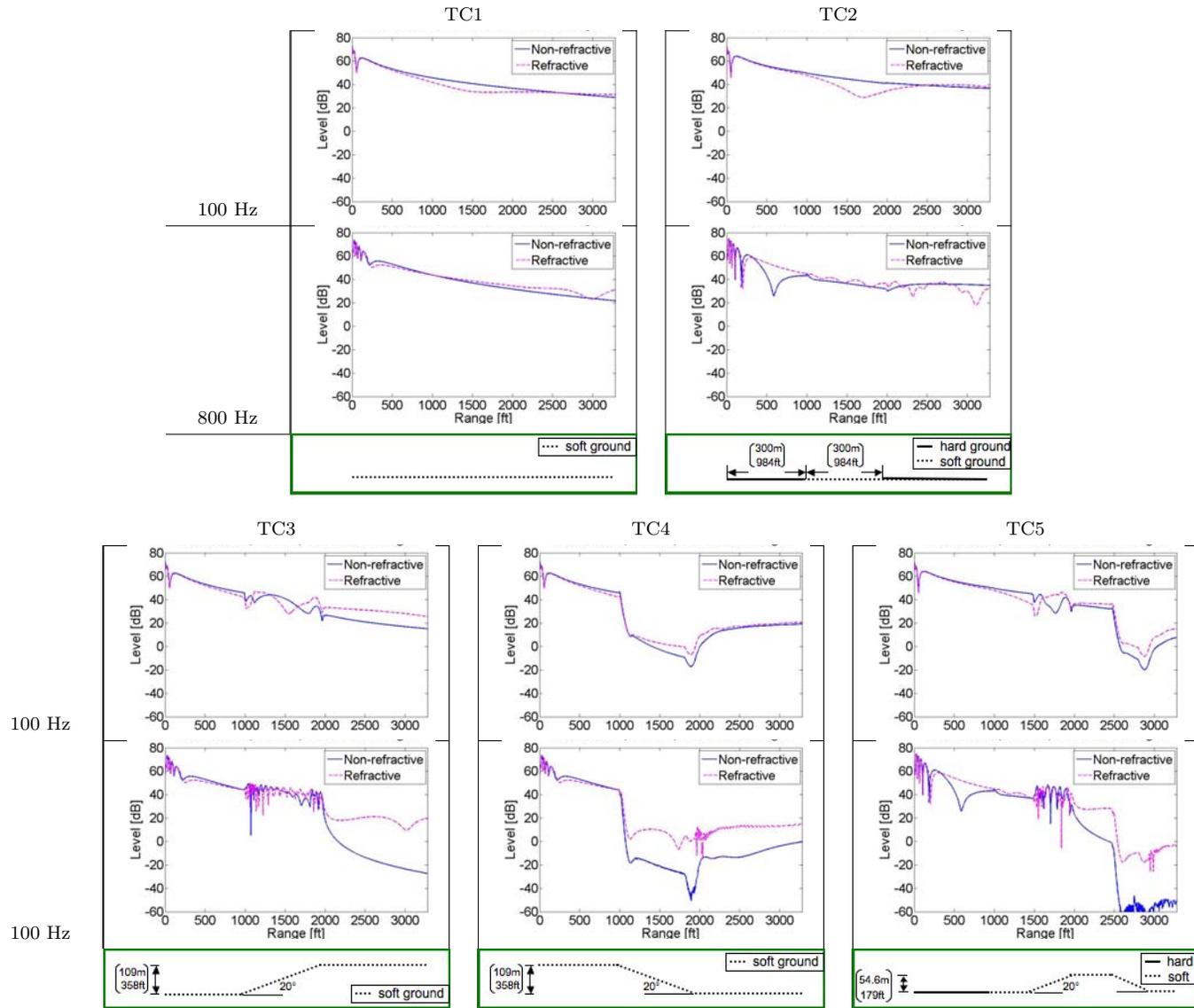


Figure 4.8. Results for Test Cases 1–5, for a downward refractive atmosphere, $b = 1$ m/s. Results are shown for 100 and 800 Hz sources at 50 ft height. The magenta lines show refractive atmosphere results and are compared to blue lines, which shows homogeneous atmosphere results. Geometries of the test cases are illustrated below the result figures.

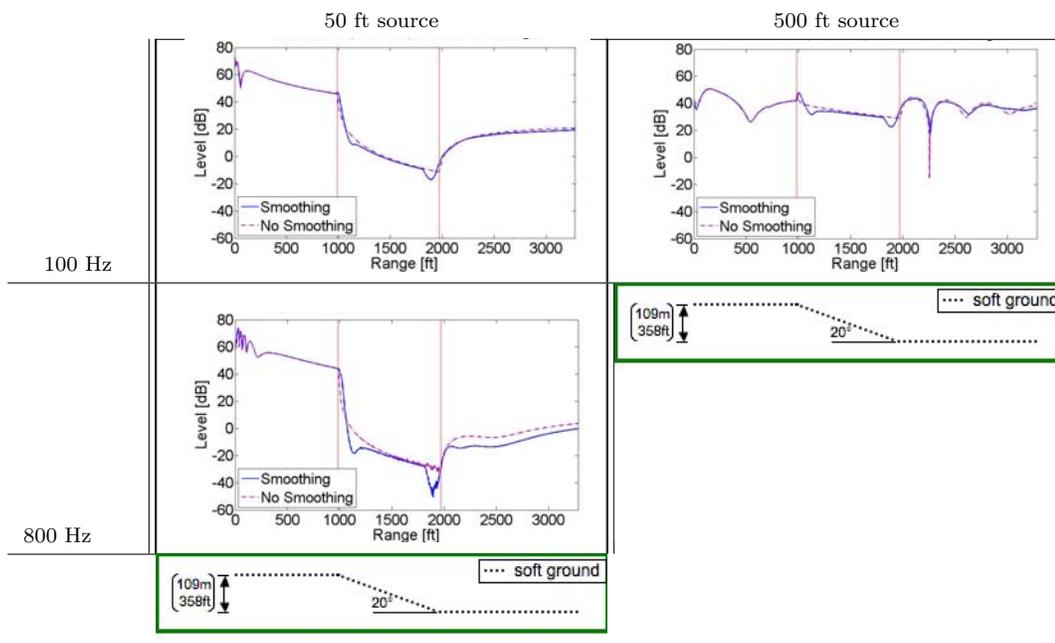


Figure 4.9. Results for Test Case 4, downward sloping, soft ground, in a homogeneous atmosphere with and without terrain smoothing. The first and second columns show results for 50 and 500 ft source heights, respectively. The first and second rows show results for 100 and 800 Hz sources, respectively. Geometries of the test cases are illustrated below the result figures. The dash-dot lines show results for which the terrain profile was represented as a series of straight line segments and are compared to the solid lines, which show results for a smoothed terrain profile.

4.3.2.4 Comparison with NMSim

While flat ground conditions have been validated against analytical and theoretical models, such benchmarks are not available to validate the model's performance in the presence of topography. Therefore, a comparison between results of the hybrid model and Wyle Laboratories' Noise Model Simulation (NMSim) propagation model [67], was made for further investigation of the hybrid model's terrain performance. The NMSim propagation algorithms are based on ray tracing and incorporate the effects of spherical spreading, atmospheric absorption, terrain, finite ground impedance, and directional sources. The effect of a vertically stratified atmosphere can be included, but only over flat terrain [29]. As a simulation model, NMSim is able to produce time histories of sound produced by moving sources. However, it is also capable of producing results using standard cumulative metrics

such as sound exposure level (SEL) and equivalent noise level (Leq). With high computational efficiency, NMSim runtimes are generally quite short, on the order of seconds, even for complicated propagation conditions.

Results for Test Cases 3 and 4, upward and downward sloping ground, respectively, were compared with NMSim. To establish a baseline of comparison, results of Test Case 1, flat soft ground, were compared between the two models as well. The input file of terrain elevation for NMSim is defined in the $x - y$ plane, while the test case terrains were defined only in one range dimension. Therefore, the geometries of the test cases were extruded into the second range dimension. The resulting geometry can be observed at the bottom of Figures 4.11 and 4.13, which show elevation contour plots of Test Cases 3 and 4, respectively. While the terrain geometry used in NMSim is not an exact comparison to the assumed axisymmetric geometry of the hybrid model, the differences were assumed to be small.

The source definition in NMSim is formed from user-defined segment endpoint locations. These segments are combined to form a realistic flight or ground track. The user specifies source speed and heading at the endpoints. To achieve the stationary point source conditions of the test cases, two endpoints at the same location were defined with a source speed equal to zero. An omnidirectional source option is available within NMSim. However, a specific source spectrum is assumed. Therefore, the associated source files were altered to perform comparisons for single 1/3-octave bands at a time.

NMSim is able to produce noise contour plots of level predictions. However, exact receiver locations can also be specified for user-controlled data output. The user-defined receiver location option was utilized and receiver locations were positioned along a straight line from the source, perpendicular to the terrain features. They were positioned with higher spatial resolution where quicker changes in level were expected—near the source, over the terrain feature, and just beyond the end of the sloped regions. Receivers were placed at a height of 4 ft above the ground.

NMSim and hybrid model results are shown in Figures 4.10, 4.12, and 4.14 for Test Cases 1, 3 and 4. They are again plotted against a line that denotes the combined effects of spherical spreading and atmospheric absorption, with the same absorption coefficients used in the hybrid model runs. These coefficients are different than those applied in NMSim, which uses the ANSI S1.26 (R2004)

standard.

Two different sources were run for each case: a 100 Hz source, and a 4000 Hz source. The reference level at 1 meter was arbitrary. However, it was recorded and used to correct the results to the 94 dB reference used in the hybrid model runs.

4.3.2.4.1 Test Case 1 Results were collected for Test Case 1, flat, soft ground, to identify any differences in the two models for the simplest propagation conditions. The comparison of results from the hybrid model and NMSim are shown in Figure 4.10. Figures 4.10a and 4.10b show results for 100 Hz and 4000 Hz sources, respectively.

The results for both the 100 Hz and 4000 Hz sources show generally good agreement between the two models. Differences that do emerge are larger for the 4000 Hz source. However, the results are quite similar at small ranges, and grow larger with distance, eventually reaching approximately 6.1 dB at 3281 ft (1000 m) from the source in range. NMSim shows higher noise levels than does the hybrid model. The range- and frequency-dependence of the disagreement between the hybrid model and NMSim suggests they may stem from differences in atmospheric absorption calculation. The atmospheric absorption standards used by INM and NMSim, SAE-ARP-866A and ANSI S1.26, respectively, are different, as shown in the comparison in Table 4.3. The standard used by INM, and the hybrid model, produces larger attenuation at 4000 Hz, consistent with the comparison of Figure 4.10b.

4.3.2.4.2 Test Case 3 A screenshot of the NMSim environment with Test Case 3 conditions is shown in Figure 4.11. The upward sloping ground can be seen at the top of the figure, where the designation of soft ground is also indicated. The center window shows the level over time—here constant for the stationary source. The illustration at the bottom of the window shows a top-down contour plot of the terrain, where receiver locations are marked by circles.

The comparison of results from the hybrid model and NMSim for Test Case 3 are shown in Figure 4.12. Figures 4.12a and 4.12b show results for 100 Hz and 4000 Hz sources, respectively. The basic patterns of sound level results are similar between the hybrid model and NMSim:

Table 4.3. Atmospheric absorption coefficients of the SAE-ARP-866A [9] and ANSI S1.26 [10] standards, used in INM and NMSim, respectively. 25 °C air temperature and 70% relative humidity.

Frequency [Hz]	Absorption coefficient [dB/km]	
	SAE-ARP-866A	ANSI S1.26
50	0.3	0.0484
63	0.3	0.0765
80	0.3	0.121
100	0.7	0.190
125	0.7	0.296
160	1.0	0.460
200	1.3	0.706
250	1.3	1.06
315	2.0	1.57
400	2.3	2.24
500	3.0	3.08
630	3.6	4.05
800	4.6	5.09
1000	5.9	6.19
1250	7.5	7.35
1600	9.8	8.68
2000	13	10.4
2500	17	12.8
3150	23	16.3
4000	31	21.9
5000	36	30.5
6300	52	44.1
8000	72	65.4

Consistent with the hybrid model, NMSim shows a decrease in level beyond the terrain incline where the receiver enters the shadow zone. Here, the level drops below spherical spreading and atmospheric absorption. The decrease from line-of-sight obstruction occurs over a short distance, after which the results decrease is more gradual, closer to the pattern of spherical spreading and atmospheric absorption. The dip in level beyond the incline is small for the low frequency, 100 Hz source and larger for the high frequency, 4000 Hz source.

Results for the 100 Hz source agree well in the regions before and after the terrain incline. Above the incline, where interference pattern behavior is observed,

the complicated pattern is not recreated in precisely the same form for both models. However, the patterns do resemble one another, capturing the nature of the sound behavior in this region. From comparison of results after the terrain flattens out, the overall effect of line-of-sight obstruction is found to be similar.

Results for the 4000 Hz source again agree well in the region in front of the topographical incline, and reveal similar complicated interference pattern behavior over the incline. Both models show severe drops in level immediately beyond the obstruction of the direct source-receiver path. However, there is a significant difference in the amount of attenuation registered, the hybrid model showing more than 20 dB additional attenuation than NMSim. Such a severe drop, as shown in the hybrid model, is unlikely to occur in realistic outdoor propagation conditions. In this case, there is an attenuation of more than 130 dB from the 94 dB reference level, corresponding to a ratio in intensity of $1 : 10^{-13}$. Rapid oscillations in level experienced below -40 dB are also shown in the hybrid model results. Because for the levels of sound this model is meant to process, these highly attenuated levels would not be detectible over ambient noise, the cause of these oscillations has not been thoroughly explored. If there was an interest in looking into this performance further, attention might first be directed to any reflections off the top of the grid that have not been completely attenuated by the absorbing layer of the PE.

4.3.2.4.3 Test Case 4 A screenshot of the NMSim environment with Test Case 4 conditions is shown in Figure 4.13. Here, the downward sloping ground can be seen at the top of the figure. Receiver locations are unchanged.

The comparisons for Test Case 4 are shown in Figure 4.14. Figures 4.14a and 4.12b show results for 100 Hz and 4000 Hz sources, respectively. Again, results for both sources are quite similar in the region before the start of the terrain feature. Over the downward sloping terrain, where line-of-site is broken, both models show a decrease in level.

A significant difference in attenuation exists between the two models over the downward sloping terrain for the 100 Hz source. Both models show a steeper decrease in level just beyond the start of the downward slope and more gradual decrease further over the slope. However, the level decrease is much larger in the hybrid model calculations and, as the terrain flattens again, a significant increase

in level is shown in the hybrid model results due to diffraction, which levels out at further distances from the terrain feature. NMSim results do not show a similar increase, remaining fairly flat. The results of the two models converge to similar levels at distance farther from the terrain feature.

Some differences between the two propagation models were anticipated to emerge for low frequency sound conditions. Because ray trace methods utilize high frequency limits while the PE and FFP have no such limitations, the hybrid model is expected to more accurately capture low frequency propagation behavior. The large differences that are seen between the models in this test case reveal that propagation in shadow zones that produce extreme drops in level benefit from prediction by models accurate to low frequencies.

The test case results shown in Figure 4.14a were produced for the hybrid model in a simple, homogeneous atmosphere. Additional hybrid model results were calculated to include the effects of turbulence to determine if the difference between the two models would be diminished in more realistic meteorological conditions. Hybrid model turbulence results are shown with a dash-dot line in Figure 4.15 for fairly strong atmospheric turbulence with normalized structure parameters $C_T^2/T_0^2 = 6 \times 10^{-7} \text{ m}^{-2/3}$ and $C_v^2/c_0^2 = 2 \times 10^{-6} \text{ m}^{-2/3}$. NMSim does not provide a method of explicitly introducing turbulence into the model and, therefore, the same NMSim data from Figure 4.14a is used. Over the downward sloping ground, the steep initial decrease in level is still observed. However, the decrease is smaller under turbulent conditions. Further, the level remains approximately constant over the remaining downward sloping region, rather than experiencing the gradual decrease of the non-turbulent conditions. The level does increase into the flat region behind the slope due to diffraction and maintains slightly higher levels for the rest of the propagation range. While the differences between the hybrid model and NMSim results are diminished under turbulent conditions, a significant difference ranging from approximately 7 to 18 dB is still observed.

The difference between the two models is less significant over the downward sloping ground for the 4000 Hz source. Figure 4.12b shows a severe drop in level for both models just past the start of the slope, which persists to longer ranges. For the 4000 Hz source, the hybrid model results generally level off, while experiencing rapid oscillation similar to those of Test Case 3. Therefore, the exact levels recorded

are likely less meaningful than the illustration of a significant decrease. Here, again, realistic outdoor propagation conditions would prohibit such small levels. Still, the presence of the rapid, severe drop is similar between the models.

While some differences were found between the hybrid model and NMSim, the influence of the terrain features do appear to be similar in the two models. Some differences manifest in shadow zones where predicted levels are too low to occur under realistic propagation conditions. However, for the lower frequency, 100 Hz sound, differences between results of the hybrid model and NMSim remain, even when turbulence is included. NMSim predicts significantly larger levels in the region over the downward sloping ground, where the shadow zone effects are most extreme. This difference may be the effect of high frequency limits used in the ray-based NMSim while the hybrid model has no such limitations. The differences in results directly above terrain features tend to decrease at larger ranges beyond the features.

4.3.2.5 Comparison with the INM Line-Of-Sight Adjustment

INM uses a Line-of-Sight Blockage Adjustment (LOS_{ADJ}) to roughly include the effect of terrain obstructions based on the difference in path length between the direct, obstructed path from source to receiver, and the path over the terrain [4]. The method of LOS_{ADJ} calculation is discussed in Section A.3.1.3. As an additional validation measure for the hybrid model test case results including terrain, the INM algorithms were programmed for comparison.

The INM method of terrain effect implementation utilizes “attenuation caps” with a maximum allowable attenuation of 18 dB. Only extremely small path length differences (less than 0.1 m) produce attenuation below the cap. Therefore, the 18 dB cap is reached just beyond the break in line-of-sight. The 18 dB attenuation in shadow zones is consistent with the broadband results of Test Case 3 and the initial line-of-sight blockage of Test Case 5. However, they do not resemble the more significant line-of-sight blockage conditions in Test Case 4 and longer ranges in Test Case 5. More significant attenuation is experienced over the downward incline of Test Case 4 than after the upward incline of Test Case 3 because the line-of-sight blockage occurs at smaller ranges from the source—approximately 984 ft (300 m) versus 1968 ft (600 m). However, at ranges larger than 1968 ft (600 m)

from the source, the results of Test Case 4 increase, approximately matching the results of Test Case 3 at 3281 ft (1000 m), and approaching the 18 dB cap of INM. Therefore, INM's LOS_{ADJ} does resemble test case results for less extreme breaks in the line-of-sight

4.4 Conclusions

A hybrid PE-FFP model has been developed and tailored for use in aviation noise propagation modeling. The model was used to run five simple test cases in order to explore the effects of ground impedance, terrain, and a refractive atmosphere, for different source heights and frequencies. The results conformed well to expectations. The most significant effect was found to be the formation of shadow zones. The impact of this effect was highly dependent on source height and frequency and atmospheric conditions. Impedance discontinuity effects were found to be less significant. Some discontinuities were observed at the range point where the transition from FFP results to PE results occurred. However, these discontinuities were mostly small and occurred primarily for higher altitude, single frequency sources. The hybrid model behavior at the FFP to PE transition point was not investigated exhaustively. Therefore, the user is advised to use caution in applying the model if the FFP to PE transition occurs at a location that is of particular interest.

Where possible, the hybrid model has been validated against analytical models and published benchmark cases. Analytical models and published results do not exist for some of the test case results presented. For the effect of terrain features, comparisons with NMSim results and algorithms used in INM were made. Results were similar under many conditions between calculation methods. Some differences exist between the hybrid model and NMSim in the more extreme conditions including regions of interference behavior with rapidly varying levels and regions of significantly reduced levels caused by line-of-sight obstructions.

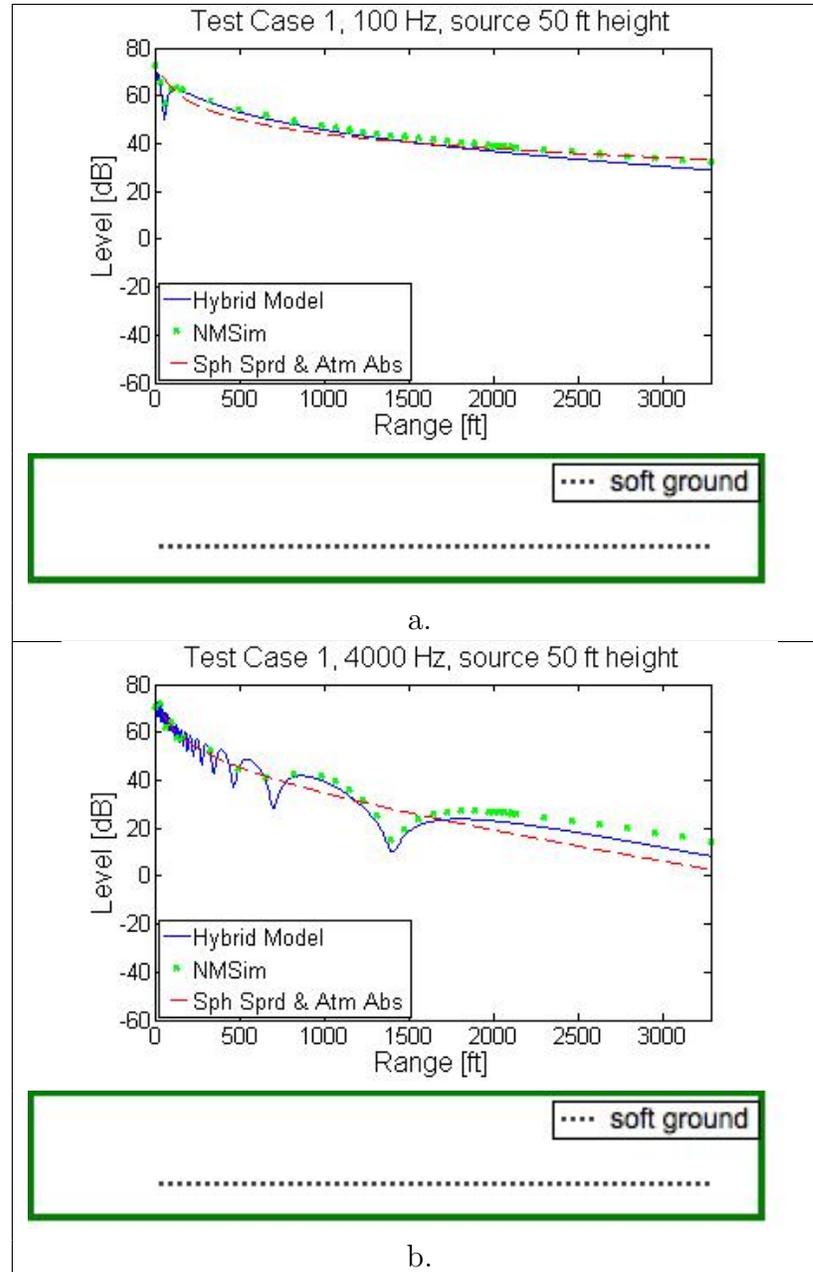


Figure 4.10. Comparison of Hybrid Model and NMSim results for Test Case 1, with (a) 100 Hz and (b) 4000 Hz sources. The solid line shows the Hybrid Model results, the dots show NMSim results, and the dash line shows spherical spreading and atmospheric absorption. Geometries of the test cases are illustrated below the result figures.

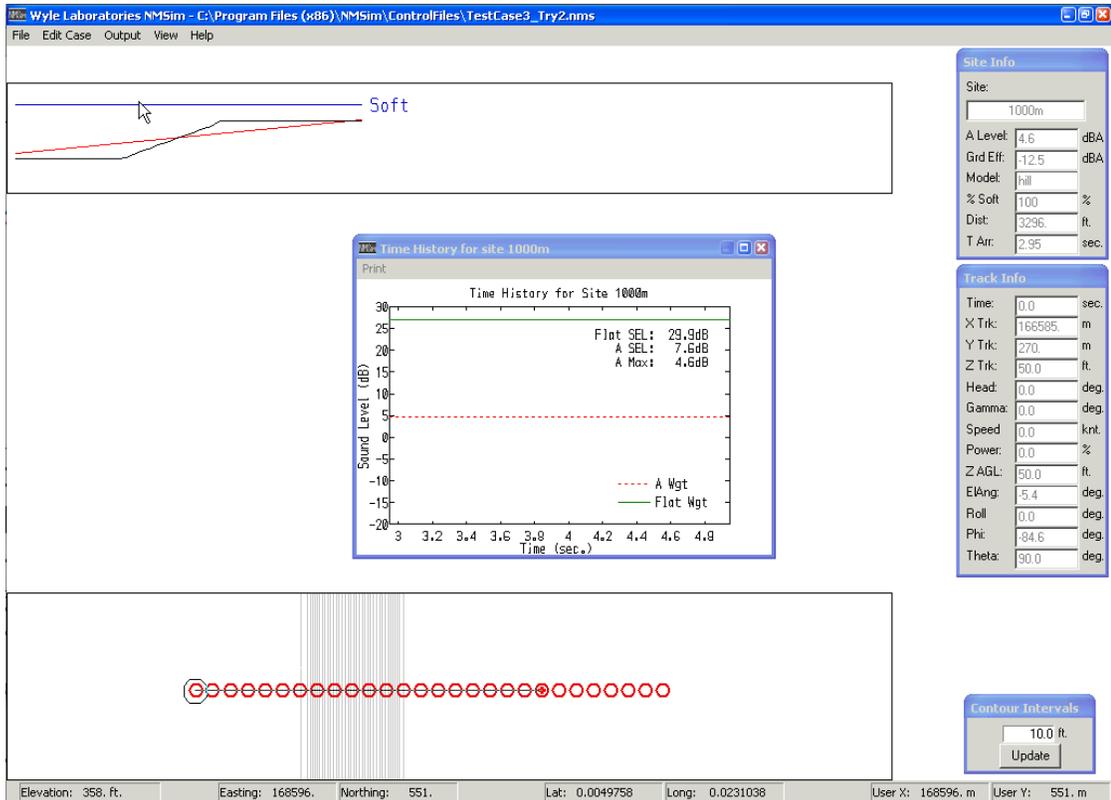


Figure 4.11. Screenshot of the NMSim run environment for Test Case 3. A cross section of the upward sloping terrain geometry and ground impedance specification are shown at top. The black line shows the terrain profile and the red line draws a straight connection between the source and the currently selected receiver. It is represented in red here because the path is obstructed by the terrain. A time history of level at a specific receiver location is shown in the center window. A top down, elevation contour view of the upward sloping terrain is shown with receiver locations at bottom.

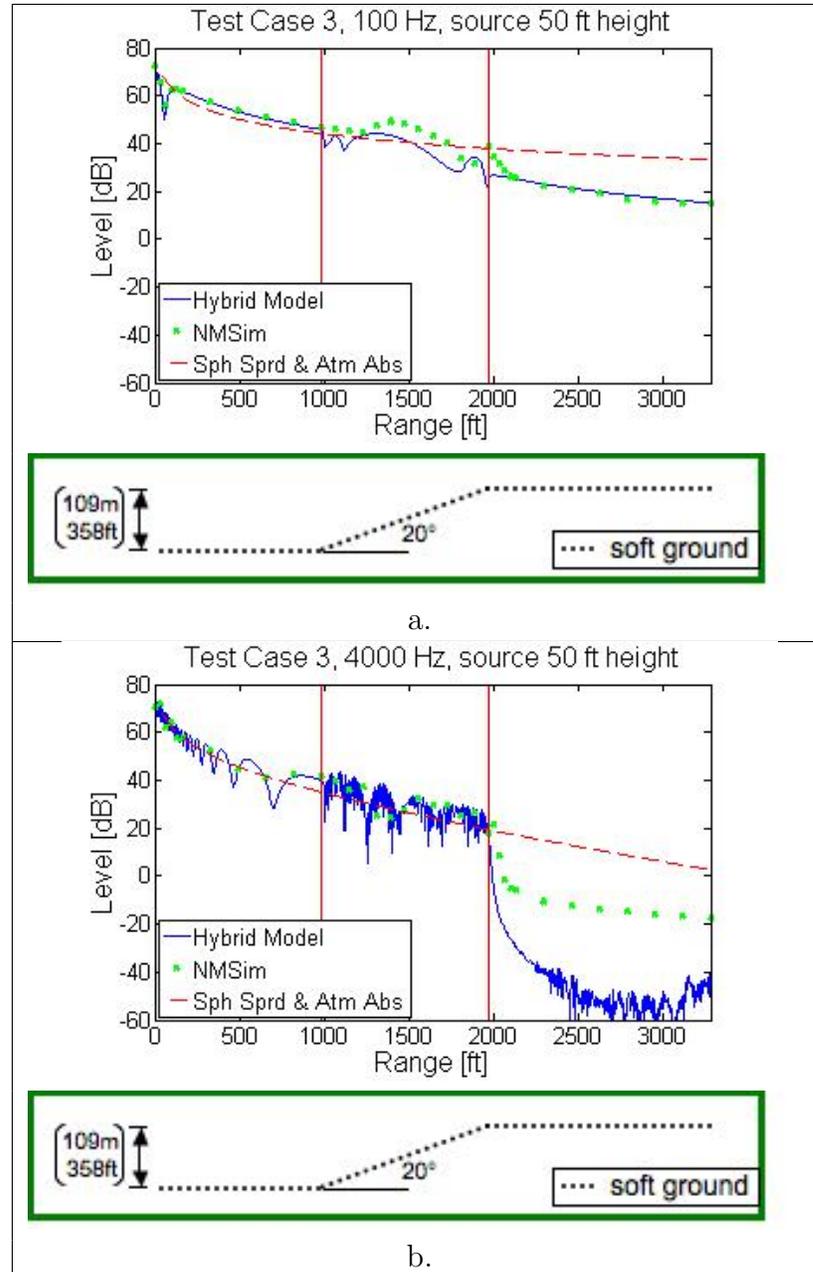


Figure 4.12. Comparison of Hybrid Model and NMSim results for Test Case 3, with (a) 100 Hz and (b) 4000 Hz sources. The solid line shows the Hybrid Model results, the dots show NMSim results, and the dash line shows spherical spreading and atmospheric absorption. Geometries of the test cases are illustrated below the result figures.

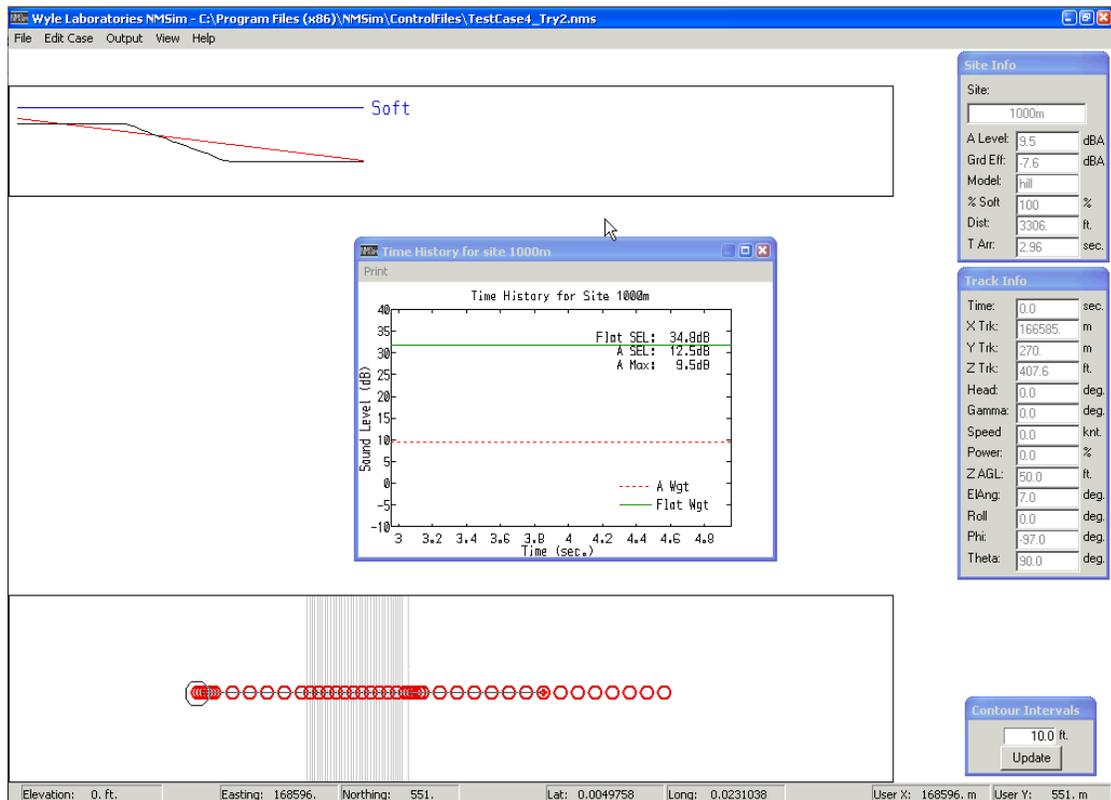


Figure 4.13. Screenshot of the NMSim run environment for Test Case 4. A cross section of the downward sloping terrain geometry and ground impedance specification are shown at top. The black line shows the terrain profile and the red line draws a straight connection between the source and the currently selected receiver. It is represented in red here because the path is obstructed by the terrain. A time history of level at a specific receiver location is shown in the center window. A top down, elevation contour view of the downward sloping terrain is shown with receiver locations at bottom.

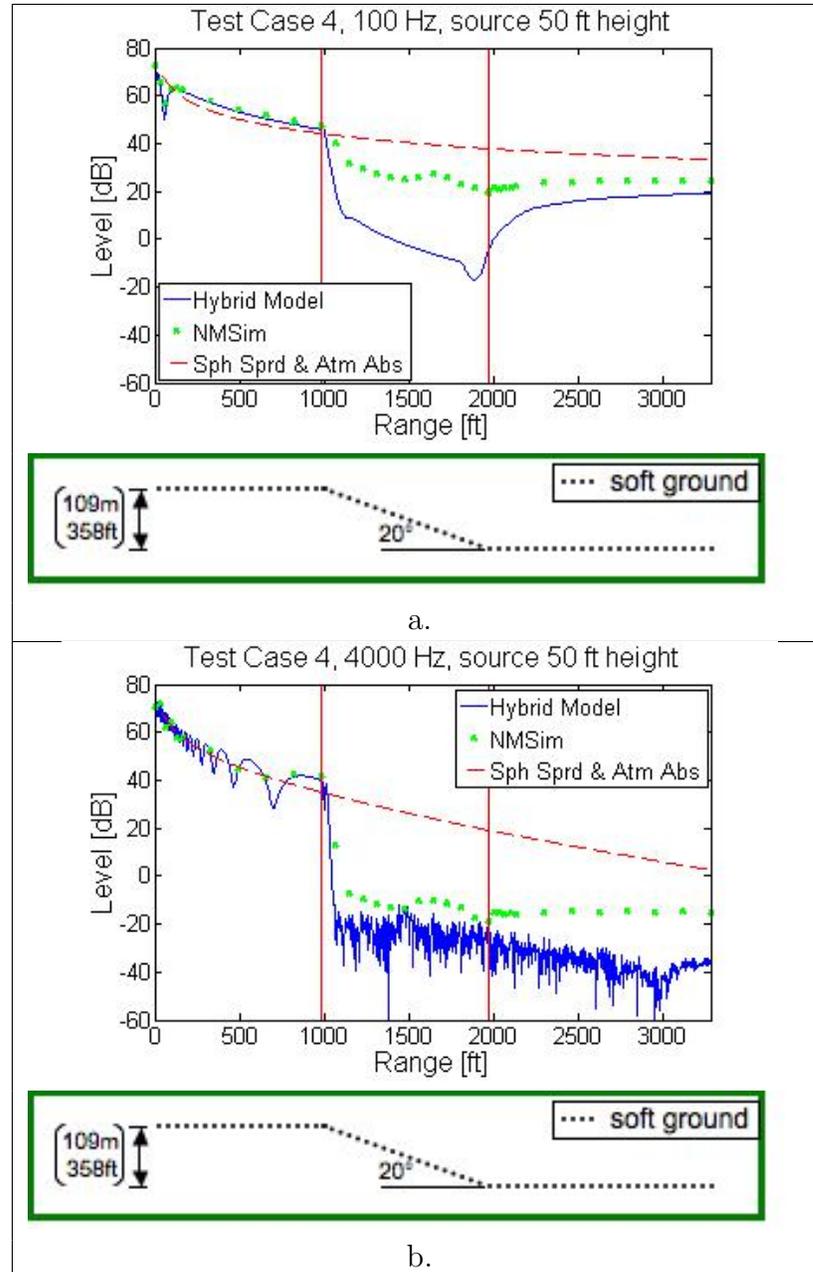


Figure 4.14. Comparison of Hybrid Model and NMSim results for Test Case 4, with (a) 100 Hz and (b) 4000 Hz sources. The solid line shows the Hybrid Model results, the dots show NMSim results, and the dash line shows spherical spreading and atmospheric absorption. Geometries of the test cases are illustrated below the result figures.

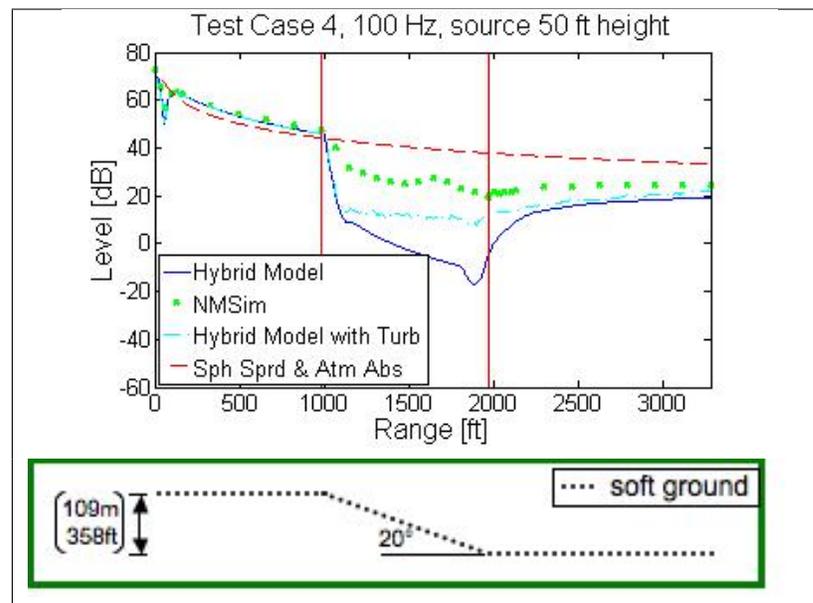


Figure 4.15. Comparison of Hybrid Model and NMSim results for Test Case 4 with a 100 Hz in a turbulent atmosphere with normalized structure parameters $C_T^2/T_0^2 = 6 \times 10^{-7} \text{ m}^{-2/3}$ and $C_v^2/c_0^2 = 2 \times 10^{-6} \text{ m}^{-2/3}$. The solid line shows the Hybrid Model results without turbulence, the dots show NMSim results, the dash-dot line shows the Hybrid Model results with turbulence, and the dash line shows spherical spreading and atmospheric absorption. The geometry of the test case is illustrated below the result figure.

Pseudo-Three-Dimensional Hybrid PE-FFP Model

5.1 Introduction

Once the PE and FFP have been combined in the hybrid model, the full two-dimensional vertical sound field for a stationary point source is represented accurately at each gridpoint for cases without extreme range-dependent changes near the source. However, aviation noise maps are customarily presented as noise contours in a two-dimensional horizontal plane at a specified receiver height. To produce this form of contour, the hybrid PE-FFP model must be expanded from the two-dimensional vertical plane to a three-dimensional model. The noise levels at a specific height can be obtained from the three-dimensional model [68].

This chapter describes the expansion of the hybrid model from two to three dimensions. It begins with a summary of the process. It continues with a more detailed explanation of the extension from single point source to flight path representation, including the transformation process of terrain and impedance information into a form that can be interpreted by the propagation model. Finally, numerical examples are presented to demonstrate the contouring capabilities of the expanded model.

5.2 Summary of the Two- to Three-Dimension Expansion Process

This section outlines the high level steps of constructing the pseudo-three-dimensional model from the required user inputs. Much of this process is centered around the diagram in Figure 5.1. This diagram shows a top-down view of a horizontal area over which propagation is to take place. Using numerical methods, this propagation area must be described by a finite number of discrete points within the space. The Cartesian system underlying the three cylindrical systems represents the grid on which terrain and impedance information is imported into the model as well as the ultimate representation of the returned aviation noise contour maps. The cylindrical coordinate systems represent the points at which direct calculation is performed by the hybrid propagation model. Each cylindrical coordinate system is employed for a particular representative source point along the given flight path and each system's origin is offset from the others' by the distance between the source points. The diagram demonstrates the disconnect between the Cartesian grid system on which information about the propagation area is initially provided, and ultimately returned, and the multiple cylindrical coordinate systems on which numerical propagation actually takes place. The following outline of the expansion process describes how this disconnect is resolved.

1. Defining the contouring grid

The user defines the boundaries of the horizontal Cartesian grid on which results are to be returned. For example, the grid may include the range in x : [-175 m, 175 m] and in y : [-30 m, 320 m]. The grid definition is used to determine the ranges, in both horizontal dimensions, to which the contouring grid will extend.

2. Representing the continuous flight path with discrete source locations

The user defines the flight path as a set of representative (x, y, z) coordinates along the continuous path. The model is run separately for each individual source point and the results are later combined.

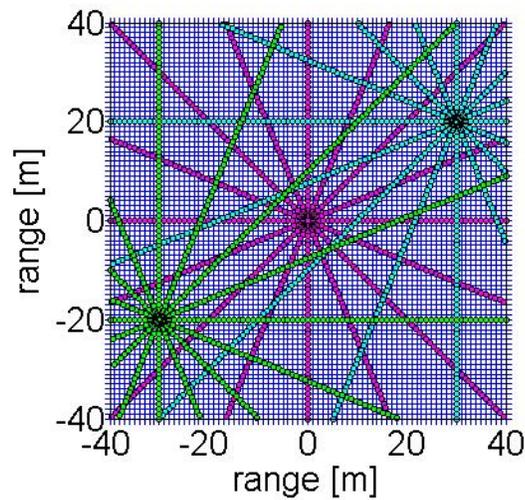


Figure 5.1. Radial systems centered at each source point along a flight path, superimposed on the Cartesian coordinate results grid.

3. Determining terrain and ground impedance geometries for each source point

The user provides a definition of terrain elevation and ground impedance in a rectangular, Cartesian grid form. However, the PE and FFP models propagate sound outward from the source. Therefore, data from rectangular terrain and impedance grids is interpolated to separate radial coordinate systems with a specified number of radial angles, centered at each representative source point, as shown in Figure 5.1. In this form it can be used in the two-dimensional hybrid propagation model.

4. Calculating the effective sound speed profile for all azimuthal angles from each source, including scalar contributions of temperature and vector contributions of wind

The user provides information about the sound speed profile produced from temperature gradients as well as the wind speed and direction. The vector characteristic of the wind produces an effective sound speed profile that varies with azimuthal angle from the source. Therefore, the correct parameter at each angle from the source is determined by including both temperature and wind contributions.

5. Running the model

Once all parameters of propagation effects are fully defined along all radial paths from each source point, the vertical propagation model can be run. It is run separately both for each source representation along the flight path and for each frequency. The user selects an angle increment that defines the discrete set of radial angles at which the vertical hybrid model will be run outward from the source. In Figure 5.1, for example, an angle increment of $\pi/8$ or 22.5° has been selected, requiring 16 independent runs of the vertical hybrid model around the source point, extending outward to the edge of the propagation grid.

6. Calculating cumulative results

To allow for easier calculation of the aggregate levels from all source points, noise data is interpolated back to a rectangular grid to show the total noise produced by the flight path.

Using this process, a horizontal sound field at a specified receiver height can be calculated to include the effects of three-dimensional terrain, impedance discontinuities, and meteorology for a given frequency. To generate the sound field for propagation from broadband sources, the model is run at each 1/3-octave band center frequency within the given frequency range and results are averaged together, weighted by the bandwidth of the 1/3-octave band. The sound field is used to generate standard aviation noise contours [69]. Results are saved and plotted.

5.3 Propagation from a Single Point Source

This section describes, in further detail, the process of finding the sound field produced by a single point source along the flight path. Figure 5.2 shows an example of a vertical flight path in the $x - z$ plane, where the symbols indicate the location of each represented point source. For each point, a two-dimensional horizontal sound field must be constructed. The transition from the two-dimensional vertical sound fields produced by the standard forms of the PE and FFP methods to the two-dimensional horizontal sound field proceeds as follows:

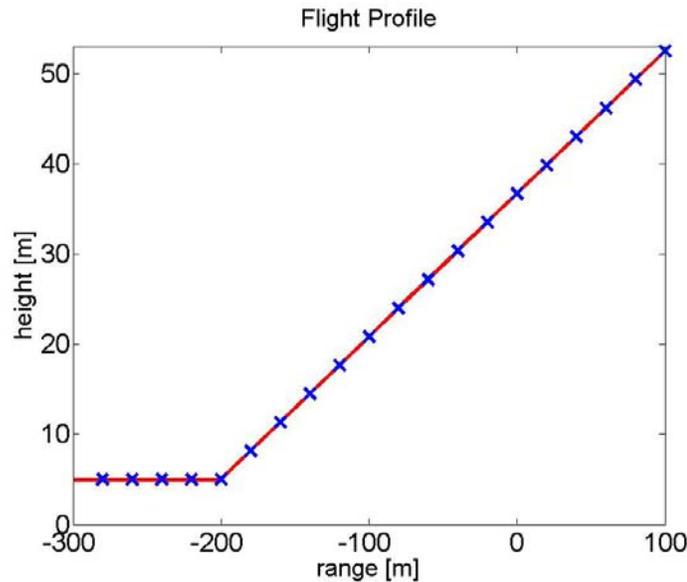


Figure 5.2. Vertical flight profile in the $x - z$ plane: symbols represent locations of 21 point sources along a sample flight profile.

1. The two-dimensional vertical hybrid model is run radially outward from the source location at specified increments of the polar angle. The form of these results corresponds to a three-dimensional cylindrical coordinate system (Figure 5.3).
2. A slice of the three-dimensional sound field at the desired receiver height is taken. The form of these results correspond to a two-dimensional polar coordinate system.
3. The sound field is interpolated to a rectangular grid. The form of the results now corresponds to a two-dimensional Cartesian coordinate system (Figure 5.4).

Figure 5.5 shows the contours produced at a height of 2 m by this process for a 50 Hz source at 100 m height in a homogeneous atmosphere over a soft ground. The contours produced by the four point linear interpolation in the polar coordinate system are smooth.

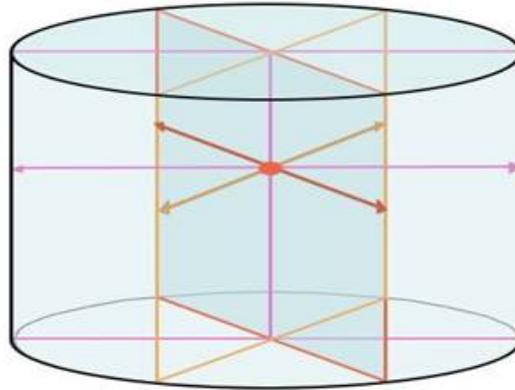


Figure 5.3. Radial run results are defined in a cylindrical coordinate system.

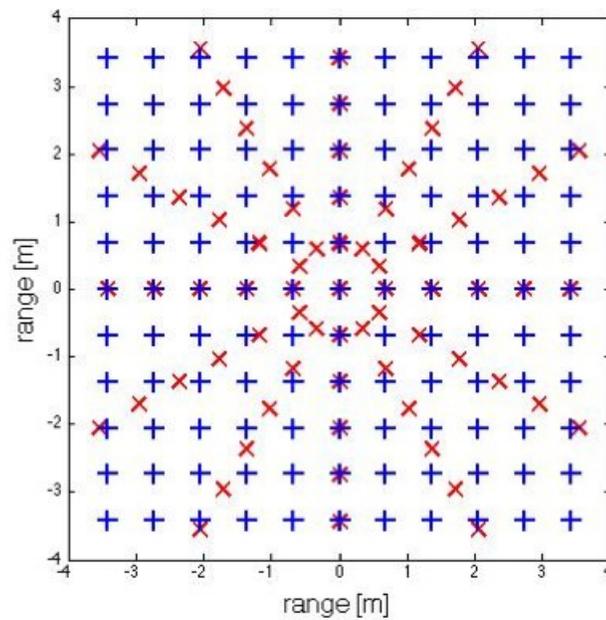


Figure 5.4. Polar gridpoints calculated with hybrid model (x) and interpolated Cartesian gridpoints (+).

5.4 Propagation from a Flight Path

Section 5.3 described the method for constructing the two-dimensional horizontal field for each individual point along the flight profile. This section shows the results of adding these sound fields together to find the cumulative noise level produced by the flight path. Figure 5.6a shows the transmission loss for a 50 Hz source moving along the vertical flight path shown in Figure 5.2 and a horizontally straight path

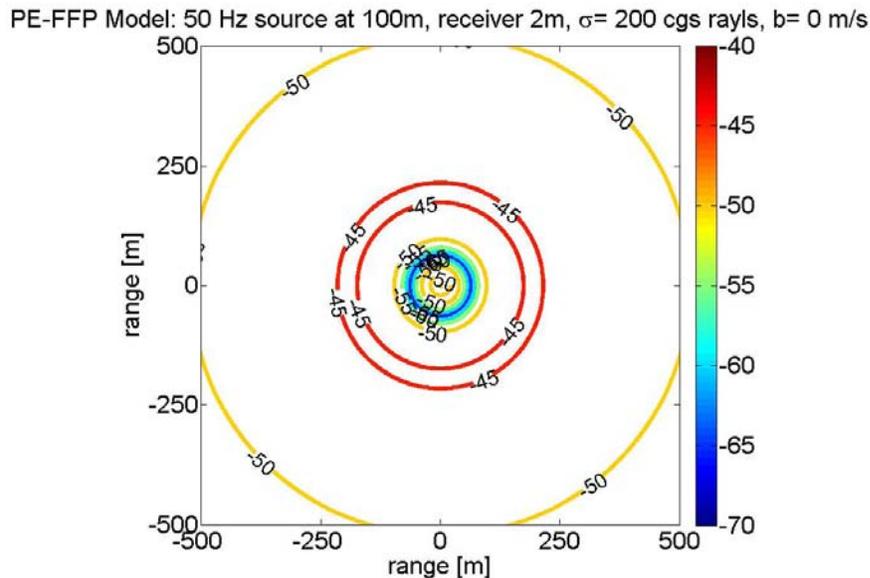
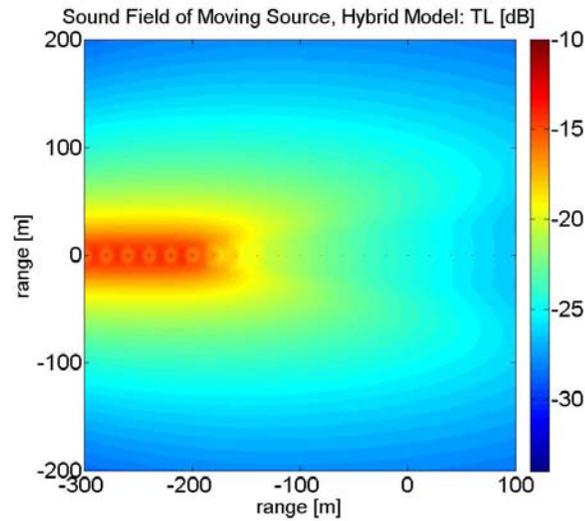


Figure 5.5. Transmission loss [dB] contours for 50 Hz source at 100 m height, homogeneous atmosphere, flat, soft ground with $\sigma = 200$ cgs rayls, receiver height of 2 m, calculated with hybrid model.

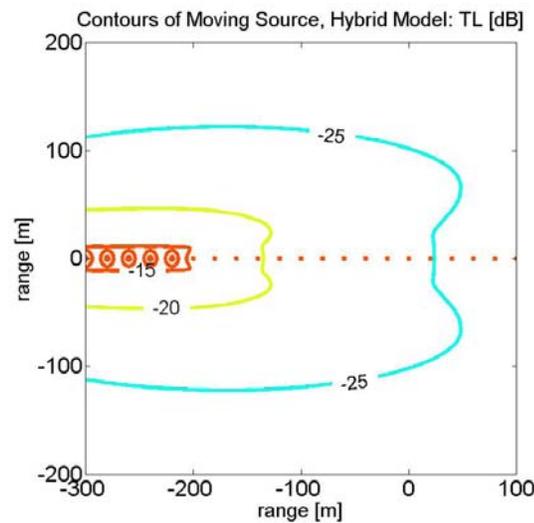
from left to right at a y -range of 0 m with a receiver at 2 m height. Transmission loss is defined as the total acoustic pressure calculated at a gridpoint as referenced to the acoustic pressure of the direct sound field at a distance of 1 m from the source [1]. Conditions include a homogeneous atmosphere and a flat, soft ground with an effective flow resistivity of 200 cgs rayls. Figure 5.6b shows the corresponding contour representation of the transmission loss. Length scales are shown in meters.

The flight profile (Figure 5.2) shows the aircraft leaving the ground at a range of -200 m. In the horizontal sound field figure, the sound field begins to decrease at -200 m and continues to decrease at larger x -range values as the airplane climbs. Thus, the figure agrees well with what is expected for an airplane traveling with the given flight profile.

Although the general trend of the transmission loss field was as expected, an initially unanticipated concave shape is observed in the contours. To investigate this result, the hybrid PE-FFP model calculations were compared to, and found to be consistent with, the results of an analytical model. The concave contour shape was determined to be the result of the interference pattern of the single frequency computation. Because it runs more quickly than the hybrid model for high



a.

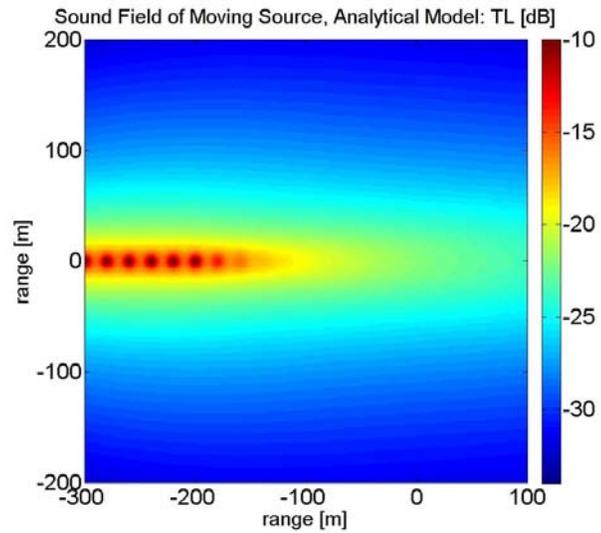


b.

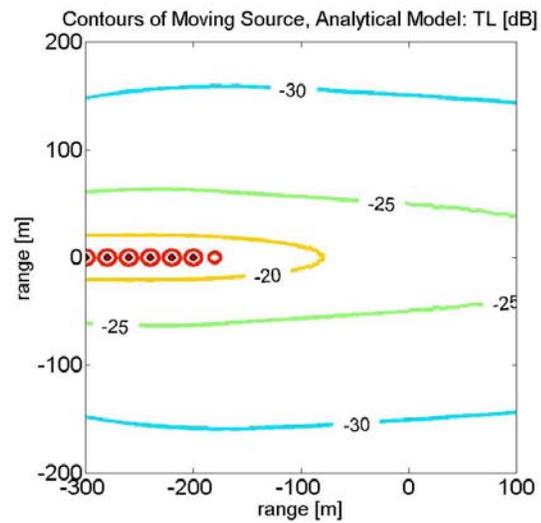
Figure 5.6. Transmission loss [dB] for a 50 Hz source following a given flight profile at a y -range of 0 m left to right, homogeneous atmosphere, flat, soft ground with $\sigma = 200$ cgs rayls, calculated with the hybrid model. (a) Sound field. (b) Contours.

frequency sound, the analytical model was used to test the contour construction methodology. As shown in Figures 5.7a and 5.7b, where the analytical model was run for a source comprised of 49 frequencies from 50 to 2000 Hz in 1/9-octave frequency bands, the broadband noise contours do not display the concave shape. Thus, it was confirmed that, when a broadband source is modeled, the different interference patterns of the various frequencies cause the concave shape to be washed

out.



a.



b.

Figure 5.7. Transmission loss [dB] for a broadband source 50-2000 Hz, 1/9-octave bands, following a given flight profile at a y -range of 0 m left to right, homogeneous atmosphere, flat, soft ground with $\sigma = 200$ cgs rayls, calculated with the analytical model. (a) Sound field. (b) Contours.

5.5 Numerical Examples of Contouring with Range-Dependent Effects

While the FFP is unable to include the effects of range-dependent conditions, including such effects is a strength of the PE. Thus, two scenarios were investigated in order to test the range-dependent contouring capabilities of the PE model component in a horizontal field. These two scenarios include uneven terrain, an impedance discontinuity, and a downward refracting atmosphere. Model results were calculated for a source traveling along a simple flight path [69].

Two sets of propagation conditions, designed to be fairly simple, while still displaying the capabilities of the model, are described. The second scenario, where uneven terrain is included, is a representation of an aircraft taking off in front of a barrier. The main concern in such a scenario is the sound that reaches the area behind the barrier, where a community may be located. Because the ranges behind the barrier are covered by the PE region of validity, this section will focus on the PE results. In the following descriptions, the $x - y$ plane corresponds to two-dimensional horizontal ranges and the z -axis represents height. Length scales are represented in meters.

The first scenario includes flat ground with an impedance discontinuity, from hard to soft ground, at $y = 30$ m. The second scenario includes the same impedance discontinuity from hard to soft ground, but also includes a hill with a width of 100 m and height of 10 m, centered at $y = 150$ m. The height of the terrain, as a function of range, is given by:

$$H(r) = a \cos^2 \left[\frac{\pi s}{2} \left(1 - \frac{r}{r_{top}} \right) \right], s = \frac{2r_{top}}{W} \quad (5.1)$$

where $H(r)$ is the height of the terrain in meters, r is range, a is the peak height of the hill, r_{top} is the range at which the hill peak occurs, and W is the width of the hill [44]. In both scenarios, a downward refracting atmosphere with the logarithmic profile, as described above, is used, with $b = 1$ m/s.

The flight path travels at $y = 0$ m, moving level to the ground at height 5 m until $x = -90$ m, at which point the aircraft takes off at a 10° angle from horizontal. Receivers are placed 1.22 m (4 ft) above the ground surface and code runs

are performed for 1/3-octave band center frequencies from 50-500 Hz and combined assuming the level of the center frequency is representative of the equivalent sound level over the 1/3-octave band. Illustrations of the flat and uneven terrain geometries, Scenarios 1 and 2, are shown in Figures 5.8 and 5.9, respectively.

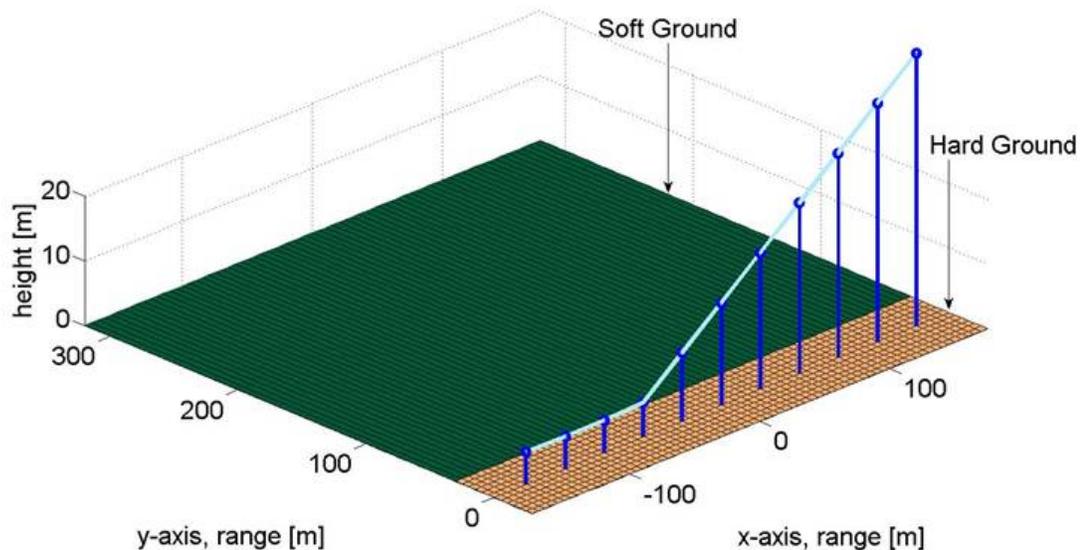


Figure 5.8. Geometry of Scenario 1, Flat Terrain. Hard ground, shown in tan, effective flow resistivity, $\sigma = 20,000$ cgs rayls from $y = -30$ m to 30 m. Soft ground, shown in green, effective flow resistivity, $\sigma = 150$ cgs rayls from $y = 30$ m to 320 m. Source path, shown in light blue, is level to the ground at 5 m height until $x = -90$ m, then increases height at 10° angle. Dark blue stem plot shows representative source locations.

The PE method is increasingly inaccurate at large elevation angles from the source. Therefore, although results are shown for all elevation angles in the following sections, results near the flight path will not be reliable. In Figures 5.10 and 5.12, the region outside the valid elevation angle range is shown between dashed lines [69].

5.5.1 Results from Flat Terrain Case

Figure 5.10 shows the results from Scenario 1, with flat ground propagation conditions. The figure is a top-down view of the sound field at the receiver height of 1.22 m (4 ft). The flight path is located at $y = 0$ m and increased source altitude extends out of the page. Sound level contours illustrate transmission loss [68].

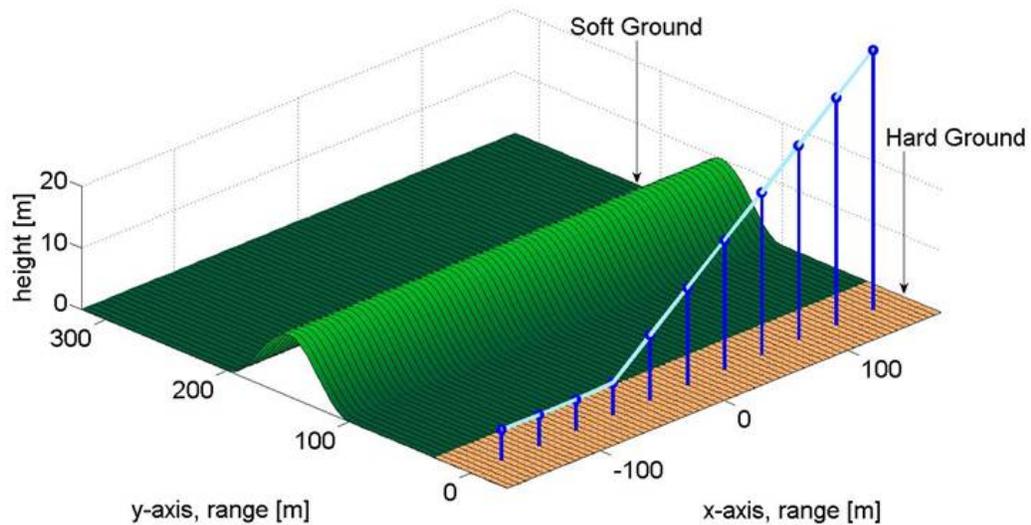


Figure 5.9. Geometry of Scenario 2, Uneven Terrain. Hard ground, shown in tan, effective flow resistivity, $\sigma = 20,000$ cgs rayls from $y = -30$ m to 30 m. Soft ground, shown in green, effective flow resistivity, $\sigma = 150$ cgs rayls from $y = 30$ m to 320 m. Source path, shown in light blue, is level to the ground at 5 m height until $x = -90$ m, then increases height at 10° angle. Dark blue stem plot shows representative source locations. Hill spans $y = 100$ m to 200 m with peak height 10 m at $y = 150$ m.

Results in Figure 5.10 show lower values of transmission loss at the larger positive y -axis values. This decrease is caused by the geometrical spreading and atmospheric absorption of sound traveling away from the sources of the flight path representation. Figure 5.10 also illustrates a decrease in level near the sources at larger x -axis values. Here, larger distances exist between source and receiver as the source altitudes increase.

The effect of the impedance discontinuity at $y = 30$ m is small relative to the large changes in sound level occurring near the flight path and is, therefore, hard to distinguish in the coarse contouring. However, Figure 5.11, displaying only the contribution of the source at $(0$ m, 0 m, 15.9 m) for 500 Hz, shows a noticeable change in the pattern of the sound level at the impedance discontinuity [69].

5.5.2 Results from Uneven Terrain Case

Figure 5.12 shows the results from the uneven ground propagation conditions. Because the PE method is unable to include backscattering from the terrain, the

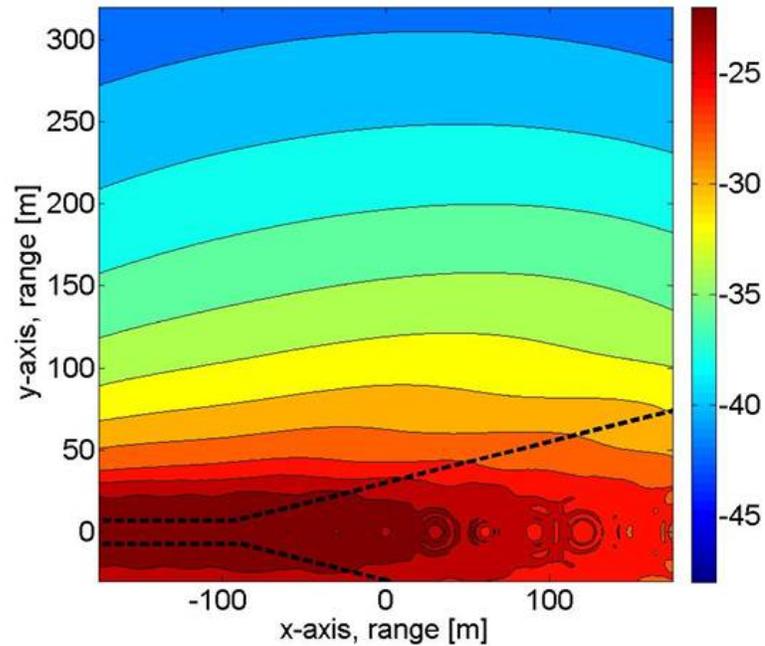


Figure 5.10. Transmission Loss Results of Scenario 1, Flat Terrain, in dB. Contours are shown as a bandwidth-weighted average of 1/3-octave band center frequency results from 50-500 Hz. Propagation geometry is as shown in Figure 5.8. Region between dashed lines is outside valid PE elevation angle range.

contours for this scenario are identical to the flat ground conditions from $y = -30$ m to $y = 100$ m, where the uneven terrain begins.

When a comparison is made of Figures 5.10 and 5.12 from $y = 100$ m to $y = 150$ m, the sound level results of Scenario 2, uneven terrain, are shown to be larger than those of Scenario 1, flat terrain. This occurs because of the interaction of sound with the upslope in Scenario 2. However, as the sound travels beyond the peak of the hill at $y = 150$ m, the levels in Scenario 2 drop rapidly. This reveals the effect of the shadow zone formed behind the hill. The shadow zone effect is more extreme beyond the hill at lower x -values as, here, the source is closer to the ground. As the source altitude increases and a more direct path exists from source to receiver at larger x -values, higher levels result [69].

These results, and those of Scenario 1, show the expected behavior of sound propagation in the specified conditions and geometries.

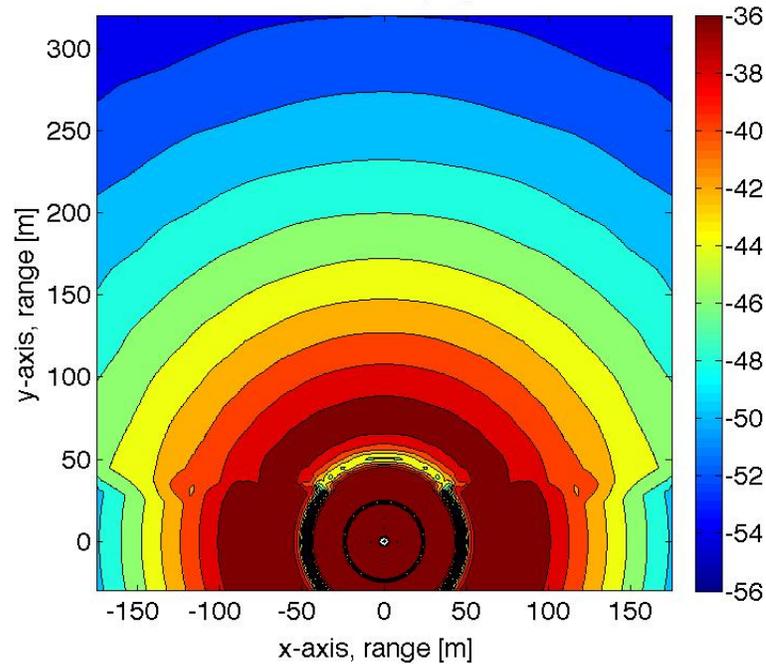


Figure 5.11. Transmission Loss Results of Scenario 1, Flat Terrain, in dB. Contours are shown for a single 500 Hz source at (0,0,15.9 m). Propagation geometry is as shown in Figure 5.8.

5.6 Conclusion

The process of extending the two-dimensional vertical sound field, calculated by standard propagation methods, to a pseudo-three-dimensional model was discussed in this chapter. The method interpolates terrain, ground impedance, and meteorological data to the polar coordinate systems required by the propagation models, and then interpolates the sound field results back to the more convenient Cartesian grid representation. By achieving pseudo-three-dimensional representation, a conventional noise contour map of a flight path is constructed.

Because the FFP is limited to range-independent conditions, two contouring scenarios were run with the PE component of the model to explore pseudo-three-dimensional, range-dependent effects. The scenarios included an impedance discontinuity, a downward refracting atmosphere, and a simple takeoff flight path. Scenarios with and without a hill were compared. The hill was found to cause an increase in level as sound interacts with the upslope of the hill, and a decrease in level past the peak of the hill as the receiver enters the shadow zone. The in-

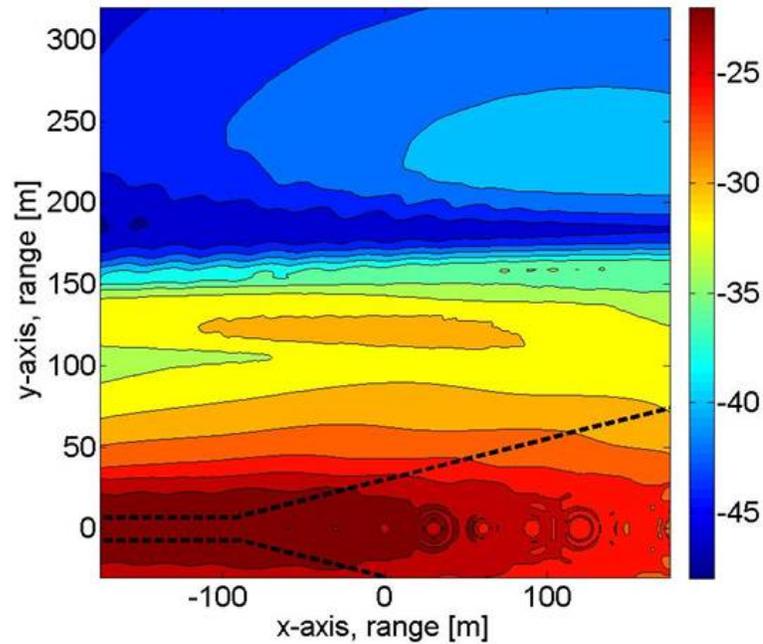


Figure 5.12. Transmission Loss Results of Scenario 2, Uneven Terrain, in dB. Contours are shown as a bandwidth-weighted average of 1/3-octave band center frequency results from 50-500 Hz. Propagation geometry is as shown in Figure 5.9. Region between dashed lines is outside valid PE elevation angle range.

creasing source altitude represented in the takeoff flight path also illustrates the decrease in strength of the shadow zone effect as the line-of-sight blockage became less extreme [69].

Inclusion of Elevation Angle Source Directivity in the Parabolic Equation Method with an Inverse Fourier Transform Technique

6.1 Introduction

Propagation models not only implement algorithms that predict the travel of sound after it leaves the source, but must also bridge the gap between the available form of source data, whether simulated or measured, and the required form of input for the propagation model. Therefore, the source data must be preprocessed and transformed into a useable model input while retaining all characteristics of the source. Adjustment of the overall source level is achieved by modifying the source strength. However, correct representation of directivity requires more complicated implementation. If enough information is known to determine the field at all points in height using a reverse propagation method, and other propagation effects, for example ground reflections, can be neglected, this representation can be used for the starting field. However, for more complicated propagation conditions, a generalized source directivity technique, outlined in this chapter, is derived for the unique source representation of the parabolic equation method.

The distinctive form of the PE starting field stems from the method used to advance the propagated sound in space. PE algorithms march results forward in range by extrapolating from vertical vectors of results at previous range steps. Therefore, to begin propagation in range, sound at the first range step must be fully defined, at all heights. Such a starting field can be regarded as a vertical source array. With this perspective, Fourier techniques can be used to synthesize an array that produces the desired far field source directivity [70].

The PE method has two inherent limitations that do not allow straightforward implementation of array synthesis techniques:

1. Vertical PE grid points must have a spacing of approximately $\lambda/10$, where λ is an average wavelength, for accurate prediction.
2. The starting field must be finite at all points and restrict energy emitted at large elevation angles from the source [1].

As a result of the first limitation, the required spacing of PE grid points is smaller than the achievable resolution, $\lambda/2$, of the sources of array synthesis from a far field directivity pattern [71]. The Fourier transform technique requires an extension of the vertical wave number spectrum, beyond the range of propagating wave numbers defined by the directivity pattern, into evanescent modes, to achieve the decreased spacing between array sources that meets the PE constraints.

The second limitation precludes the use of standard representations of some types of sources in the PE starting field. For example, the standard representation of a monopole diverges at the source and emits sound at large elevation angles. While this restriction prevents accurate representation of the near field for compact sources, approximations can be imposed on the starting field that allow for accurate far field predictions.

Two approaches to the array synthesis technique are explored in this chapter. Both approaches overcome the PE limitations, but differ in the order of execution of two critical steps.

Approach 1: Synthesize Array \rightarrow Extend Resolution

1. An array of sources with the spacing of $\lambda/2$ is synthesized based on the defined source directivity pattern.

2. Each array element is treated as a monopole source with a weighted source strength. The standard Gaussian monopole representation, centered at each array element, is used to populate intermediate grid points in the starting field, achieving finer PE grid point spacing.

Approach 2: Extend Resolution → Synthesize Array

1. The vertical wave number spectrum is extended beyond the range of propagating wave numbers, defined by the far field directivity pattern, to achieve $\lambda/10$ PE grid point spacing resolution. This extended directional function is constructed using a modified form of zero-padding. The spectrum is windowed to filter out evanescent vertical wave number components and diminish large propagating vertical wave number components.
2. The windowed spectrum is used to synthesize the starting field with PE grid point spacing.

6.2 Array Synthesis for Directional PE Sources

Assuming linear superposition, the total field produced by an array is calculated as the sum of the fields of each individual element in the array. The PE starting field uses a vertical line array composed of monopole elements.

For the N -element linear array shown in Figure 6.1, the acoustic pressure field is calculated as

$$p(R) = \sum_{n=-\ell}^{\ell} \frac{A_n}{R_n} e^{ikR_n} \quad (6.1)$$

where A_n is the aperture function that defines the weighting for each array element, R_n is the distance from each element to the receiver and, for N odd, $\ell = (N-1)/2$.

Assuming that the extent of the array is small compared to the distance between the source and receiver allows the distance from each array element to the receiver to be approximated as $R_n \approx R - n\Delta z \sin \theta$ in the exponential, where Δz is the array element spacing and θ is the elevation angle from the source, and $R_n \approx R$ in the denominator, where $R = R_0$ is the source-receiver distance measured from

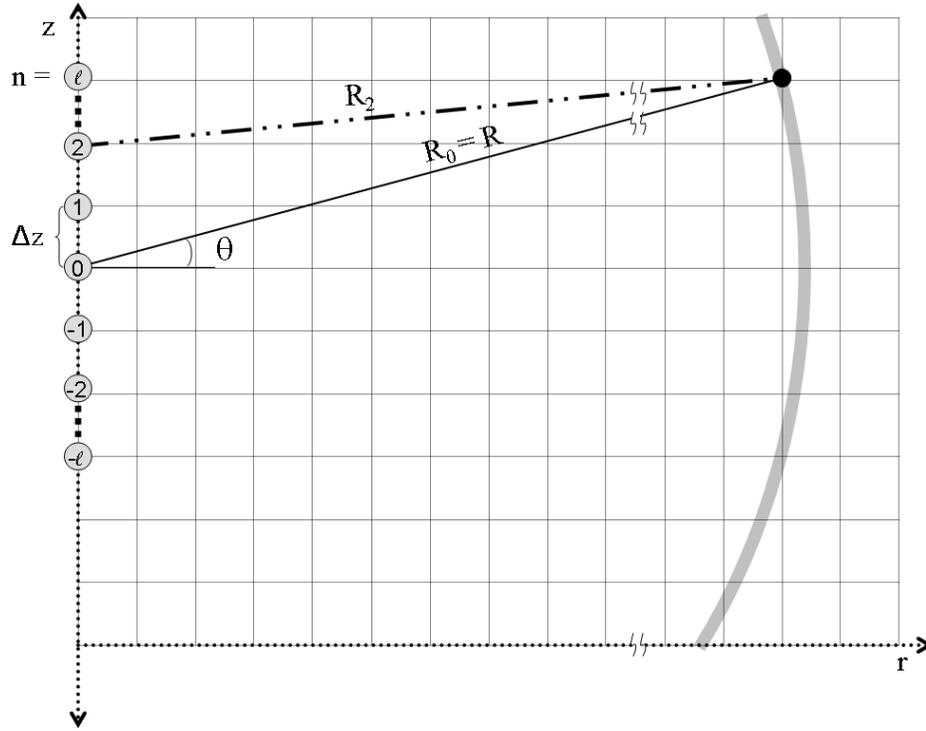


Figure 6.1. Geometry of the starting field grid parameters. The shaded circles represent monopole elements of the source array.

the center of the array

$$p(R, \theta) = \frac{e^{ikR}}{R} \sum_{n=-\ell}^{\ell} A_n e^{-ik(n\Delta z \sin \theta)}. \quad (6.2)$$

The term in front of the summation takes the form of a unit amplitude monopole located at the center of the array. It is referred to as the axial pressure

$$P_{ax}(R) = \frac{e^{ikR}}{R}. \quad (6.3)$$

The summation term is the far field directional function, accounting for the variation of the pressure with angle from the center of the array

$$D(\theta) = \sum_{n=-\ell}^{\ell} A_n e^{-ik(n\Delta z \sin \theta)}. \quad (6.4)$$

Thus, Equation 6.2 can be rewritten

$$p(R, \theta) = P_{ax}(R)D(\theta). \quad (6.5)$$

The vertical orientation and discrete and finite nature of the array place limitations on the vertical wave number components able to be resolved by the array. By the Nyquist criterion, the smallest wavelength that is resolvable by the array in the vertical axis is twice the element spacing $\lambda_{z,\min} = 2\Delta z$. Thus, the largest resolvable vertical wave number is

$$k_{z,\max} = \frac{\pi}{\Delta z}. \quad (6.6)$$

The largest resolvable wavelength is equal to the length of the array $\lambda_{z,\max} = N\Delta z$. Thus, the smallest resolvable wave number is $\Delta k_z = 2\pi/N\Delta z$. The vertical wave number can be discretized as $k_z = m\Delta k_z$ where, for N odd, $m = -(N-1)/2, -(N-1)/2+1, \dots, 0, \dots, (N-1)/2-1, (N-1)/2$. Then, k_z can be written as

$$k_z = k \sin \theta = \frac{2\pi m}{N\Delta z}. \quad (6.7)$$

Substituting Equation 6.7 into Equation 6.4 leads to the far field directional function taking the (shifted) form of a discrete Fourier transform

$$D_m = \sum_{n=-(N-1)/2}^{(N-1)/2} A_n e^{-i2\pi nm/N}. \quad (6.8)$$

The inverse Fourier transform can then be used to find the aperture function A_n , given the discrete far field directional function D_m

$$A_n = \frac{1}{N} \sum_{m=-(N-1)/2}^{(N-1)/2} D_m e^{i2\pi nm/N}. \quad (6.9)$$

6.3 Two Approaches for the Generalized Source Directivity Technique

Section 6.2 lays the groundwork for the two approaches to PE directivity discussed in this section. However, the implications of Equation 6.6 prevent the direct implementation of array synthesis. If a fine vertical PE grid point spacing, for example $\lambda/10$, is chosen, the maximum value of the vertical wavelength will be $k_{z,\max} = 10\pi/\lambda = 5k$. By Equation 6.7 the vertical wave number component for large N will range from approximately $-5k \leq k_z \leq 5k$. Outside the range of $-k \leq k_z \leq k$, the horizontal wave number $k_r = \sqrt{k^2 - k_z^2}$ is an imaginary quantity. These components are evanescent and do not propagate into the far field. They are, therefore, not resolvable in the far field directional function, used as input to Equation 6.9. Consequently, either

1. the propagating wave number range $-k \leq k_z \leq k$ can be used in the array synthesis, forcing the array element spacing to be $\Delta z = \lambda/2$ by Equation 6.6, and requiring a different method of achieving a finer starting field resolution (Approach 1), or
2. the larger vertical wave numbers can be artificially defined in an extended directional function to achieve array synthesis that produces a fine-resolution starting field (Approach 2).

In this section, the results of the proposed approaches for directivity inclusion are compared to analytical results for horizontal and vertical dipole sources both in the free field and at low altitude above a rigid ground. The two different orientations and two different heights of the dipoles are used to test the limits of the methods. While the horizontal dipole emits sound primarily at smaller elevation angles where the PE is expected to perform well, the vertical dipole emits sound primarily at larger elevation angles where PE accuracy is degraded. Similarly, the high altitude source field is unimpeded by a ground surface, while reflection of sound off the ground may degrade results for the low altitude sources. The acoustic pressure inputs for the unit amplitude horizontal and vertical dipole sources are

$$p_h(R, \theta) = \frac{e^{ikR}}{R} \cos \theta \quad (6.10)$$

$$p_v(R, \theta) = \frac{e^{ikR}}{R} \sin \theta, \quad (6.11)$$

respectively, where again θ is the angle from the horizontal.

Figure 6.2 shows the sound levels for (a) horizontal and (b) vertical dipole sources with a frequency of 50 Hz in the free field and a homogeneous atmosphere as calculated with Equations 6.10 and 6.11, respectively. The horizontal dipole results show a single lobe in the range direction, extending horizontally from the source point. The vertical dipole results for the range direction show two half-lobes, extending vertically from the source point.

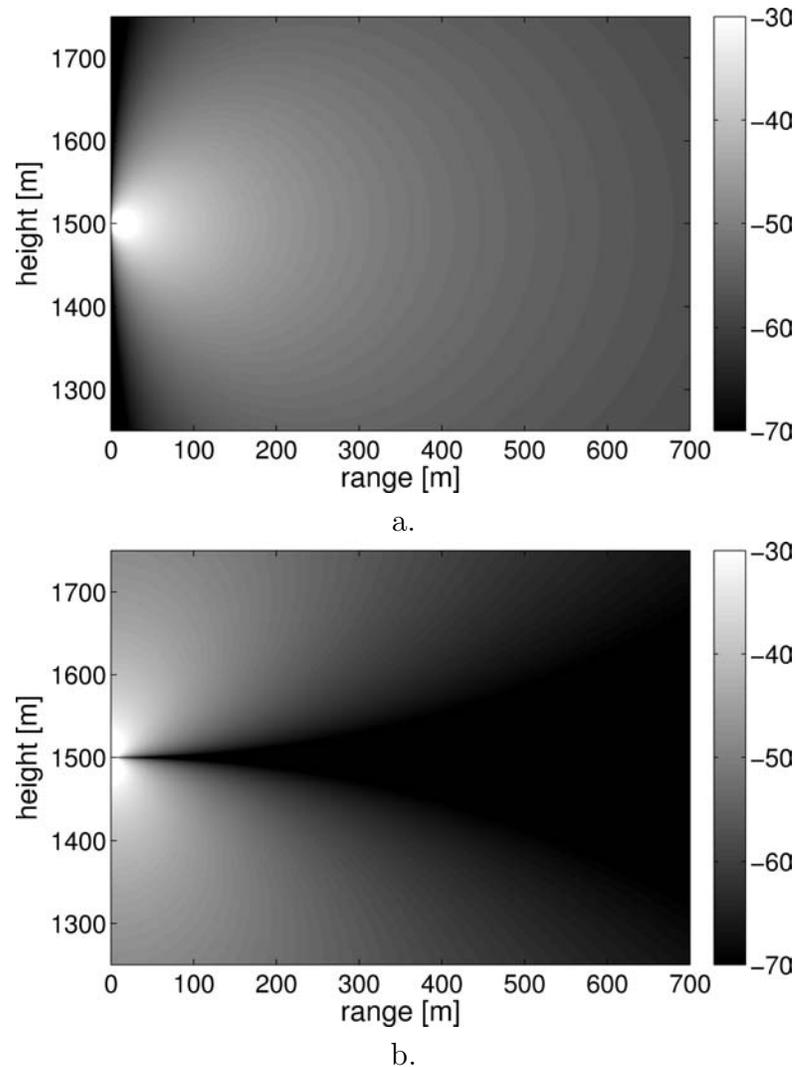


Figure 6.2. Sound field, in dB, produced by a (a) horizontal and (b) vertical dipole in free field, calculated with an analytical model. The source is at 1500 m height.

The analytical solutions for horizontal and vertical dipole sources above a rigid ground are

$$p_h(R, \theta) = \frac{e^{ikR}}{R} \cos \theta + \frac{e^{ikR_{image}}}{R_{image}} \cos(-\theta) \quad (6.12)$$

$$p_v(R, \theta) = \frac{e^{ikR}}{R} \sin \theta + \frac{e^{ikR_{image}}}{R_{image}} \sin(-\theta) \quad (6.13)$$

respectively, where the first term in both equations represents the direct source, the second term represents the image source, and R_{image} is the distance from the image source to the receiver.

Figure 6.3 shows results for a (a) horizontal and (b) vertical dipole 4 m above a rigid ground. Here interference patterns from the interaction of the direct and ground-reflected sound are illustrated. Levels are large near the ground for the horizontal dipole and small for the vertical dipole.

These four conditions will be used as references to test the performance of the two generalized source directivity starting field techniques discussed in the remainder of this chapter.

6.3.1 Approach 1: Synthesize Array \rightarrow Extend Resolution

In the first approach to implementing generalized source directivity, the demands placed on calculating the far field directional function D_m are small. As dictated by its discretization, D_m is defined at linear increments of k_z space and, thus, linear increments of $\sin \theta$ (Equation 6.7). As a consequence, the directional function elements are more densely concentrated around the horizontal, where the elevation angle $\theta = 0$.

Applying Equation 6.9 to the directional function D_m , which encompasses a range of vertical wave numbers $-k \leq k_z \leq k$, yields an aperture function A_n defined for elements of spacing $\Delta z = \lambda/2$, centered at the source height. Each array element represents a monopole source with weighting A_n .

The traditional PE representation for a monopole takes the form of a Gaussian function with the same order of accuracy as the square-root approximation, Equation 2.12. The Gaussian starting field can be derived from the exact expression for a monopole, applying far field and small elevation angle approximations [48, 52]. It is a smooth function, finite at all points, with limited energy in the larger vertical

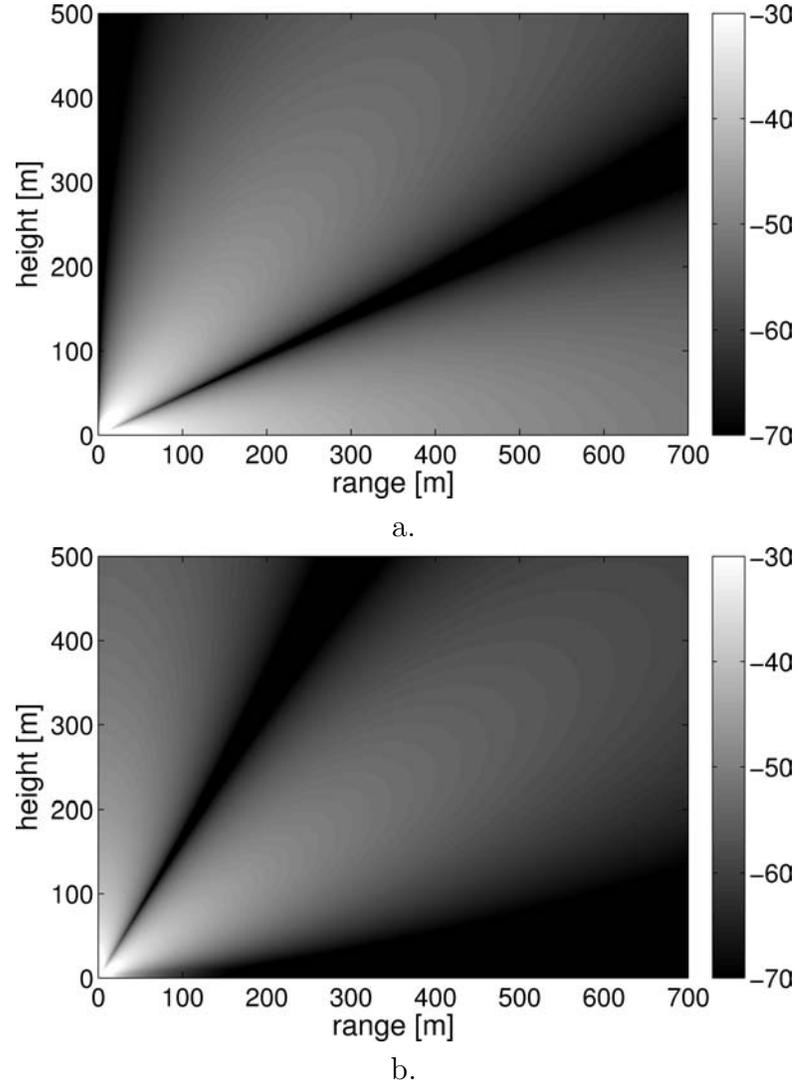


Figure 6.3. Sound field, in dB, produced by a (a) horizontal and (b) vertical dipole 4 m above a rigid ground, calculated with an analytical model.

wave numbers. It, therefore, satisfies the requirements of the PE while serving as an acceptable substitute for the exact monopole expression.

The second-order, wide-angle Gaussian start field used in this research is [1]

$$q(0, z') = \sqrt{ik_a} (A_0 + A_2 k_a^2 (z' - z_s)^2) \exp\left(-\frac{k_a^2 (z' - z_s)^2}{B}\right) \quad (6.14)$$

where z' is the vector of vertical PE grid points, z_s the source height, $A_0 = 1.3717$, $A_2 = -0.3701$, and $B = 3$. Because an amplitude weight in pressure corresponds

to the same amplitude weight in the PE propagated quantity q , the total starting field can be constructed as the sum of N Gaussian functions,

$$q_{tot}(0, z') = \sum_{n=-(N-1)/2}^{(N-1)/2} A_n \sqrt{ik_a} (A_0 + A_2 k_a^2 (z' - z'_n)^2) \exp\left(-\frac{k_a^2 (z' - z'_n)^2}{B}\right) \quad (6.15)$$

where z'_n is the vector of array element heights, with $\lambda/2$ spacing.

6.3.1.1 Numerical Considerations for Approach 1

Approach 1 attempts to avoid evanescent wave numbers by employing, and compensating for, wider-spaced array elements. However, increasing the resolution of the starting field with the Gaussian function (Equation 6.14), by nature, extends the wave number spectrum to include evanescent wave numbers, by Equation 6.7. The effect of the Gaussian functions on wave number spectrum must, therefore, be investigated to ensure there is no significant impact on the propagated sound.

Figure 6.4 shows the PE results using Approach 1, with $N = 1025$, for the (a) horizontal and (b) vertical dipole sources under the same conditions as Figure 6.2. The vertical grid point spacing is 0.61 m, slightly smaller than $\lambda/10$ for a sound speed of 343 m/s. The source is positioned high above a very soft ground (an effective flow resistivity of 10 cgs rayls was used with the Delany and Bazley calculation method), so that it may be considered to be in the free field. Therefore, the image source is neglected.

While results at small elevation angles from the source are reasonable for the horizontal dipole, the behavior at larger elevation angles is inaccurate. These results, when compared to Figure 6.2, show two additional lobes formed at large elevation angles above and below the source. The absolute difference between the levels of the PE results using Approach 1 and the analytical results is shown in Figure 6.5. Only absolute differences in the range of 0 to 1 dB are differentiated in order to show the smaller disparities with adequate contrast. All absolute differences of 1 dB and above are shown in white. The figure shows that not only are the levels at the large elevation angles inaccurate, as might be expected for a PE method, but the sound emitted at large elevation angles causes reduced accuracy in the smaller elevation angles at larger ranges. This is seen in Figure 6.5a as a

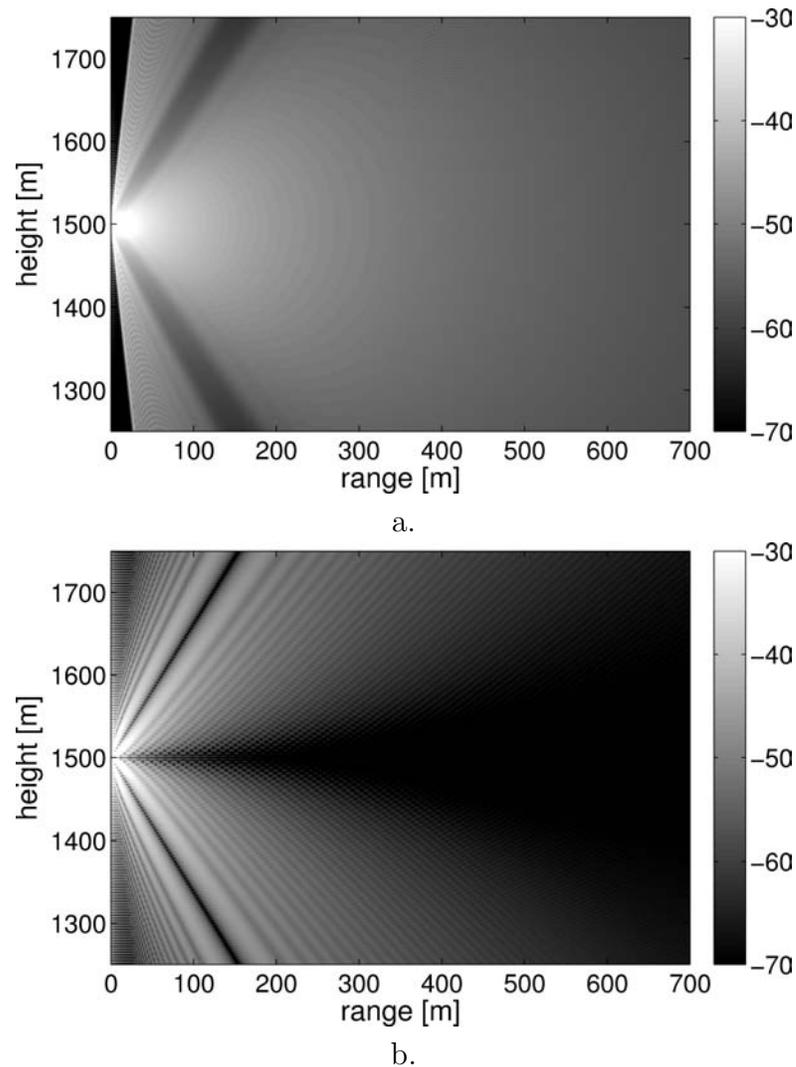


Figure 6.4. Sound field, in dB, produced by a (a) horizontal and (b) vertical dipole high above a soft ground, calculated with Approach 1. The source is at a height of 1500 m.

sudden onset of degradation that begins at a range of about 300 m and persists for the remainder of the propagation range. The reduction in accuracy is caused by the reflection of the spurious sound emitted at large elevation angles off the distant, soft ground. The reflected sound eventually reemerges in the region of small elevation angles from the source, beginning at a removed distance from the source.

The results of Approach 1 for the vertical dipole also show deviation from the analytical results. Here, an erroneous, striated pattern develops. Although the

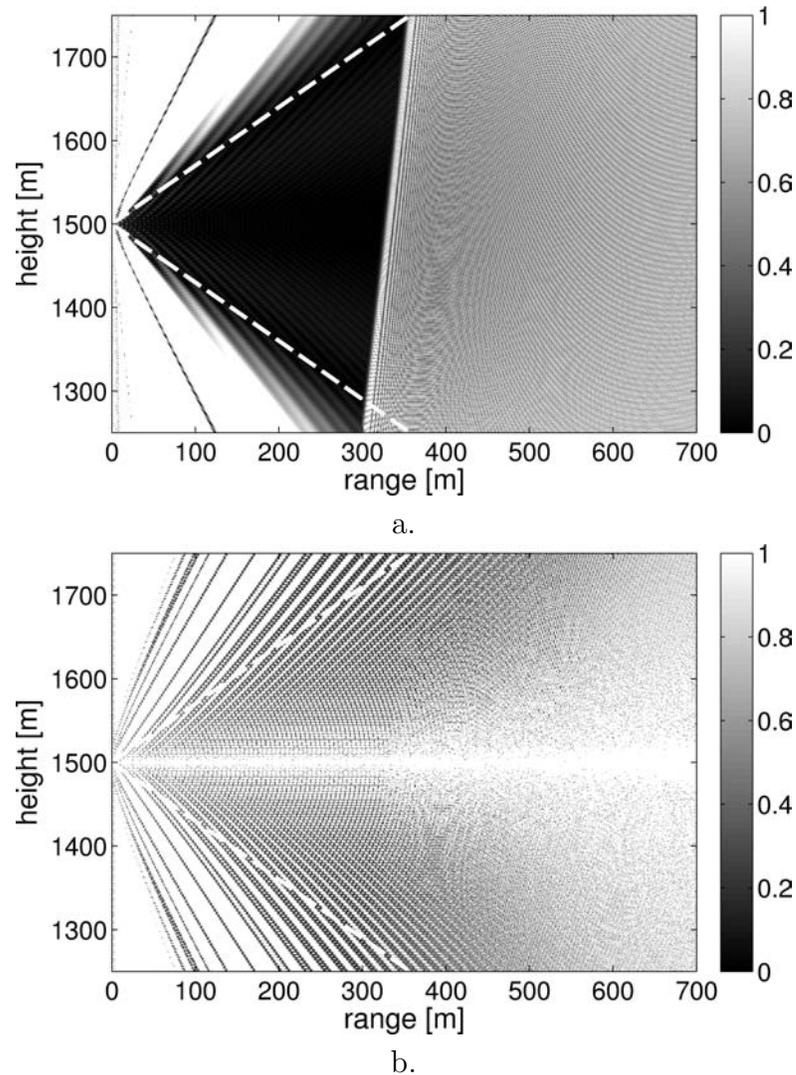


Figure 6.5. Absolute difference in the sound fields of a free field (a) horizontal and (b) vertical dipole in dB, calculated with Approach 1 and the analytical model. The source is at a height of 1500 m. The dashed line indicates the elevation angles of $\pm 35^\circ$, within which the PE results are expected to be valid. All absolute differences of 1 dB and larger are shown in white.

accuracy deteriorates with larger elevation angles, Figure 6.5 shows that the error is greater than 0.5 dB at all angles.

These results indicate that the placement of the Gaussian functions at increments of $\lambda/2$ and subsequent extension of the represented angular spectrum to evanescent wave numbers influences the propagating wave number content of the starting field. The directional function that is transformed in the synthesis of the

array is untreated and, thus, may not go to zero at $\theta = \pm\pi/2$. A discontinuity between the values of the directional function at the two extreme values $\theta = \pm\pi/2$ would cause ringing in the spatial domain, as is shown for the vertical dipole source.

Additionally, energy at $k_z = 2k$ is introduced by the periodicity of the Gaussian placement every one-half wavelength. Because $k_z = 2k$ likely does not fall on-bin, spill-over to surrounding bins can occur. Such additions to the wave number spectrum violate the PE requirement that sound not be emitted at large vertical wave numbers. The impact is seen in the PE run results.

In order to decrease the additional angular spectral content of the Gaussian array starting field (Equation 6.15), a window can be applied using the following process:

1. The starting field is transformed with Fourier techniques to obtain the wave number spectrum.
2. An exponential window, used in holographic processing, is applied to filter out larger wave number content [71]

$$\Pi = \begin{cases} 1 - \frac{1}{2}e^{-(1-|k_z|/k)/\alpha} & |k_z| < k \\ \frac{1}{2}e^{(1-|k_z|/k)/\alpha} & |k_z| > k \end{cases} \quad (6.16)$$

where α determines how quickly the window decays.

3. An inverse discrete Fourier transform is taken to return the new PE starting field.

6.3.1.2 Numerical Examples with Modified Approach 1

Figure 6.6 shows the absolute difference between the PE results using the modified Approach 1 and the analytical model for the (a) horizontal and (b) vertical dipole sources under the same conditions as Figure 6.5. The window decay rate is $\alpha = 0.1$ and the source height is 1500 m. Again, all absolute differences of 1 dB and larger are shown in white.

The additional lobes at large elevation angles from the source in the horizontal dipole results have been removed by the window. The absence of the lobes

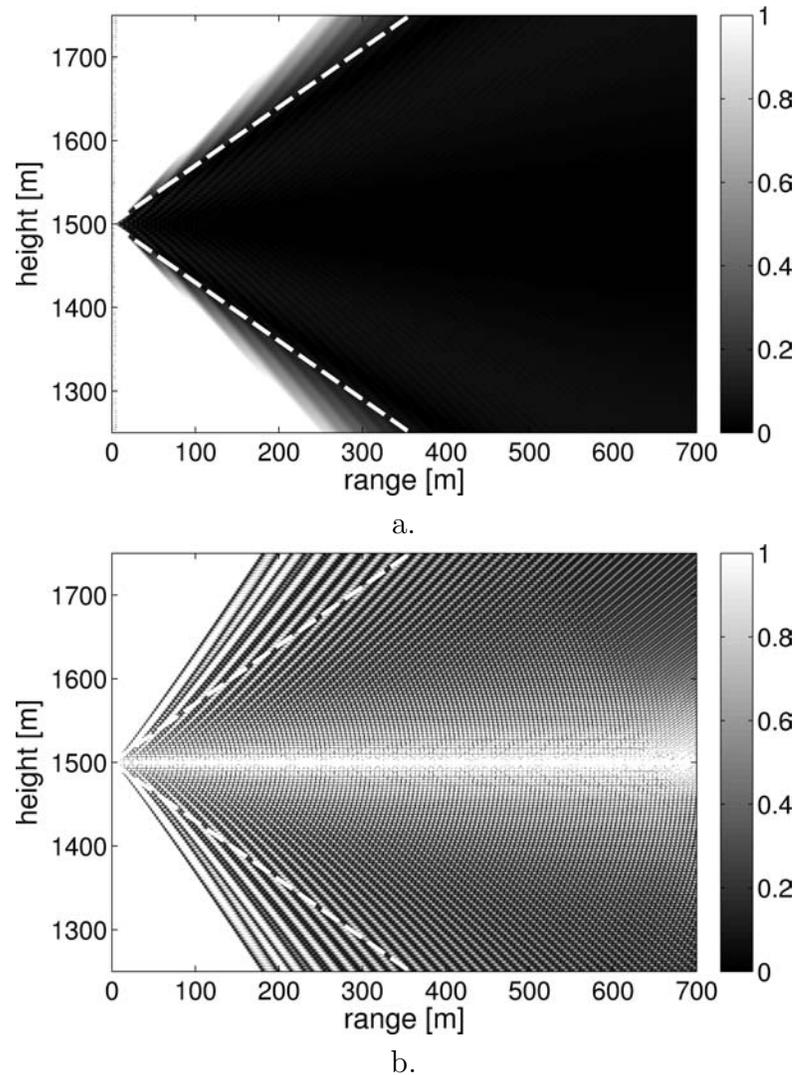


Figure 6.6. Absolute difference in the sound fields of a free field (a) horizontal and (b) vertical dipole in dB, calculated with modified Approach 1 and the analytical model. The source is at a height of 1500 m. The dashed line indicates the elevation angles of $\pm 35^\circ$, within which the PE results are expected to be valid. All absolute differences of 1 dB and larger are shown in white.

eliminates the erroneous reflections off the ground of the sound coming from these lobes.

Striations still appear in the vertical dipole results, likely because the exponential window function does not force the angular spectrum to zero at $\theta = \pm\pi/2$, but rather to half its original value. However, the overall error has improved to below approximately 0.5 dB within most of the valid PE angle range.

Figure 6.7 shows the absolute difference between the levels of the PE results using the modified Approach 1 and the analytical model for sources at a height of 4 m above a rigid ground.

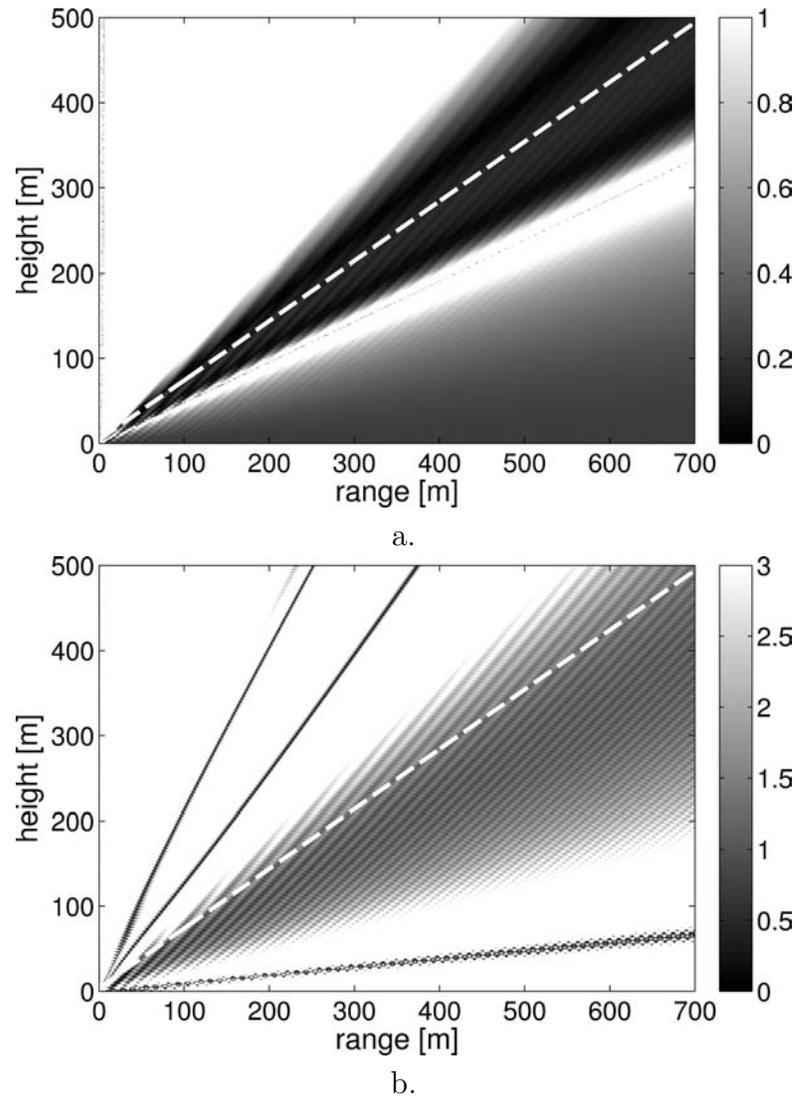


Figure 6.7. Absolute difference in the sound fields of (a) horizontal and (b) vertical dipole 4 m above a rigid ground, in dB, calculated with modified Approach 1 and the analytical model. The dashed line indicates the elevation angles of $\pm 35^\circ$, within which the PE results are expected to be valid. All absolute differences of 1 dB and larger in (a) and 3 dB and larger in (b) are shown in white.

The absolute difference between the PE and analytical model for the horizontal dipole is well under 1 dB for the smaller elevation angles. The error increases in the first interference minimum where the level is quite low, but regains accuracy

entering the second region of constructive interference.

The accuracy of the vertical dipole results is significantly degraded. The errors are high near the ground where the levels are small. However, even in the region of constructive interference, the error is approximately 2 dB (note the change of grayscale limit).

The modified Approach 1 performs well for a horizontal dipole source in both the free field and at low altitudes over a rigid ground. However, it is less accurate for the vertical dipole source, especially when ground effects must be included.

6.3.2 Approach 2: Extend Resolution \rightarrow Synthesize Array

In Approach 2, the problem of the fine starting field resolution is tackled immediately. Acknowledging the inevitable extension of the wave number spectrum to include evanescent modes, this approach shapes the final wave number spectrum of the starting field at the outset.

6.3.2.1 Defining the Directional Function

Approach 2 begins by determining the final wave number spectrum range, using the PE vertical grid point spacing Δz as input in Equation 6.6. (A spacing of $\lambda/10$, yielding a wave number spectrum range of $-5k \leq k_z \leq 5k$, is used in this discussion for descriptive ease; however, a different spacing may be used so long as it complies with PE requirements.) The extended directional function is then computed in three steps:

Step 1. The far field directional function is assigned to the extended directional function within the range $-k \leq k_z \leq k$. The extended directional function is set to zero at the evanescent wave numbers.

The top plots (Step 1 plots) of Figures 6.8a and 6.8b show the directivity patterns plotted as functions of the normalized vertical wave number k_z/k , where k is the acoustic wave number, assumed to be constant over height, for a (a) horizontal and (b) vertical dipole, respectively, with $N = 57$. The envelope of the function is relatively smooth for the horizontal dipole, but exhibits extreme discontinuities for the vertical dipole. A discontinuity in

wave number space will cause contamination of the aperture function over all array elements in space. The discontinuity, therefore, must be eliminated.

Step 2. Anticipating that Step 3 will be used to diminish energy at the evanescent wave numbers, the directional function at the evanescent wave numbers is filled with the value at the nearest propagating wave number, to avoid discontinuities.

The middle plots (Step 2 plots) of Figure 6.8 show the modified directional function for the horizontal and vertical dipoles. Because the next step in the approach will remove most of the added energy, the method of removing the discontinuity between propagating and evanescent wave numbers is of little importance. It is most important that the discontinuity be removed.

Step 3. The energy at evanescent wave numbers is restricted using the exponential window defined by Equation 6.16.

The bottom plots (Step 3 plots) of Figure 6.8 show the windowed functions, with $\alpha = 0.1$. The original functions and their windowed counterparts are similar. However, the windowed functions have been smoothed. With this process, the windowed directional functions have sufficient spectral range to represent a fine resolution PE starting field while minimizing evanescent and large propagating vertical wave number components.

6.3.2.2 Calculating the Starting Field

With the extension of the vertical wave number spectrum, a standard inverse Fourier transform is taken to obtain the aperture function A_n at the PE resolution in space (Equation 6.9). However, to calculate the PE starting function, the aperture function, derived using the quantity of acoustic pressure, must be converted to the propagated quantity of the PE, q . The relationship between the pressure and this quantity $q = p\sqrt{r}$ cannot be used at the starting field range $r = 0$. Therefore, a relationship employing the far field approximation as r approaches 0 is used.

The derivation begins with the one-way form of the Helmholtz equation, Equa-

tion 2.7, re-written as

$$\frac{\partial q}{\partial r} - ik \left(\sqrt{\frac{1}{k^2} \frac{\partial^2}{\partial z^2} + 1} \right) q = 0. \quad (6.17)$$

A Fourier transform pair is defined to transform Equation 6.17 into the angular spectrum domain

$$Q(r, u) = \mathcal{F}\{q(r, z)\} = \int_{-\infty}^{\infty} q(r, z) e^{-i2\pi uz} dz \quad (6.18)$$

$$q(r, z) = \mathcal{F}^{-1}\{Q(r, u)\} = \int_{-\infty}^{\infty} Q(r, u) e^{i2\pi uz} du. \quad (6.19)$$

The second-order partial derivative with respect to z and the partial derivative with respect to r for a regular function $q(r, z)$ can, therefore, be rewritten as

$$\mathcal{F}\left\{\frac{\partial^2 q}{\partial z^2}\right\} = -4\pi^2 u^2 \mathcal{F}\{q\} \quad (6.20)$$

$$\mathcal{F}\left\{\frac{\partial q}{\partial r}\right\} = \frac{\partial(\mathcal{F}\{q\})}{\partial r}, \quad (6.21)$$

respectively. Using Equations 6.20 and 6.21, the Fourier transform of Equation 6.17 is

$$\frac{\partial Q}{\partial r} - ikMQ = 0 \quad (6.22)$$

where $MQ = \sqrt{1 - \frac{4\pi^2 u^2}{k^2}} Q$. The solution to Equation 6.22 is given by

$$Q(r, u) = Q(0, u) e^{ikMr}. \quad (6.23)$$

An inverse transform can be performed to express the field $q(r, z)$ in terms of the starting field $q(0, z)$

$$q(r, z) = \mathcal{F}^{-1}\{\mathcal{F}\{q(0, z)\} e^{ikMr}\}. \quad (6.24)$$

Looking ahead to the use of the convolution theorem to represent this equation in integral form, the inverse Fourier transform of the propagator e^{ikMr} can be

expressed in terms of a Hankel function as [38]

$$\mathcal{F}^{-1}\{e^{ikMr}\} = \frac{ikr}{2\sqrt{r^2+z^2}}H_1^{(1)}(k\sqrt{r^2+z^2}) \quad (6.25)$$

where $H_1^{(1)}$ is the Hankel function of the first kind of order 1. The asymptotic expansion of this Hankel function for large arguments is

$$H_1^{(1)}(k\rho) \approx \sqrt{\frac{2}{\pi k\rho}}e^{ik\rho-i3\pi/4}. \quad (6.26)$$

This equation can be related to Equation 6.25 with the variable definition $\rho(z) = \sqrt{r^2+z^2}$ and Equation 6.25 can be rewritten [38]

$$\mathcal{F}^{-1}\{e^{ikMr}\} = \frac{ikr}{2\sqrt{r^2+z^2}}H_1^{(1)}(k\sqrt{r^2+z^2}) \approx \frac{r}{\rho(z)^{3/2}}\sqrt{\frac{1}{i\lambda}}e^{ik\rho(z)}. \quad (6.27)$$

Multiplication in the angular spectrum domain corresponds to convolution in the spatial domain. Therefore, using the convolution properties $f(x) * g(x) = \int_{-\infty}^{\infty} f(x-x')g(x')dx'$ for the corresponding Fourier transform equation $\mathcal{F}_x^{-1}[F(u)G(u)] = f(x) * g(x)$ [71], Equation 6.24 can be transformed as

$$q(r, z) = q(0, z) * \mathcal{F}^{-1}\{e^{ikMr}\} = \int_{-\infty}^{\infty} \sqrt{\frac{1}{i\lambda}}q(0, z')\frac{r}{\rho'(z-z')^{3/2}}e^{ik\rho'(z-z')}dz' \quad (6.28)$$

where $\rho'(z-z') = \sqrt{r^2+(z-z')^2}$ and $z-z'$ is interpreted as a vector quantity of the height difference between the receiver and all points along the source array.

Plugging Equation 6.28 into the original relationship $q = p\sqrt{r}$, the pressure is represented as

$$p(r, z) = \frac{q(r, z)}{\sqrt{r}} = \int_{-\infty}^{\infty} \sqrt{\frac{1}{i\lambda}}q(0, z')\frac{\sqrt{r}}{\rho'(z-z')^{3/2}}e^{ik\rho'(z-z')}dz'. \quad (6.29)$$

Assuming that the extent of the array is small compared to the distance between the array elements and receiver, $|z'-z_s| \ll \rho'(z-z')$, where z_s is the source height, the receiver range can be approximated as $r \approx \rho'(z-z')\cos\theta$, where θ is the elevation angle from the source. Using the trigonometric identity $\cos\theta =$

$\sqrt{1 - \sin^2 \theta}$ and a small angle approximation that preserves the 2nd order accuracy,

$$\sqrt{\cos \theta} = \sqrt[4]{1 - \sin^2 \theta} \approx 1 - \frac{\sin^2 \theta}{4}, \quad (6.30)$$

Equation 6.29 becomes

$$p(r, z) = \int_{-\infty}^{\infty} \sqrt{\frac{1}{i\lambda}} \left(1 - \frac{\sin^2 \theta}{4}\right) q(0, z') \frac{e^{ik\rho'(z-z')}}{\rho'(z-z')} dz'. \quad (6.31)$$

Using a method similar to Equation 6.1 for a continuous line source, an equation for the pressure can also be written

$$p(r, z) = \int_{-\infty}^{\infty} \frac{B(z')}{\rho'(z-z')} e^{ik\rho'(z-z')} dz'. \quad (6.32)$$

By comparing Equations 6.31 and 6.32, a relationship between $q(0, z')$ and the line source weightings $B(z')$ is found

$$q(0, z') = \sqrt{i\lambda} \frac{B(z')}{1 - \sin^2 \theta/4}. \quad (6.33)$$

Finally, it is recognized that a discretized form of the coefficients $B(z')$ can be related to the discretized aperture function as $A_n = B_n \Delta z$. Therefore, the discrete, N -element PE starting field can be calculated from the aperture function as

$$q(0, z'_n) = \frac{\sqrt{i\lambda}}{\Delta z} \frac{A_n}{1 - \sin^2 \theta/4}. \quad (6.34)$$

Note: because the elevation angle θ is not defined in the starting field, the factor $1/(1 - \sin^2 \theta/4)$ must be applied during the definition of the directional function, before the inverse transform is performed.

Using the method described in Section 6.3.2.1, in conjunction with Equation 6.34, the PE starting field can be calculated with an inverse Fourier transform of the extended far field directional function. Thus, the calculation procedure of Approach 2 is complete.

6.3.2.3 Numerical Examples with Approach 2

Figure 6.9 shows the absolute difference between the levels of the PE results using Approach 2 with window decay rate $\alpha = 0.1$ and the analytical model for a (a) horizontal and (b) vertical dipole, respectively, under the same conditions as Figure 6.6. The source is at a height of 1500 m and is assumed to be in the free field.

The absolute difference between the PE and analytical models is well under 1 dB for all angles in the valid PE range for the horizontal dipole. The region of accuracy stretches further into high elevation angles than does that of the modified Approach 1.

The vertical dipole results also agree with the analytical results to within 1 dB for most angles inside the valid elevation angle range, a marked improvement over the results from the modified Approach 1. However, accuracy falls off quickly outside the valid elevation angle region. Nonetheless, accurate results using the PE are achieved, despite sound being emitted at larger elevation angles.

Figure 6.10 shows the absolute difference between the levels of the PE results using Approach 2 and the analytical model for a (a) horizontal and (b) vertical dipole, respectively, under the same conditions as Figure 6.7. The sources are at a height of 4 m above a rigid ground.

The absolute difference between the PE and analytical models for the horizontal dipole is well under 1 dB for the smaller elevation angles, showing an improvement over the modified Approach 1. However, large errors begin at the first region of destructive interference and remain large in the next region of constructive interference, where modified Approach 1 recovers. Thus, accuracy to the full range of elevation angles $\pm 35^\circ$ is not achieved.

The vertical dipole results agree with the analytical results to within 1 dB for all angles in the valid PE range. Accuracy is achieved even near the ground where an interference dip exists and sound levels are low, and extends well into the region of constructive interference where the levels are high.

The model agreement demonstrates that the dipole directivity for a low altitude source over a rigid ground surface can be accurately represented by this starting field calculation method, with image source inclusion for regions of high sound levels. It may or may not be accurate at or after regions of destructive interference.

6.4 Comparison of Approaches 1 and 2

A comparison of propagation results from horizontal and vertical dipole sources in the free field and at low altitude above a rigid ground indicate that both approaches generally perform well within the valid PE angle range. Results were mostly within 1 dB of analytical results, and often much closer.

The two approaches use different strategies to accomplish the objective of generalized source directivity representation. Approach 1 remains in touch with the accepted Gaussian start field for a monopole, which limits energy at higher wave numbers. It tries to avoid evanescent wave numbers by using a coarse resolution aperture function. However, in extending starting field resolution with the Gaussian array, energy is reintroduced into the evanescent wave numbers that must later be windowed out. Approach 2 deviates from the standard Gaussian starting field. However, it addresses the problem of evanescent wave numbers directly and no later fix is required.

Because Approach 2 shows a slight advantage in accuracy and consistency and, needing no modification, is more straightforward in implementation, it is identified as the preferred generalized source directivity starting field technique. Motivated by efforts to model aviation noise in communities surrounding airports, the most relevant results are those near the ground. Here, Approach 2 shows excellent accuracy for both the horizontal and vertical sources.

Figure 6.11 plots the sound pressure level relative to spherical spreading for three source heights of 4, 60, and 200 m over a rigid ground with a receiver height of 1.22 m. Results calculated with Approach 2 are shown in gray and analytical results are shown in black. The transitions to valid elevation angle ranges occur at 4, 85, and 284 m for the 4, 60, and 200 m high sources, respectively.

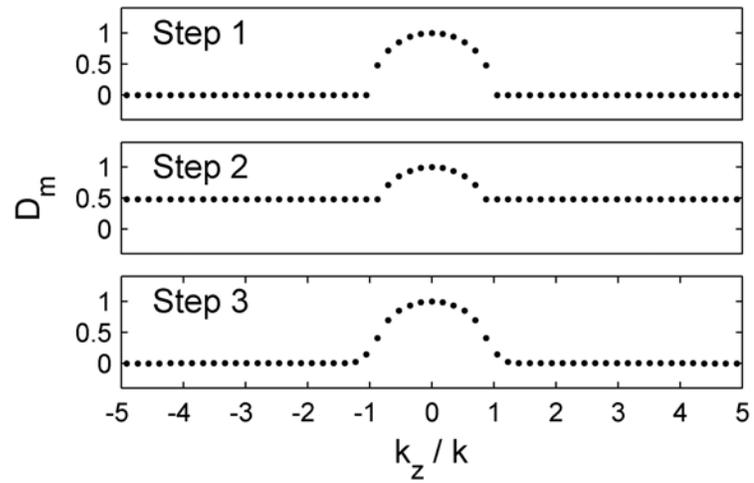
The figures show quick convergence of the PE results to the analytical results for both the horizontal and vertical dipole sources at all three heights for receivers near the ground. By the range of transition to valid elevation angles, results for both the horizontal and vertical dipoles at all heights show errors of less than 1 dB. Accuracy of propagation increases at longer ranges.

6.5 Conclusions

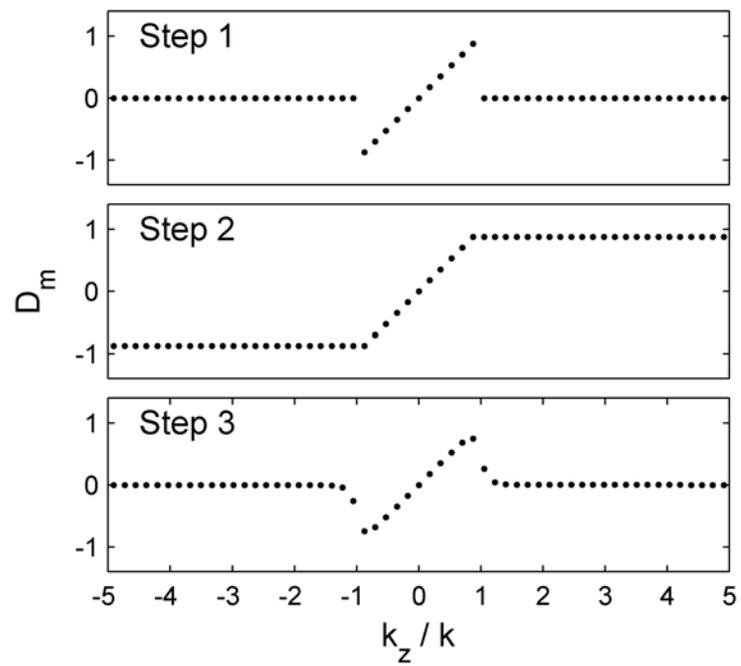
Two approaches were investigated for generalized source directivity representation. Both approaches use Fourier techniques to transform between the directivity pattern and weightings of starting field elements. To overcome the PE limitations that require a finite, high resolution starting field, Approach 1 synthesizes an array with coarse element spacing, and increased starting field resolution by populating intermediate grid points with Gaussian functions. It requires modification to eliminate evanescent components inadvertently added by the Gaussian array. Approach 2 extends the angular spectrum to include evanescent wave numbers and directly synthesizes an array of finer element spacing.

While both approaches proved to be generally accurate to within 1 dB, Approach 2 achieves slightly better accuracy, particularly near the ground, where it is most important that results be correct for community noise concerns. It also offers a simpler implementation procedure. Therefore, it is identified as the preferred generalized source directivity starting field technique.

Results for horizontal and vertical dipole directivities are discussed in this chapter. However, it is a straightforward process to replace the dipole directivity with an arbitrary, user-specified profile. The examples used were not meant to closely resemble realistic directivity patterns of commercial aviation noise sources, which tend to show less severe directivity than the dipoles, but rather to test the limits of the model—the horizontal dipole expected to show high accuracy, and the vertical dipole expected to challenge the capabilities of the PE. Because the method performed well for both sources, with and without ground reflections, it is expected to be accurate for more realistic aviation noise directivity profiles, so long as variation with elevation angle is not extreme.



a.



b.

Figure 6.8. Process of directional function calculation for Approach 2 for (a) horizontal and (b) vertical dipole sources.

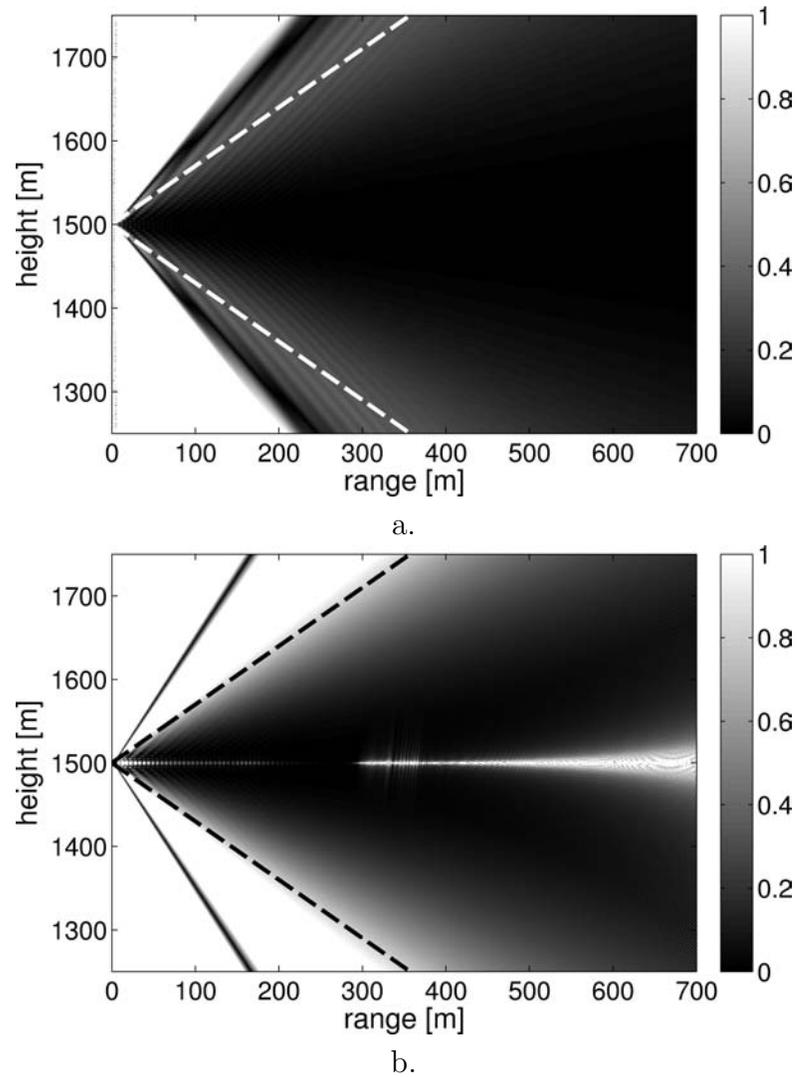


Figure 6.9. Absolute difference in the sound fields of a free field (a) horizontal and (b) vertical dipole in dB, calculated with Approach 2 and the analytical model. The source is at a height of 1500 m. The dashed line indicates the elevation angles of $\pm 35^\circ$, within which the PE results are expected to be valid. All absolute differences of 1 dB and larger are shown in white.

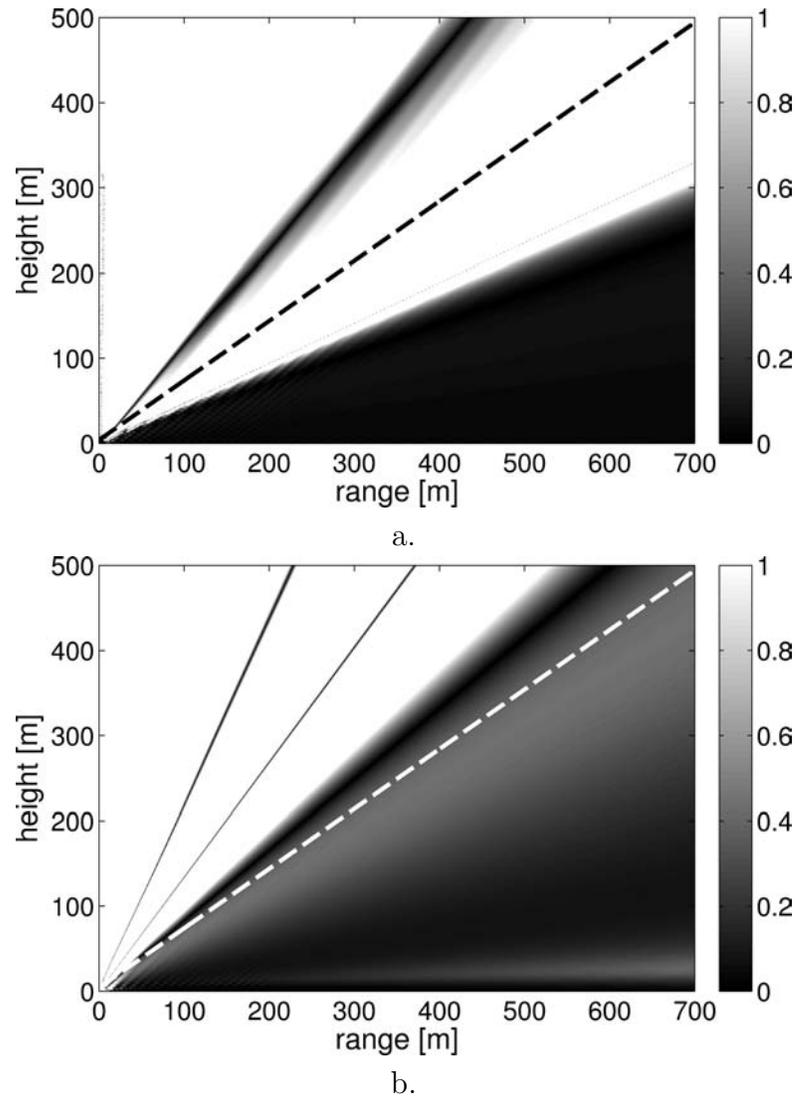
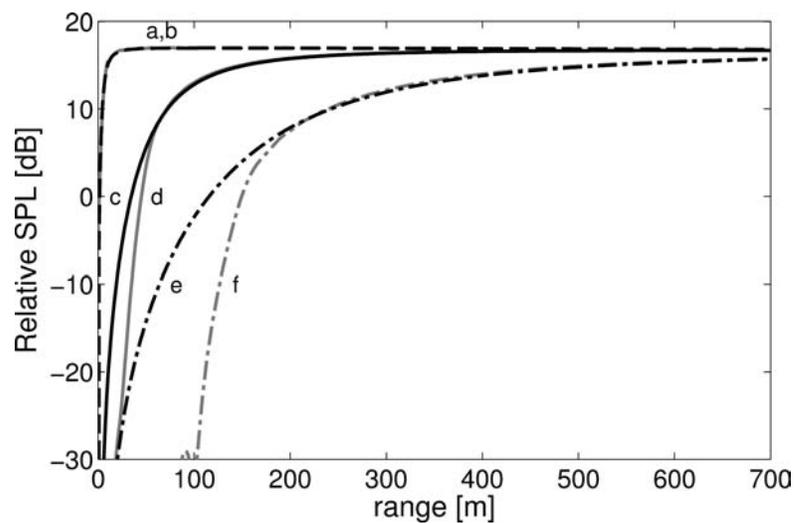
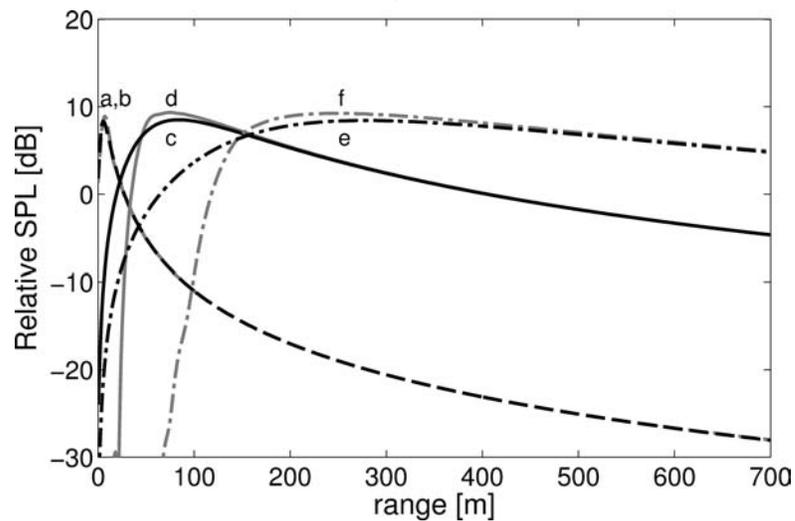


Figure 6.10. Absolute difference in the sound fields of (a) horizontal and (b) vertical dipole 4 m above a rigid ground, in dB, calculated with Approach 2 and the analytical model. The dashed line indicates the elevation angles of $\pm 35^\circ$, within which the PE results are expected to be valid. All absolute differences of 1 dB and larger are shown in white.



a.



b.

Figure 6.11. Sound pressure level relative to spherical spreading, in dB, at a receiver height of 1.22 m for (a) horizontal and (b) vertical dipoles over a rigid ground. Lines *a* and *b*, *c* and *d*, and *e* and *f* represent sources at heights of 4, 60, and 200 m, respectively. Lines *a*, *c*, *e* and *b*, *d*, *f* are calculated with an analytical model and Approach 2, respectively. Lines *a* and *b* are largely indistinguishable in both figures.

Conclusion

7.1 Summary

This dissertation presents a model designed to accurately predict aviation noise levels under complicated propagation conditions, particularly at low frequencies. The model was constructed as a composite of the parabolic equation and fast field program methods. The PE allows the hybrid model to incorporate range-dependent effects at smaller elevation angles from the source. The FFP ensures that the model is accurate at low frequencies to moderate elevation angles, for cases without extreme range-dependent changes near the source. The resulting model can include terrain, ground impedance, meteorology, and directional source effects in noise levels presented in a standard aviation noise contour map.

The contributions of this research can be summarized as follows:

Progress in hybrid modeling

PE and FFP methods combined. Taking a hybrid approach utilizes the strengths of complementary numerical models. In this research, the PE and FFP, discussed in Chapters 2 and 3, respectively, are combined to increase the effectiveness of the propagation model as a whole. The construction of the hybrid model, discussed in Chapter 4, begins by joining the PE and FFP components in the appropriate regions in the two-dimensional vertical plane. PE results are used in elevation angles of 35° and below, and FFP results are used above 35° . The performance of both individual PE and FFP models

was verified against both analytical solutions and published results. Merging them into a hybrid model was validated by running test cases that explored results of different source heights, source frequencies, and propagation conditions.

Methodology developed for extending a two-dimensional model to a pseudo-three-dimensional form. The standard two-dimensional formulations of the PE and FFP methods propagate sound from a point source in a vertical plane. To run correctly, they require knowledge of the terrain profile, ground impedance, and meteorological conditions in that vertical plane. However, available data describing propagation environments may be stored in formats (e.g., terrain elevation or ground impedance raster files) that are inconsistent with these input needs of the propagation model. Consequently, the data must be interpolated into usable forms for the model.

In the two-dimensional hybrid PE-FFP model developed for this research, an extension is made to a pseudo-three dimensional form, discussed in Chapter 5, to better represent aviation noise in communities. The extended model calculates propagation results on multiple polar coordinate systems, centered at each source point along a flight path. These results are then transformed to a Cartesian coordinate system grid to be combined into a cumulative sound exposure of the full flight path. However, in using the propagation model in polar coordinate systems, the model must access terrain, ground impedance, and meteorological inputs defined along these systems. Therefore, a method of interpolation between coordinate systems was established to navigate the gap between Cartesian grid-defined data and the hybrid model polar coordinate input form. Thus, a pseudo-three-dimensional representation of aviation noise from a full flight path is achieved.

The progress in hybrid modeling is understood as the combination of different facets of prior research and propagation techniques into a new aviation noise contouring model. The synthesis of the full model provides an advanced and versatile engineering tool.

Advancement in characterizing a realistic aviation noise source for input into the PE model

Two methods of representing a directional source based on array processing techniques were developed for the unique form of the PE starting field, discussed in Chapter 6. Both approaches recognize the elevation angle limitation of the PE method and the required resolution of vertical grid points in the PE starting field. Approach 1 calculates an aperture function with the standard half-wavelength array element spacing and uses the traditional Gaussian PE representation of a monopole to populate intermediate grid points and increase resolution. Approach 2 anticipates the starting field resolution and extends the vertical wave number spectrum of the directional function to allow use of straightforward inverse Fourier transform techniques. Both approaches produce accurate results within the valid PE elevation angle range for horizontal and vertical dipole examples. However, Approach 2 is more direct and slightly more accurate. Therefore, it is identified as the preferred generalized source directivity starting field technique.

These advancements constitute building blocks in developing accurate aviation noise propagation models. They were used in this research to create a self-contained, inclusive model. However, they can also be applied separately for other propagation applications. For example, the hybrid model alone can be used without employing the directional source option, or the directional source capability can be utilized in a standard PE model. Either option can also be used for simple, range-independent conditions for which interpolation of terrain or impedance data is unnecessary.

Further, the model developed for this research was designed to accept a variety of user-specified option combinations and was programmed to be modular. By choosing different user-specified inputs in the input text file, the model can be directed, for example, to return a single vertical sound field at one chosen angle from a source, or to return a full, horizontal noise contour map. In addition, either a pure PE or pure FFP model can be called, instead of the hybrid model option, for the purposes of comparison. The ability to adapt to the user's chosen application extends the versatility of the model.

7.2 Future Work

While there are numerous conditions for which the hybrid PE-FFP model results are anticipated to be accurate, scenarios involving, for example, a flight path traveling directly over complicated topography or ground impedance transitions challenge the limitations of the model. In such cases, the range-dependence limitation of the FFP may produce errors at larger elevation angles from the source. The extent of the errors should be assessed if such areas are of particular concern, with special consideration given to conditions where range-dependent effects are expected to be significant. Further, if a discontinuity between the results of the two models is found, more extensive investigation into the difference between the results may be performed. Methods that limit the errors in this large elevation angle region, potentially ray-trace and higher-order PE methods, could also be explored.

In future work, a continued investigation of the sensitivities of the hybrid model is also encouraged. Chapter 4, for example, introduced the significant effect of terrain shape definition. It was shown that, though the changes in a particular terrain definition may be small, large differences in sound level can emerge. In this discussion, results using a simple, straight line segment terrain representation were compared to a terrain profile that had been smoothed. Differences were most noticeable near locations where the terrain profile had been altered. However, in some conditions, differences were also seen at ranges well beyond the end of the terrain feature. Consequently, care should be taken to avoid inattentive or imprecise application of the model, especially when the cause of sensitivities occurs at locations of particular interest.

Another possible improvement to the hybrid model is the addition of a directional source capability to the FFP module. Models exist for calculating propagation from dipole [32] and quadrupole [33] sources with the FFP method. A multipole expansion could determine the contribution of the monopole, dipole, and quadrupole components of a particular source in order to fully incorporate directional sources into the hybrid model.

Finally, the full hybrid PE-FFP propagation model is computationally expensive and requires a variety of inputs to define the propagation environment. The

model's need for specific meteorological data limits its use to those with access to or means of calculating such data. The ability of the American Meteorological Society (AMS) and U.S. Environmental Protection Agency (EPA) Regulatory Model (AERMOD) to provide the necessary meteorological inputs to the propagation model is explored in Appendix C. It is suggested that an interface between the outputs of AERMOD and inputs of the hybrid model could be developed to utilize existing capabilities, originally developed to calculate meteorology parameters for emissions models.

To ease computational demand and further increase the model's accessibility, an approximate, fast access version of the code could be constructed. If numerous full model runs in different scenarios were compiled to create a representative collection of transfer functions, as described in Appendix B, the data could be assembled into manageable database forms. Methods of transfer function manipulation might then be used to represent a variety of meteorology and ground conditions without need to run the full hybrid model. Alternately, to reduce computation time while still using the full version of the model, low frequencies could be run with the full hybrid model and higher frequencies could be run with a faster method, such as ray tracing.

Existing Standard Aviation Noise Models

A.1 Introduction

In the following description and gap analysis for aviation noise assessment models, the Advanced Acoustic Model (AAM), the Integrated Noise Model (INM), and NOISEMAP have been investigated and compared.

The Advanced Acoustic Model (AAM) was developed by Wyle Laboratories under the Strategic Environmental Research and Development Program project SI-1304. It was designed to, eventually, replace NOISEMAP as the required method of modeling noise exposure near military air bases caused by flights and engine run-ups [13].

The Integrated Noise Model (INM) was developed by Volpe National Transportation Systems Center. It is the standard noise-modeling program employed by the Federal Aviation Administration (FAA) since 1978 and is designed to predict noise levels near airports [4].

NOISEMAP was developed by Wasmer Consulting and the Air Force Research Laboratory. It is the required method of modeling noise exposure near military air bases caused by flights and engine run-ups [72, 73].

A description and analysis of the algorithms in the four models is included below. It is separated into two categories: source representation and propagation algorithms.

A.2 Source Representations

A.2.1 INM

INM is a segmentation model that uses tabulated data known as Noise-Power (or operational mode, for helicopters)-Distance (NPD) curves in combination with spectral classes, which represent frequency content, to characterize the noise generated by infinitely long, straight flight tracks [74, 4]. Each curve corresponds to a particular aircraft engine power setting, determined by a measure called the corrected net thrust for fixed-wing aircraft or operational mode for helicopters, and provides noise levels at specific distances from the source. The helicopter noise-distance data are presented in sets of three curves for the dynamic operational modes, in order to account for the asymmetrical directivity of helicopter noise. A single curve, in combination with a helicopter-specific directivity adjustment, is presented for static operational modes. Actual flight paths are then modeled as combinations of finite, straight segments for fixed wing aircraft or an ordered set of procedure steps for helicopters, whose contributions are calculated based on corrections to the NPD data. INM does not correct helicopter procedure steps for nonstandard atmospheric conditions. As will be discussed later, the program adds other corrections for different effects that influence propagation such as atmospheric absorption, lateral attenuation, and terrain line-of-sight blockage [4].

A.2.2 NOISEMAP

NOISEMAP also uses data in the form of NPD curves, based on flyover spectral sound power levels for many aircraft, power settings and airspeeds, to represent a source. In fact, some NPD data for INM aircraft is used, specifically. However, NOISEMAP can also utilize hemisphere files for new aircraft in RNM, if available.

A.2.3 AAM

AAM uses sound spheres to characterize noise sources. The sound spheres may be broadband or pure tone noise data for a single flight condition. For fixed-wing aircraft, they represent a specific thrust, speed, and engine thrust vector angle. For conventional helicopters, they represent a specific flight path angle and airspeed.

For quasi-steady flight conditions for a rotorcraft, they represent a specific flight path angle, pylon angle, and airspeed [13].

A.3 Propagation Algorithm Adjustments

A.3.1 INM

Algorithms in this section are obtained from the INM Version 7.0 Technical Manual [4]. The following adjustments are applied to all INM aircraft in order to account for propagation effects:

- Atmospheric absorption (AA_{ADJ}), to correct for non-reference temperature and relative humidity.
- Acoustic impedance (AI_{ADJ}), to correct for non-reference temperature, atmospheric pressure and, indirectly, altitude of the airport.
- Noise fraction (NF_{ADJ}), to correct for a finite-length flight segment instead of an infinitely long flight path.
- Duration (DUR_{ADJ}), to correct for the flight segments finite duration, for non-reference aircraft speeds.
- Lateral attenuation (LA_{ADJ}), to correct for engine-installation, ground reflection, refraction, and scattering effects.
- Line-of-sight blockage (LOS_{ADJ}), to correct for any line-of-sight blockage between source and receiver due to terrain. Note: Invoking the LOS_{ADJ} is optional. If it is invoked, LOS_{ADJ} and LA_{ADJ} are compared, and the larger adjustment is applied while the other is ignored.

Two additional adjustment terms, used only for fixed-wing aircraft, include:

- Ground-based directivity adjustment (DIR_{ADJ}), used only for calculating noise behind the start-of-takeoff ground roll and for run-up operations.
- Thrust Reverser Adjustment (TR_{ADJ}), applied only when thrust reversers are deployed as part of the landing ground roll.

For a fixed-wing aircraft, the sound exposure ratio (E_{seg}) for a single flight-path segment operation, when the LOS_{ADJ} is invoked, is calculated as

$$E_{seg} = 10^{\frac{[L_{E,P,d-ADJ} + NF_{ADJ} + DUR_{ADJ} - \text{Max}[L_{A_{ADJ}}, LOS_{ADJ}] + TR_{ADJ} + DIR_{ADJ}]}{10}} \quad (\text{A.1})$$

where $L_{E,P,d-ADJ}$ corresponds to L_{AE} , L_{CE} , or L_{EPN} , as the desired sound exposure metric, in dB, resulting from interpolation of the NPD data and including atmospheric absorption and acoustic impedance.

Adjustment terms that are only applicable to helicopter flight operations include:

- Source noise due to advancing Mach Number (MN_{ADJ}), to correct for the non-reference, helicopter-specific airspeed, temperature and/or rotor RPM.
- Lateral Directivity (LD_{ADJ}), to correct for an elevation angle, β , different than those represented by the three sets of NPDs (45° to the right, 45° to the left, and 90° at the center), as shown in Figure A.1.

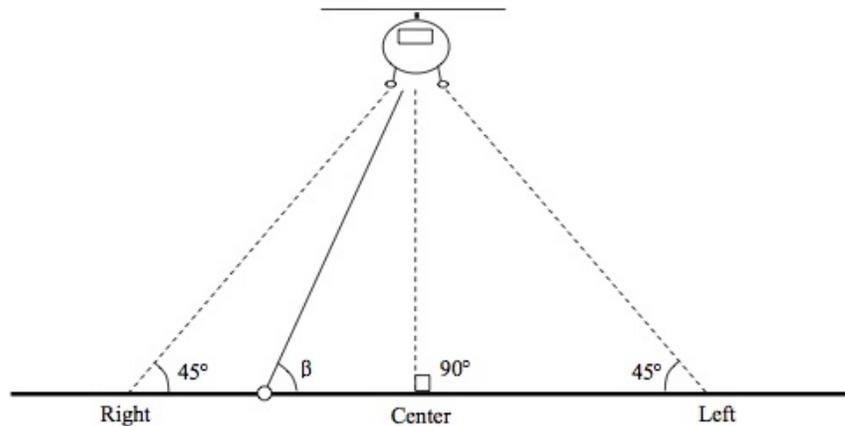


Figure A.1. Diagram of the elevation angle for helicopter lateral directivity adjustment. (Figure appeared as Figure 3-15 in the INM 7.0 Technical Manual [4]. Reproduced with permission.)

The sound exposure ratio ($E_{seg-HELI}$) for a single flight-path segment of a flight operation, when the LOS_{ADJ} is invoked, is calculated as

$$E_{seg-HELI} = 10^{\frac{[L_{E,P,d-ADJ} + NF_{ADJ} + DUR_{ADJ} - \text{Max}[L_{A_{ADJ}}, LOS_{ADJ}] + MN_{ADJ} + LD_{ADJ}]}{10}} \quad (\text{A.2})$$

where $L_{E,P,d-ADJ}$ corresponds to L_{AE} , L_{CE} , or L_{EPN} , as the desired sound exposure metric, in dB, resulting from interpolation of the NPD data and including atmospheric absorption and acoustic impedance.

Adjustment terms that are only applicable to helicopter static operations include:

- Static directivity (DIR_{HELI_ADJ}), to correct for the directivity of static operations at different angles around the helicopter.
- Static operation duration (t_{HELI_ADJ}), to correct for the duration of time spent on a segment during a static operation.

The sound exposure ratio ($E_{seg_HELI_static}$) for helicopter static operations, when the LOS_{ADJ} is invoked, is calculated as

$$E_{seg_HELI_static} = t_{HELI_static} 10^{\frac{[L_{E,P,d-ADJ} + N_{FADJ} - Max[L_{ADJ}, LOS_{ADJ}] + DIR_{HELI_ADJ}]}{10}} \quad (\text{A.3})$$

Out of all the listed adjustment terms, noise fraction, lateral attenuation, and line-of-sight blockage have been chosen to be discussed in further detail. These terms were selected because they have more elaborate implementations and may have larger implications for the accuracy of the physical situation being represented.

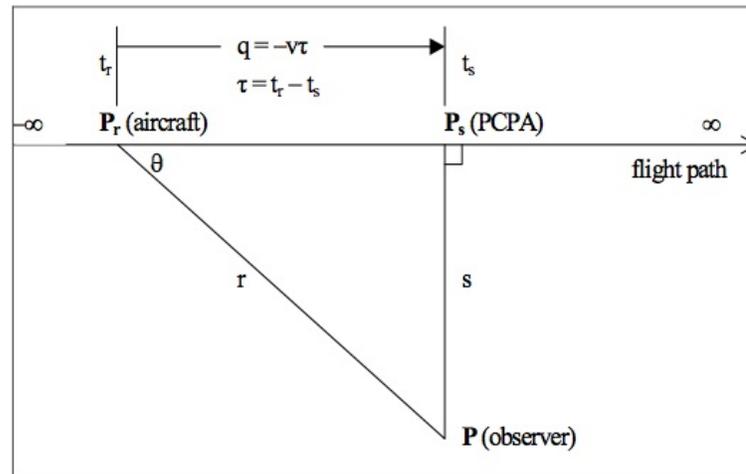
A.3.1.1 Noise Fraction

The noise fraction is calculated by assuming a “fourth-power, 90-degree dipole model of sound radiation.” This corresponds to the equation

$$p_r^2 = p_s^2 \left(\frac{s^4}{r^4} \right) \quad (\text{A.4})$$

where p_r , p_s , s , and r are defined in Figure A.2. The description “dipole” refers to the assumed directivity of the source and the description “fourth-power” comes from the inverse proportionality of p_r^2 to r^4 .

The noise fraction is calculated as the ratio between the mean-square receiver pressure integrated over the time range taken to cover the flight segment and the mean-square receiver pressure integrated for all time.



- r distance from the observer at point P to the aircraft at point P_r (feet),
- s perpendicular distance from the observer at point P to PCPA at point P_s (feet),
- q distance along the flight path relative to PCPA (feet),
- v speed of the aircraft (feet/s),
- t_r time at which the aircraft is located at point P_r (seconds),
- t_s time at which the aircraft is located at point P_s (seconds),
- τ time difference, t_r minus t_s (seconds),
- θ angle formed by the flight path and a connecting segment from the aircraft at point P_r to the observer at point P ,
- p_r root-mean-square sound pressure generated by the aircraft at point P_r , and
- p_s root-mean-square sound pressure generated by the aircraft at point P_s .

Figure A.2. Diagram of the observer/flight-path geometry. (Figure appeared as Figure C-1 in the INM 7.0 Technical Manual [4]. Reproduced with permission)

A.3.1.2 Lateral Attenuation

The lateral attenuation adjustment incorporates the engine-installation and other airplane shielding effects, ground reflection effects, and refraction and scattering effects. INM uses the algorithms specified in the SAE-AIR-5662 standard, “Method for Predicting Lateral Attenuation of Airplane Noise” to calculate the lateral attenuation for two different types of engine mountings for fixed wing aircraft. A geometric illustration of the variables used in the following equations can be seen in Figure A.3.

The directivity correction that incorporates the engine-installation effects, $E_{ENGINE}(\phi)$, is a function of ϕ , the depression angle, and is governed by different equations for jet aircraft with fuselage-mounted engines and jet aircraft with

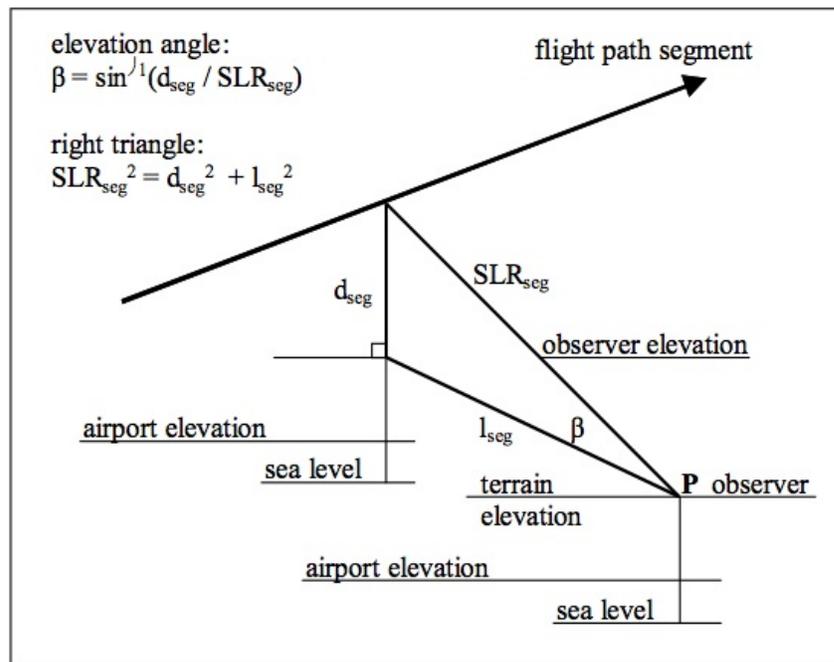


Figure A.3. Diagram of the lateral attenuation geometry. (Figure appeared as Figure 3-5 in the INM 7.0 Technical Manual [4]. Reproduced with permission.)

wing-mounted engines. A polar plot of these functions can be seen in Figure A.4.

Because directivity is represented by the set of 3 NPD curves and additional directivity adjustments for helicopters, the helicopter directivity correction for engine-installation effects is set to zero.

The lateral attenuation correction that incorporates ground effects $G(l_{seg})$ is a function of l_{seg} , the sideline distance (meters) in the horizontal plane from the observer to the projection of the closest point of approach of the flight segment. The ground effect attenuation, increases with increasing sideline distance and then levels off.

The lateral attenuation correction that incorporates the refraction and scattering effects $\Lambda(\beta)$ is a function of β , elevation angle. The refraction and scattering effect attenuation decreases with increasing elevation angle and reaches 0 dB at about 50° .

The total lateral attenuation adjustment, LA_{ADJ} , combines the contributions of engine-interaction, ground, and refraction-scattering effects for each receiver

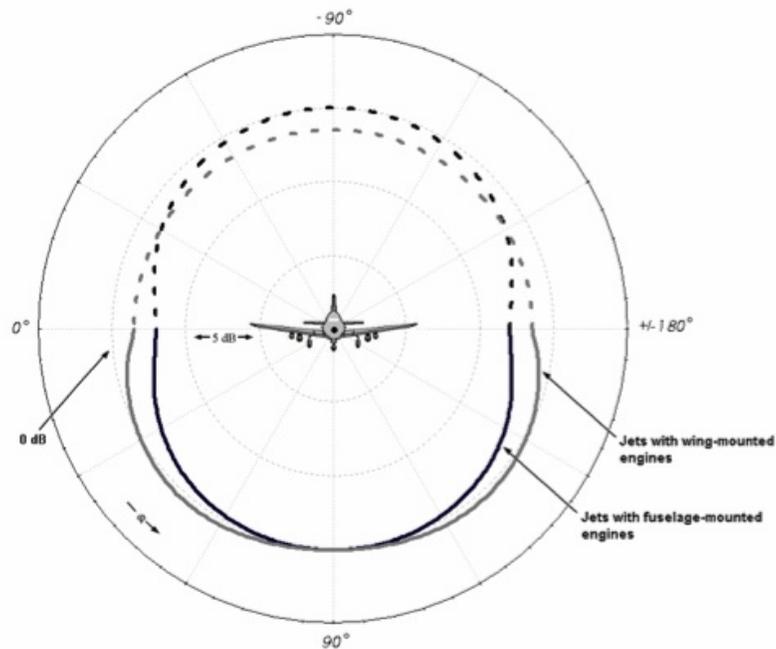


Figure A.4. Directivity of the engine-installation effects for wing-mounted and fuselage-mounted engines of jet-powered airplanes. (Figure appeared as Figure 3-7 in the INM 7.0 Technical Manual [4]. Reproduced with permission.)

location and is calculated as

$$LA_{ADJ(INM)} = - \left[E_{ENGINE}(\phi) - \frac{G(l_{seg})\Lambda(\beta)}{10.86} \right] \quad (\text{A.5})$$

Different algorithms are used for the lateral attenuation adjustment for “NOISEMAP aircraft,” which are included in the INM database. As in the algorithms used in the NOISEMAP model, the lateral attenuation adjustment neglects engine installation effects.

A.3.1.3 Line-of-Sight Blockage

The line-of-sight blockage adjustment is used to incorporate the effects of terrain that blocks the direct path from the source to the receiver. It is based on the difference in length between the direct path and the path over the top of the

terrain. This path length difference is calculated as

$$\delta_0 = (A + B) - C \quad (\text{A.6})$$

where A is the length of the path from the source to the diffraction point, B is the length of the path from the diffraction point to the receiver, and C is the length of the direct path from source to receiver, as can be seen in Figure A.5.

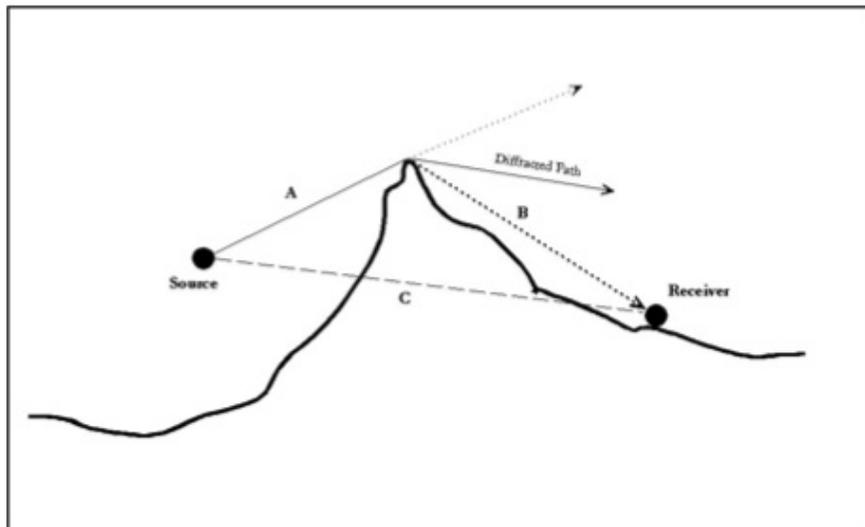


Figure A.5. Diagram of the line-of-sight (LOS) blockage geometry. (Figure appeared as Figure 3-12 in the INM 7.0 Technical Manual [4]. Reproduced with permission.)

The path length difference is used to predict the Fresnel Number N_0 as

$$N_0 = \pm 2 \left(\frac{\delta_0}{\lambda} \right) = \pm 2 \left(\frac{f \delta_0}{c} \right) \quad (\text{A.7})$$

where the positive sign is used when the line of sight between source and receiver is lower than the diffraction point and the negative sign is used when the line of sight is higher than the diffraction point, λ is the wavelength of the sound, f is the frequency of the sound, and c is the speed of sound.

The Fresnel Number is used to calculate the Barrier Effect as

$$\text{Barrier Effect} = \begin{cases} 5 + 20 \log_{10} \left(\frac{(2\pi|N_0|)^{1/2}}{\tan \left((2\pi|N_0|)^{1/2} \right)} \right) & N_0 < 0 \\ 5 + 20 \log_{10} \left(\frac{(2\pi|N_0|)^{1/2}}{\tanh \left((2\pi|N_0|)^{1/2} \right)} \right) & N_0 > 0 \end{cases} \quad (\text{A.8})$$

The Barrier Effect for each of the one-third octave-band center frequencies is used to calculate line-of-sight blockage adjustment as

$$LOS_{ADJ} = 10 \log_{10} \left(\sum_{i=17}^{40} 10^{\frac{\text{Barrier Effect}_i}{10}} \right) \quad (\text{A.9})$$

where *Barrier Effect_i* is the Barrier Effect for the *i*th one-third octave band. As previously stated, if the line-of-sight blockage is invoked, the line-of-sight blockage adjustment is compared to the lateral attenuation adjustment and only the larger is used. A limit of 18 dB is imposed on the line-of-sight blockage adjustment.

A.3.2 NOISEMAP

Algorithms in this section are obtained from the NMAP 7.0 User's Manual [73]. NOISEMAP incorporates atmospheric absorption, lateral attenuation, two modeled categories of aircraft events—flight (departure, arrival, closed pattern and interfacility), and static (preflight and maintenance), an adjustment to correct for non-reference aircraft speeds, an adjustment to correct for non-reference aircraft altitudes, a takeoff roll model, and corrections for topography and ground impedance.

Atmospheric absorption is calculated using the ANSI S1.26-1995 (ASA 113-1995), "Method for Calculation of the Absorption of Sound by the Atmosphere" standard [10]. Monthly average temperature, relative humidity and atmospheric pressure are incorporated.

NOISEMAP uses the lateral attenuation algorithm of "Prediction Method for Lateral Attenuation of Airplane Noise During Takeoff and Landing" SAE AIR 1751 for civilian jet and propeller-driven aircraft and that of "Lateral Attenuation of Military Aircraft Flight Noise" Technical Report AAMRL-TR-89-034, for military jet and propeller-driven aircraft and helicopters. In this standard, the lateral

attenuation algorithm is based on ground effects and neglects engine installation effects.

The airspeed correction is used to correct the source data from the 160 knot reference speed to the actual speed of the aircraft. It is calculated as:

$$\text{Airspeed Adjustment} = -10 \log_{10}(V/160) \quad (\text{A.10})$$

An adjustment is also applied to correct the source data from the sea-level reference condition to a better estimate of the aircrafts height, decreasing the reference noise level by 2 dB per 10,000 feet. The correction is applied under the assumption that the effective thrust decreases with altitude and that as effective thrust decreases, the noise produced decreases. The correction is calculated as:

$$\text{Altitude thrust adjustment} = \log_{10}^{-1}[0.00002(1000 - \text{alti})] \quad (\text{A.11})$$

where *alti* is aircraft altitude in feet.

Takeoff roll modeling is divided into two parts, initial directivity for the run-up and an acceleration model. Data for the run-up directivity is stored in look-up tables and the acceleration model calculates two different adjustment terms, one for the start-of roll position and another for the aircraft rotation position.

Corrections for ground impedance incorporate effects of hard, soft, or a mixture of hard and soft surfaces for flat ground. Topography, including valleys and hills, can also be accounted for. However, when the topography mode is invoked, the lateral attenuation adjustment is not used. The algorithms are based on the developments of the NATO/CCMS Working Group on Topography, including the Plovsing paper, "Aircraft Sound propagation over Non-flat Terrain. Development of Prediction Algorithms." They are rooted in the Maekawa and Rasmussen algorithms for shielding and diffraction effects. However, as in INM, anisotropic effects such as terrain (and weather) are not handled properly.

A.3.3 AAM

Algorithms in this section are obtained from the Advanced Acoustic Model Technical Reference and User Manual [13]. AAM is a ray-trace model. It applies the gen-

eral adjustments of geometric spreading (A_{spread}), atmospheric absorption (A_{atm}), and ground reflection and attenuation losses (A_{grd}). Nonlinear propagation effects (A_{nlin}) are incorporated for high thrust military aircraft. Optional effects include topographic attenuation (A_{topo}) and wind attenuation or amplification (A_{wind}). To add the contribution of all these adjustment terms, the sound levels at a distance r for a single flight-path segment operation, when the topographic attenuation and wind attenuation or amplification are invoked, is calculated as

$$L(r) = L(r_0) + A_{spread} + A_{atm} + A_{grd} + A_{topo} + A_{wind} + A_{nlin} \quad (\text{A.12})$$

where $L(r_0)$ is the free-field, lossless sound level at a distance r_0 from the source.

Atmospheric absorption is calculated using the ANSI S1.26 (R2004) method. It is computed at height intervals of 1,000 feet.

Ground impedance effects are based on the work by Chien and Soroka and then by Chessel with corrections of Daigle, using Doppler-shifted frequencies. The properties of the ground are characterized by the flow resistivity parameter.

Nonlinear effects are based on the theory developed by Gee, et. al., utilizing the Burgers Equation. The adjustment term is obtained by calculating the difference between the nonlinear calculation and a corresponding linear calculation.

Varying terrain effects are based on a theoretical model, developed by Rasmussen, which extends the geometrical theory of diffraction to finite impedance ground. AAM categorizes terrain features as flat terrain, uphill, downhill, valley, wedge or screen with one flat, or wedge or screen with two flats. It accounts for shielding (by wedges), structures (screens), multiple reflections in valleys, and ground impedance.

Wind and temperature gradient effects are based on the theory developed by Plotkin and Huber. The model can either use effects calculated with a straight ray propagation model that calculates atmospheric absorption based on the user defined atmospheric profile at different altitudes, or it can use effects calculated with the weather module RNMwea pre-processor that utilizes a curved ray tracing propagation model. Curved ray effects have only been incorporated when analyzing a single event analysis occurring over uniform, flat terrain. A horizontally stratified media is assumed. A turbulent decoherence effect, as described by Chessell, can be

included by specifying a turbulence parameter. Diffraction of sound into a shadow zone is calculated using the method developed by Berry and Daigle.

Contributions of broadband, narrowband, and pure tone sound to noise levels at observer points are tracked independently, combined accounting for phase and coherence, and tracked over time.

A.4 Discussion of Algorithms

A.4.1 Source Representation

Among the more standard noise calculation methods such as INM, NOISEMAP, and AAM, there are two types of noise source characterizations: NPD-type data and sound spheres. The NPD-type data, as used in INM and NOISEMAP, represents an infinite flight path. Corrections are applied to these curves to make them represent finite segments. Sound spheres, as used in AAM, represent sound emanating from a single point with a particular directivity built in for different frequency representations.

While a sound sphere is a more detailed characterization of the source (AAM is even capable of producing a time history of sound levels) and has the potential to yield more accurate noise level calculations, the sound spheres are rather expensive and time-consuming to obtain. Consequently, as an extensive NPD database already exists for a large range of aircraft and engine types, the NPD representation prevails for the time being.

Although the effects incorporated by the different models are often different in form or implementation, certain categories of effects emerge and will be discussed below. A distinction is made between physical propagation effects and adjustments required to compensate for the NPD source representation. Therefore, the first section in the following analysis will discuss the adjustments used in INM and NOISEMAP. The second will discuss the physical propagation effects.

While the considered models all contain adjustments for atmospheric absorption, often the same or similar standards are used. It will, therefore, not be considered here in depth.

A.4.2 Segmentation Model Adjustments

A.4.2.1 NPD-specific adjustments

Because of the nature of the NPD data and segmentation method, adjustments must be added in order to convert an infinite flight path representation at reference conditions to a finite flight segment representation at actual conditions. Such terms in INM are the noise fraction adjustment and the duration adjustment. These additional approximations are applied only to correct the simplified representations of the source. In order to apply these approximations, certain assumptions must be made regarding the nature of the sound produced. INM justifies the directivity pattern it assumes as an incorporation of the phenomena that “include atmospheric absorption, which is accentuated in front of the aircraft due to Doppler shift, sound refraction away from the hot gases behind the airplane, and ground attenuation” [4]. However, the directivity pattern does not seem to incorporate the cardioid-like shape that might be expected from jet directivity and convection effects, or the dipole directivity of unsteady drag, but does resemble the trailing edge noise dipole directivity. Therefore, a generalization is made, as the calculated directivity is applied indiscriminately to all flight segments regardless of flight operation being performed (which might affect the configuration of the aircraft and, thus, which sound sources are present or dominant). These types of adjustment terms and the assumptions that accompany them are not included in AAM, which characterizes sources differently.

A.4.2.2 Lateral attenuation

The lateral attenuation adjustment used in INM, for INM aircraft, incorporates the engine-installation and other airplane shielding effects, ground reflection effects, and refraction and scattering effects. Similar to INM’s noise fraction adjustment, the lateral attenuation adjustment implies a certain directivity profile, dependent on the characteristics of a representative aircraft. It also assumes general ground and atmospheric conditions. The lateral attenuation adjustment used in NOISEMAP does not incorporate the airplane shielding effects. Again, this is a generalization that applies certain assumptions and approximations that may not be accurate for all types of aircraft and all ground and atmospheric conditions.

Alternatively, AAM does not use any explicit lateral attenuation adjustment. It relies on the source representation to provide the directivity profile and adds separate corrections, which allow for more variety in representation, for ground and atmospheric conditions.

A.4.3 Physical Propagation Effects

A.4.3.1 Uneven terrain

One of the only range-dependent effects taken into account in the noise models discussed is uneven terrain. Uneven terrain can be incorporated, to an extent, in INM by invoking the optional line-of-sight blockage adjustment. The approach used has been validated for distances up to 1000 feet. However, it does not make a distinction between different terrain shapes. NOISEMAP goes one step further by distinguishing between flat ground, valleys, and hills. However, when the topography mode is invoked, calculation times can be expected to increase by an order of magnitude. AAM goes further still by categorizing terrain features as flat terrain, uphill, downhill, valley, wedge or screen with one flat, or wedge or screen with two flats.

A.4.3.2 Ground Effects

In INM, ground effects are built into the lateral attenuation adjustment. However, the lateral attenuation algorithms were developed from field measurements over acoustically soft ground and the user is warned that sound levels may be under-predicted if the ground being represented is a hard surface. INM does accept directivity data for static helicopter operations for both hard and soft ground.

NOISEMAP has two ways of applying ground effects. The first is similar to that of INM, where it is included in the lateral attenuation adjustment. However, it can also be invoked in a separate mode in which an impedance file, which specifies a grid of ground impedance data, is used. The ground effects for AAM are based on the method developed by Chien and Soroka, extended by Chessel with corrections by Daigle. These use a more descriptive ground impedance parameter for a more accurate ground effect representation.

A.4.3.3 Meteorological Effects

INM does correct for the effect of headwind on the power required of an aircraft during the beginning of the flight model (for example, it is incorporated in such parameters as takeoff ground-roll distance, climb angle, and acceleration segment distance). However, INM only minimally accounts for meteorological effects in propagation with the atmospheric attenuation adjustment (calculated using the temperature and relative humidity entered by the user), the acoustic impedance adjustment (a function of temperature, atmospheric pressure and, indirectly, altitude), and the lateral attenuation adjustment (meant to incorporate, in part, effects of refraction and scattering, but including no atmospheric input parameters).

NOISEMAP's treatment of meteorological effects is no more sophisticated than that of INM.

AAM includes a curved ray model for incorporation of meteorological effects, using a horizontally stratified atmosphere. However, its use is limited to single event analysis occurring over uniform, flat terrain.

A.5 Conclusion

In the comparison of INM, NOISEMAP, and AAM, it was found that many of the necessary aviation noise production and propagation effects have been addressed. However, terrain and meteorology were the effects most often neglected.

The comparison also revealed different degrees of accuracy in the models' source representations. These included a simplified representation, with a compact, extensive database and smaller computational requirements and a detailed source representation with a limited database.

Guidance on Full Model-Fast Access Model Conversion

B.1 Introduction

The full version of the Hybrid Parabolic Equation (PE)-Fast Field Program (FFP) Model is primarily a research tool to develop accurate predictions including a wide variety of real-world propagation and environmental effects. It is not intended to be a fast model. In order to manage runtime, additional work is needed to translate typical outputs of the Hybrid Model into a fast access approach. This can be accomplished by choosing appropriate classes of conditions, running the code under those conditions, amassing a database of results, and formulating transfer functions for each combination of conditions. The transfer functions can be used as the basis of a fast access propagation model.

B.2 Meteorology

B.2.1 Refractive Atmosphere Classes

In the full Hybrid Model, a refractive atmosphere is described with an effective sound speed profile, defined by an unrestricted parameter. The Parabolic Equation and Fast Field Program methods do not limit the definition of the sound speed profile to specific functions and, without too much adjustment of the Hybrid Model

code, an arbitrary sound speed profile could be accommodated. In order to construct a fast access model with a more reasonable runtime, the range of possibilities for characterizing the atmosphere must be restricted. This can be accomplished by defining different classes of the atmosphere, based on realistic conditions. Each class is characterized by a specific sound speed profile.

It has been found that using 9 different classes of a combined logarithmic-linear sound speed profile to characterize many random (though realistic) atmospheric conditions “reduced the risk to less than 20% that the average sound level at 1000 m range in an arbitrary climate is estimated with an error of more than 2 dB” [75]. Additionally, 25 classes could “further reduce the risk of exceeding 2 dB deviation from the reference to less than 10% of the cases” [75].

The European Harmonoise propagation model specifies a logarithmic-linear sound speed profile function:

$$c_m(z) = c_0 + a_m \ln \left(1 + \frac{z}{z_0} \right) + b_m z \quad (\text{B.1})$$

where c_m is the effective sound speed of profile m ($m = 1, 2, \dots, M$), z is height, c_0 is a constant sound speed, z_0 is the roughness length of the ground surface, and a_m and b_m are parameters of the sound speed profile m . It is stated that $M = 25$ is a typical number of profiles and typical parameter values are $z_0 = 0.1$ m, $a_m = -1.0, -0.4, 0, 0.4, 1.0$ m/s and $b_m = -0.12, -0.04, 0, 0.04, 0.12$ 1/s [76].

A similar manner of classifying states of the atmosphere is described in detail in E. M. Salomons, et al. “Long-term average sound transfer through the atmosphere: predictions based on meteorological statistics and numerical computations of sound propagation” [5]. Schemes of the meteorological model and the method for computing long-term average transfer functions are shown in Figures B.1 and B.2, respectively, as they appear in the paper [5].

In Salomons, et al. [5], 27 different sound speed profiles are used to classify the atmosphere. A meteorology model uses inputs of the part of the day (day-time or nighttime), season, ground roughness, and direction of sound propagation to construct 1620 sound speed profiles by way of Pasquill stability classes and Businger-Dyer profiles. The representative 27 profiles are chosen from these 1620 profiles. Because calculations of average sound levels are dominated by downward

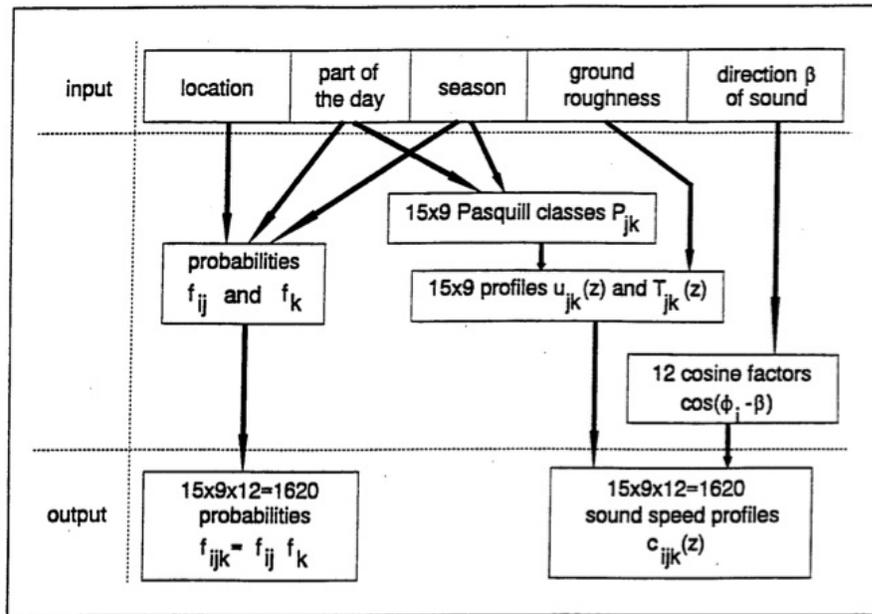


Figure B.1. Scheme of meteorological model used in [5]. Reproduced with permission.

refracting atmospheric conditions, which produce higher sound levels, the classes are not split evenly between upward and downward refracting profiles, and more downward refracting profiles are used [5].

Once the 27 profile classes are determined, each is input into a PE run for a specific frequency, source height, and ground impedance. Therefore, a database of 27 different transfer functions is constructed for each frequency-source height-ground impedance combination [5]. Numerous combinations should be run for each of the 27 meteorology classes until an adequate database is amassed. Salomons, et al. [5] averages over 10 frequencies per octave band in order to obtain octave band average transfer functions.

If one average transfer function is sought to predict long-term average sound levels for a specific location, the statistical properties of the locations meteorological conditions must be known. In this case, statistical weights can be applied to give appropriate contributions of each meteorology class to the total level [5, 77].

Because of the consideration for efficient categorization of classes, and a detailed and complete method description in Salomons, et al. [5], it is recommended that this technique for calculating transfer functions be used to formulate the fast access version of the Hybrid Model. Transfer functions can be stored in lookup tables,

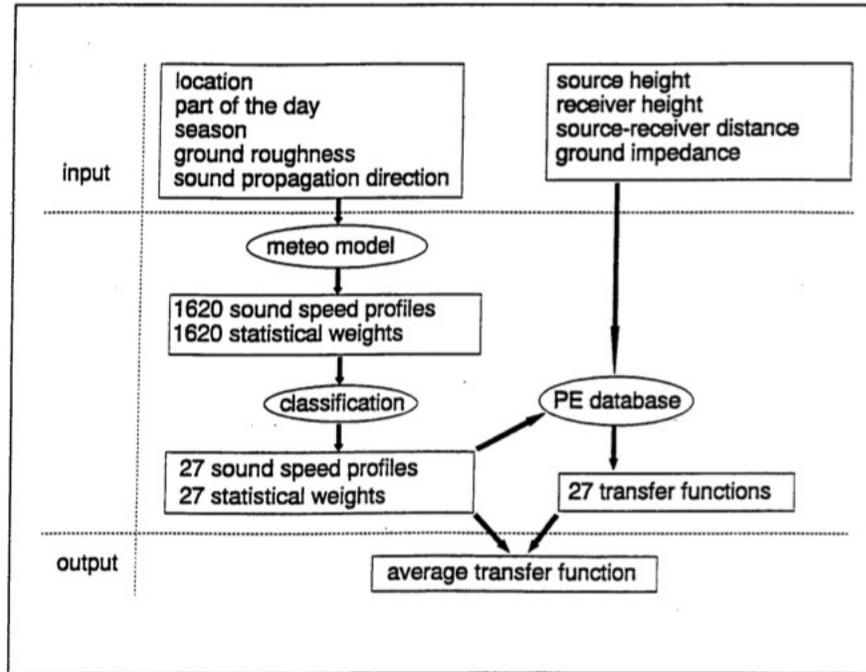


Figure B.2. Scheme of long-term average transfer function computation used in [5]. Reproduced with permission.

or polynomials (20th order polynomial had been found adequate in one instance [78]) for each octave band, ground type, meteorological class, source height, and receiver height.

B.2.2 Turbulence

The main effects of turbulence on sound propagation are the scattering of sound into shadow zones by turbulent eddies, and loss of coherence between sound traveling different paths, due to phase fluctuations [76]. Turbulence is not included in the Salomons, et al. [5] reference because, although the effect of turbulence in shadow zones can be large, the low levels in the shadow region have a negligible contribution to the average transfer function [5]. Since the influence of turbulence on average level values is small, and the computational requirements of modeling it accurately with numerous realizations are large, the effect of turbulence can be simplified and incorporated into propagation models as an upper allowable limit of excess attenuation (not including atmospheric absorption or spherical spreading). The Harmonoise reference model uses a limit of 15 dB [76]. A similar restriction

can be imposed in the fast access version of the Hybrid Model.

B.3 Ground Impedance Discontinuities

The full Hybrid Model can include the effects of propagation over varying impedance ground (for example grass to water) when it is provided the different impedances of ground and the ranges over which they are present. A separate full model run is then performed for each different heterogeneous ground condition.

To quickly include the effects of propagation over ground with varying impedance in a fast access model, an interpolation method has been developed by E. M. Salomons and F. H. A. van den Berg [6]. As described in their paper, the effects of a heterogeneous ground can be computed from correctly combining results using separate calculations over each homogeneous ground surface. With this method, results for arbitrary distances of different ground surfaces could be quickly calculated from a smaller subset of code runs for each representative type of ground surface.

Salomons and van den Berg found that, when sound propagated over a grass-water (soft-hard ground) interface in a downward refracting atmosphere, a transfer function calculated with a PE model follows the grass curve up to the transition, and next continues with the slope of the water curve [6]. Using the simple formulae:

$$\begin{aligned} L(r) &= L_g(r) && \text{for } r < r_x \\ L(r) &= L_w(r) + L_g(r_x) - L_w(r_x) && \text{for } r > r_x \end{aligned} \quad (\text{B.2})$$

where L is the heterogeneous ground transfer function, L_g and L_w are the transfer functions for grass and water, respectively, r is range, and r_x is the range of the transition between ground surfaces, good agreement was achieved between the heterogeneous ground PE results and the interpolation of homogeneous ground PE results. A comparison of the results for a grass-water transition can be seen in Figure B.3. This method gave poor agreement when the g and w indices were interchanged. However, when the principle of reciprocity was utilized to compute reciprocal situations, agreement between models was, again, good.

The method was also tested on ground surfaces with both grass-water-grass-water and water-grass-water-grass combinations. Good agreement was achieved

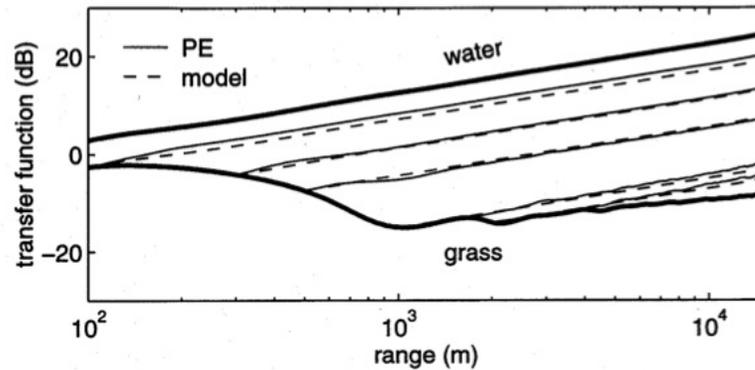


Figure B.3. Transfer function versus range for propagation in a downward refracting atmosphere over homogeneous water and grass (thick lines) and over various grounds with a grass-water transition (thin lines). Thin solid lines were calculated with a single PE run, using the exact grass-water configuration. Thin dashed lines were calculated using the interpolation method to combine homogeneous grass and homogeneous water PE runs [6]. Reproduced with permission.

when the ground was permuted such that all grass areas were moved toward the source and all water areas were moved to the receiver. In the paper, a low altitude source and only hard and soft ground conditions were explored [6]. Thus, application of this method to other conditions should be validated.

B.4 Terrain

All discussion in this report so far has assumed a flat ground surface. Because there are an unlimited number of possible configurations of terrain geometry, other than Fresnel number methods (i.e., “A+B-C” methods) we are not aware of a general simplified method of including terrain effects into a fast access form. Fast access techniques for canyon-like environments, where there are multiple reflections, would be especially challenging. If a fast access model including terrain effects is needed for a specific location with a fixed geometry, a location-specific database could be created with full Hybrid Model runs using only that terrain geometry.

However, as results illustrated in Chapter 4 for the Near Ground Operations Test Cases suggest, unless terrain obstructions cause a break in the direct line of sight, forming a shadow zone, terrain effects may be minimal. If the effects are small enough, they could be neglected.

B.5 Conclusion

In order to construct a fast access version of the Hybrid Model, transfer functions can be generated for different combinations of meteorological and (homogeneous) ground conditions. These transfer functions can be stored in lookup tables or polynomials for each octave band, ground type, meteorological class, source height, and receiver height. Turbulence can be approximated by setting an upper limit to the allowable excess attenuation.

To incorporate the effects of a heterogeneous ground, an interpolation method that correctly combines transfer functions from homogeneous ground runs can be used. Therefore, the fast access model can utilize existing transfer functions, eliminating the need for individual runs of specific heterogeneous ground conditions.

Because there are an unlimited number of possible terrain configurations, a method of incorporating terrain effects into a general fast access model is not known. However, test case results suggest that inclusion of terrain effects may be necessary only when shadow zones are formed.

Finally, if long-term transfer functions are desired, they can be constructed by statistically weighting the individual transfer functions by the probability distribution of the chosen set of meteorological classes.

Assessment of the Potential for Using Emissions Meteorological Models for Noise Propagation

C.1 Introduction

The Hybrid PE-FFP Propagation Model can incorporate meteorological effects such as a refractive atmosphere and homogeneous, isotropic atmospheric turbulence. However, it requires specific meteorological input parameters to characterize the atmosphere. The viability of exploiting emissions model, AERMOD, meteorological capabilities to compute Hybrid Model inputs is explored.

C.2 Model Requirements

Inputs required to adequately define the state of the atmosphere in the Hybrid Model in terms of both refractive sound speed profiles and refractive-index fluctuations of turbulence are discussed below.

C.2.1 Refractive Sound Speed Profiles

To define the refractive state of the atmosphere, the Hybrid Model code uses an effective sound speed profile defined as

$$c_{eff} = c + u \quad (\text{C.1})$$

where c is the adiabatic sound speed and u is the wind velocity in the direction of propagation. The effective sound speed profile is a way of approximating a moving atmosphere by a non-moving atmosphere [1]. Currently, the code has built-in linear and logarithmic profile options defined as

$$\begin{aligned} c_{eff} &= c_0 + az \\ c_{eff} &= c_0 + b \ln \left(\frac{z}{z_0} + 1 \right) \end{aligned} \quad (\text{C.2})$$

respectively, where c_0 is the sound speed at the ground, a the sound speed gradient, z the height above the ground, b the parameter of logarithmic sound speed, and z_0 the roughness length of the ground surface. The user must specify the sound speed at the ground, chose either the linear or logarithmic function to represent the profile, and provide either the linear gradient, if the linear function is chosen, or the parameter of logarithmic sound speed profile, if the logarithmic profile is chosen. For a logarithmic profile, the roughness length of the ground surface must also be specified.

Without major code adjustment, either an alternate profile function, a detailed vector definition of effective sound speed at specified height values, or a detailed matrix definition of sound speed at specified height and range values could be accommodated.

C.2.2 Atmospheric Turbulence

To describe the turbulent state of the atmosphere, the code assumes frozen turbulent fluctuations and approximates the turbulence as homogeneous and isotropic with a von Kármán spectrum [1]. This is a simplified representation, but such approximations have been successfully used in modeling outdoor sound propagation

experiments [42]. To characterize the turbulence, the model requires inputs of the:

- Outer length scale of turbulence L_0
- Normalized structure parameters of temperature and wind velocity fluctuations, $\frac{C_T^2}{T_0^2}$ and $\frac{C_v^2}{c_0^2}$, respectively

The model uses these inputs to calculate the von Kármán spectral density function, from which the refractive-index fluctuation (fluctuation of the effective sound speed) is calculated. The refractive-index fluctuation is used to introduce a phase factor at each range step, mimicking the effect of turbulence [1].

C.3 AERMOD Outputs Transformed to Noise Propagation Model Inputs

The American Meteorological Society (AMS) and U.S. Environmental Protection Agency (EPA) Regulatory Model (AERMOD) [79] can calculate a number of meteorological parameters using measured and estimated inputs. The calculated parameters can, with manipulation, be useful inputs to the Hybrid Model.

C.3.1 Refractive Atmosphere Parameters

To supply all necessary Hybrid Model refractive atmosphere inputs, AERMOD must provide the parameters to fully define the effective sound speed profile. This can be accomplished by providing both a wind speed profile (including direction) and an adiabatic sound speed profile (based on temperature). AERMOD is able to estimate vertical wind speed profiles based on measurements and similarity parameterizations. Divided into three vertical layers, the wind speed is calculated as:

$$\begin{aligned}
 u\{z\} &= u\{7z_0\} \left(\frac{z}{7z_0} \right) && \text{for } z < 7z_0 \\
 u\{z\} &= \frac{u_*}{k} \left[\ln \left(\frac{z}{z_0} \right) - \Psi_m \left\{ \frac{z}{L} \right\} + \Psi_m \left\{ \frac{z_0}{L} \right\} \right] && \text{for } 7z_0 \leq z \leq z_i \\
 u\{z\} &= u\{z_0\} && \text{for } z > z_i
 \end{aligned} \tag{C.3}$$

where u is wind speed, z is height, z_0 is the roughness length, u_* is friction velocity, k is the von Kármán constant, Ψ_m is defined by the 1984 Panofsky and Dutton paper for the convective boundary layer and by the 1985 van Ulen and Holtslag paper for the stable boundary layer, L is the Monin-Obukhov length, and z_i is the mixing height [79].

The wind direction is defined by AERMOD to be constant with height both above the highest and below the lowest measurement and to vary linearly between measurements [79]. While AERMOD is able to calculate the potential temperature gradient [79], ultimately, an absolute temperature profile is required to compute an effective sound speed profile. Following Salomons [1], the potential temperature can be related to the absolute temperature and pressure as:

$$\theta = T \left(\frac{p_0}{p} \right)^{(\gamma-1)/\gamma}, \quad (\text{C.4})$$

where θ is potential temperature, T is absolute temperature (in kelvin), $p_0 = 105$ Pa, p is pressure, and γ is the specific-heat ratio ($\gamma = 1.4$ for air). Assuming a vertical gradient of the average pressure:

$$\frac{dp_{av}}{dz} = -\rho_{av}g, \quad (\text{C.5})$$

where p_{av} is average pressure, ρ_{av} is the average density, and g is the acceleration of gravity, and applying the adiabatic equation of state and ideal gas law, the temperature gradient can be defined as:

$$\alpha_0 \equiv \frac{dT}{dz} = -\frac{\gamma-1}{\gamma} \frac{\rho T}{p} g. \quad (\text{C.6})$$

From Equations C.4 and C.6, and approximating $p \approx p_0 + \frac{dp}{dz}z$ the relationship between potential and absolute temperature is:

$$\theta \approx T - \alpha_0 z. \quad (\text{C.7})$$

This can be used to calculate the absolute temperature profile. The absolute temperature profile can be used to determine the adiabatic sound speed profile

with the relationship:

$$c = \sqrt{\gamma RT} \quad (\text{C.8})$$

where $R = 287 \text{ J kg}^{-1} \text{ K}^{-1}$ is the universal gas constant [1].

Therefore, using the above manipulations of AERMOD outputs, it seems likely that an effective sound speed profile can be calculated. However, this has yet to be tested.

C.3.2 Turbulence Parameters

Because atmospheric turbulence is approximated as homogeneous and isotropic, and a von Kármán spectrum of refractive-index fluctuations is assumed, the Hybrid Model only requires inputs of the outer scale of turbulence, and the normalized structure parameters of temperature and wind velocity fluctuations.

Turbulence is often described in terms of a spectrum including three subranges: the energy-containing subrange, the inertial subrange, and the dissipative subrange, shown in Figure C.1, as it appears in Salomons [1]. In the energy-containing subrange, the von Kármán spectral density levels off, due to the limiting boundary conditions of the flow. In the inertial subrange, the von Kármán spectral density follows the Kolmogorov spectral density, based on the cascade process in which energy from large eddies gets passed to smaller eddies, which pass energy to even smaller eddies, etc. In the dissipative subrange, energy from the eddies is converted into heat. However, these eddies are small compared to acoustic wavelengths and are, therefore, unimportant to sound propagation [1].

The “outer scale of turbulence” is the length scale that marks the transition between the energy-containing and inertial subranges, or the size of the largest, energy-containing eddies. This size is on the order of the boundary layer height [1]. The “outer scale” is also referred to in the literature as essentially equivalent to the “integral length” [80] and “mixing length” [81]. Therefore, the mixing height z_i calculated by AERMOD, should be a satisfactory input for L_0 .

The structure parameters of temperature and velocity fluctuation, C_T^2 and C_v^2 are measures of the temperature and velocity fluctuations in a turbulent atmosphere [82]. If a homogeneous, isotropic turbulence is assumed, as in the Hybrid Model, these parameters will be constants. AERMOD, however, does not make

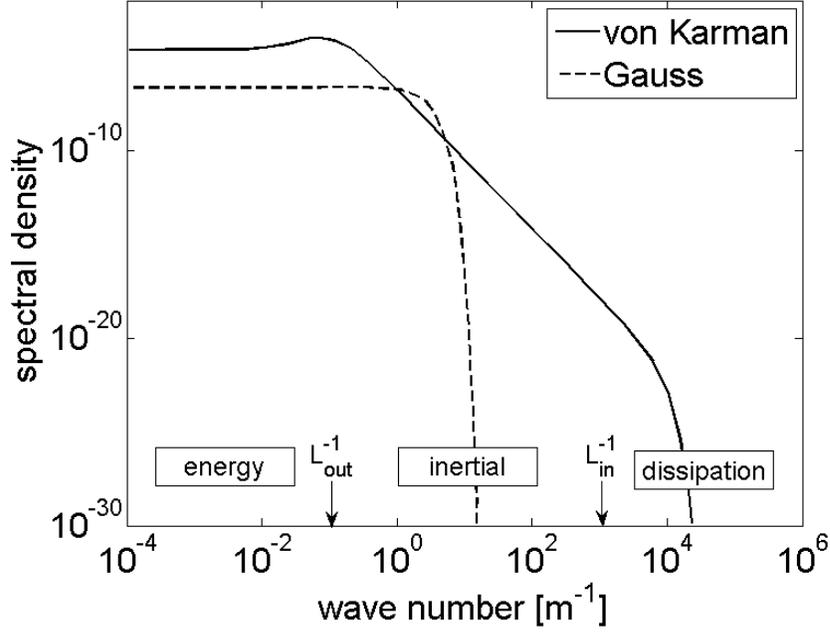


Figure C.1. Example of a von Kármán spectrum, after [1] Figure I.3. The von Kármán spectrum parameters used are $K_0^{-1} = 10 \text{ m}$, $\frac{C_T^2}{T_0^2} = 6 * 10^{-7} \text{ m}^{-2/3}$, $\frac{C_v^2}{c_0^2} = 2 * 10^{-6} \text{ m}^{-2/3}$. The energy-containing, inertial, and dissipative subranges are shown.

these assumptions, and calculates variances of the mechanical (caused by wind) and convective (caused by temperature) components of turbulence as functions of height. Therefore, a gap exists in achieving compatible definitions of the structure parameters. To reconcile this difference, either (1) a method of calculating effective temperature and velocity fluctuation variance constants can be sought, or (2) a different model of turbulence can be substituted for the one currently used by the Hybrid Model. If the first option is chosen, the variances of temperature and velocity fluctuations can then be related back to the structure parameters of temperature and velocity fluctuations using the outer scale of turbulence as

$$\begin{aligned}
 \sigma_t^2 &\cong 0.523 C_T^2 K_0^{-2/3} \\
 \sigma_v^2 &\cong 0.523 C_v^2 K_0^{-2/3} \\
 K_0 &= \frac{2\pi}{L_0}
 \end{aligned} \tag{C.9}$$

where σ_t^2 is the variance of temperature fluctuations, C_T^2 the structure parameter of temperature, σ_v^2 the variance of wind velocity fluctuations, C_v^2 the structure

parameter of wind velocity, and L_0 the outer scale of turbulence (largest eddy size).

Finally, the normalized structure functions can be calculated using a representative temperature and sound speed for T_0 and c_0 .

C.4 Conclusion

To incorporate meteorological effects into the Hybrid Model, input parameters that fully define the effective sound speed profile and the assumed homogeneous, isotropic turbulence must be provided. Most of the necessary parameters can be obtained by manipulation of meteorological parameters calculated by AERMOD. The AERMOD outputs of vertical wind speed profile and vertical potential temperature gradients can likely be utilized to define the effective sound speed profile. However, this should be confirmed through testing of the variable manipulation method. The AERMOD outputs of mixing height, and turbulence variance parameters can be utilized in the effort to define the turbulence. However, the gap between use of a homogeneous, isotropic turbulence assumption in the Hybrid Model and no such assumption in AERMOD must be closed if compatible definitions of the structure parameters of temperature and wind velocity fluctuations are to be achieved. If this difference is reconciled and the method of determining sound speed profile is validated, the turbulence and refractive atmosphere will be fully defined.

Appendix **D**

Hybrid PE-FFP Model Code Diagram

D.1 Description of the Diagram

This appendix presents a flow chart of the hybrid PE-FFP model code. The first section provides a list of the input options available to the user within the input text file, 'Hybrid Model Parameters.txt'. The second section gives an overview of the main functions of the code and brief descriptions of their purpose. The third section gives a detailed breakdown of all functions used in the code, where the supplemental functions are called, and the steps of calculations within the major functions.

Inputs:

Model Description

- propagation model used ('PE', 'FFP', or 'hybrid')
- final desired grid geometry ('vertical' or 'horizontal')
- axisymmetry of propagation conditions ('yes' or 'no')
- option for producing vertical 2D plot ('yes' or 'no')
- reference source level at 1 m distance
- directivity of the source ('yes' or 'no')

Frequency Specification

- run for discrete frequencies or 1/3-octave bands ('discrete' or 'bands')
- vector of discrete frequencies, if 'discrete' was entered
- start and stop frequencies, if 'bands' was entered
- option for saving broadband level data and producing broadband plot ('yes' or 'no')

Propagation Grid Definition

- grid range
- grid height
- elevation angle limit used for PE
- increment angle for azimuth
- receiver height
- range step to be saved for plotting purposes

Terrain

- characteristic of the terrain ('flat' or 'uneven')

Impedance

- number of different impedances of the ground (1 or 2)
- option to input impedance or the effective flow resistivity ('Z' or 'EFR')
- vector with different impedances
- vector with range values at which the discontinuities occur

Meteorology--Sound Speed Profile

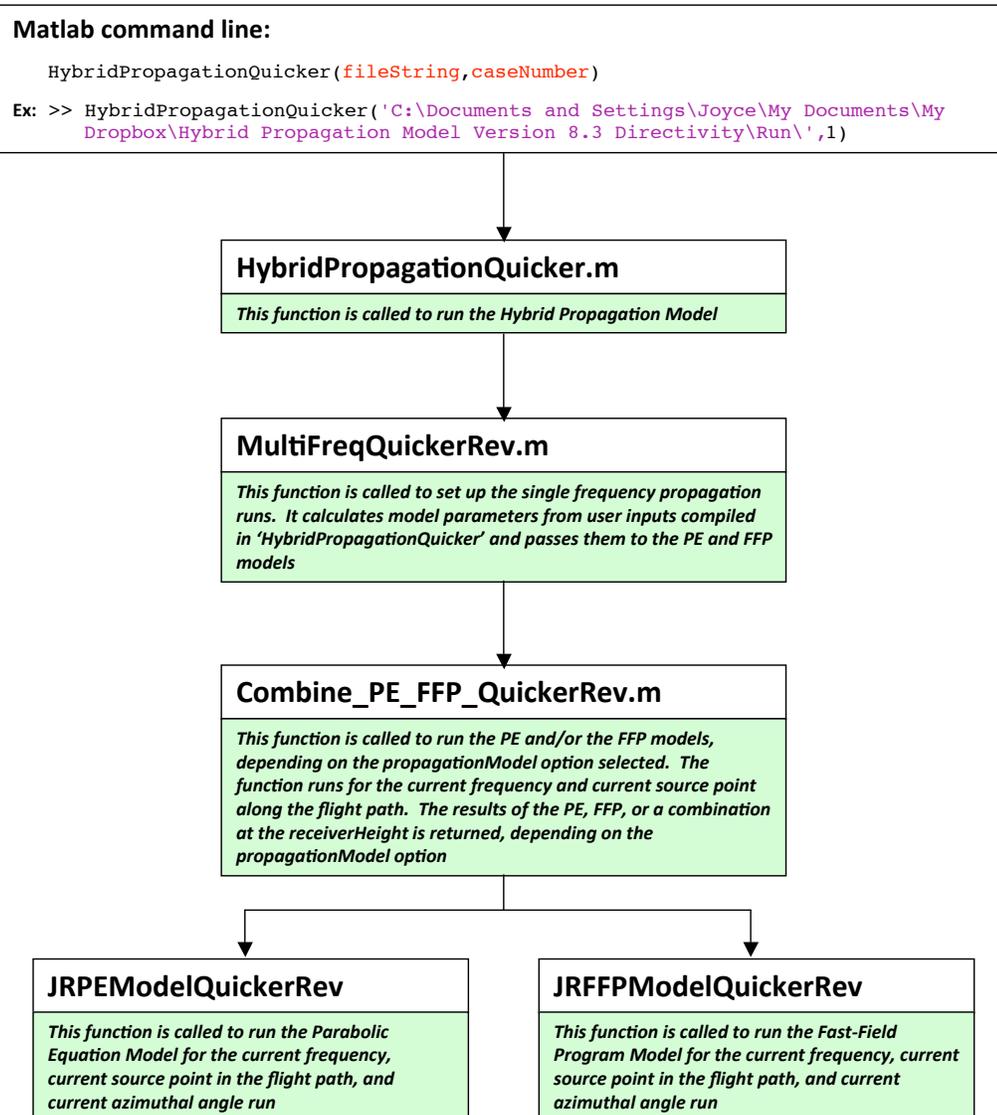
- characteristic of the atmosphere ('homogeneous' or 'refractive')
- sound speed at 0 m height c_0
- type of sound speed profile ('linear' or 'ln')
- sound speed parameter from thermal profile
- sound speed parameter from wind profile, in the direction of the wind
- direction of the wind in degrees (corresponding to the azimuthal angle toward which the wind vector points)
- aerodynamic roughness (ignored if ssp is linear)

Meteorology--Turbulence

- option to include turbulence ('yes' or 'no')
- number of realizations to include
- turbulence parameters A, K0, CTovT0, and Cvovc0

Meteorology--Atmospheric Absorption

- option for user-defined vs. model-calculated parameter ('user' or 'code')
- relative humidity
- absolute temperature



Matlab command line:

```
HybridPropagationQuicker(fileString,caseNumber)
```

Ex: >> HybridPropagationQuicker('C:\Documents and Settings\Joyce\My Documents\My Dropbox\Hybrid Propagation Model Version 8.3 Directivity\Run\','1')

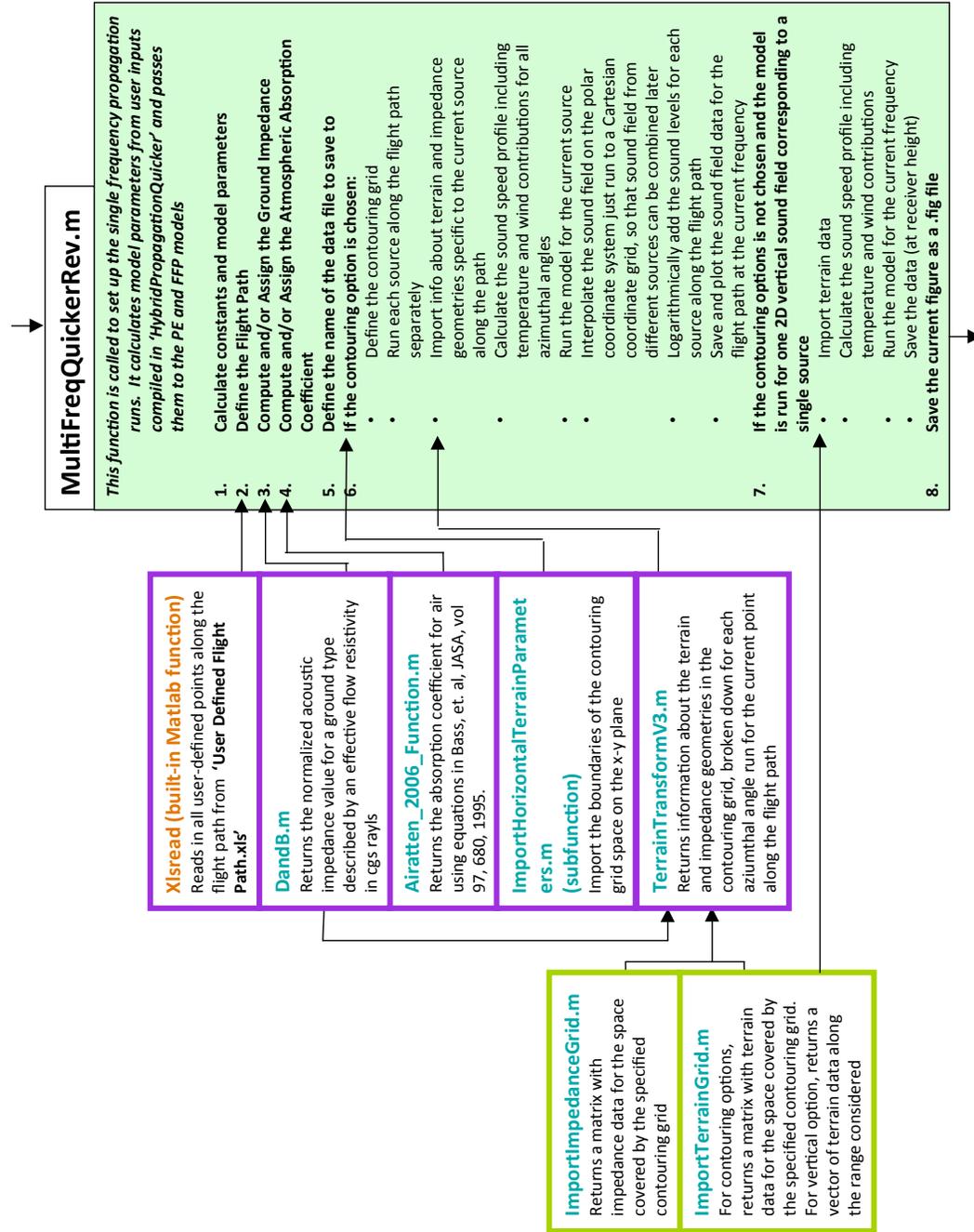
readText.m
Reads in all inputs from 'Hybrid Model Parameters.txt' (see right). Does not include any user-defined atm abs coefficients

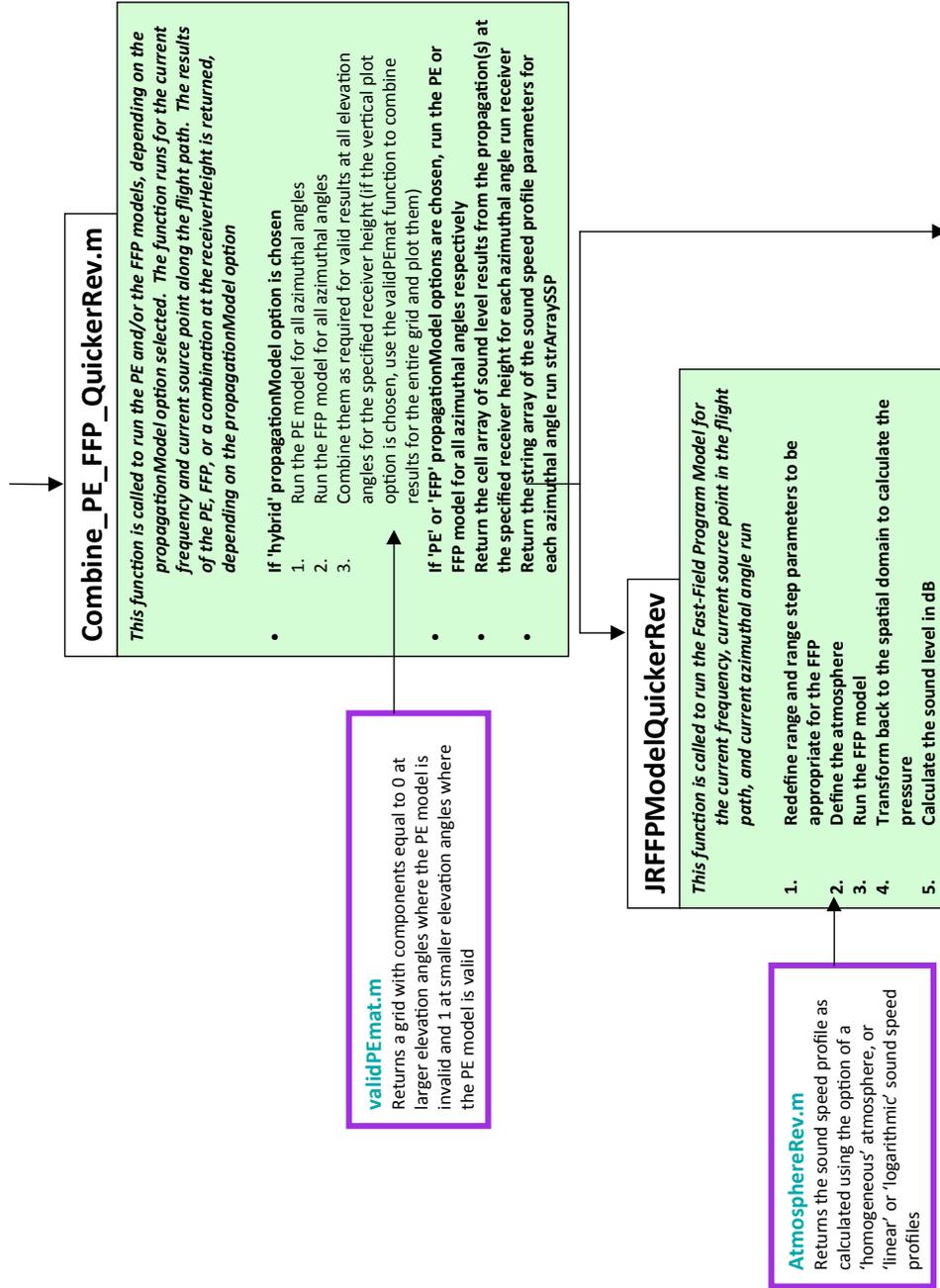
readAtmAbsText.m
Reads in all user-defined atm abs coefficients from 'User Defined Atm Abs Coeffs.xls'

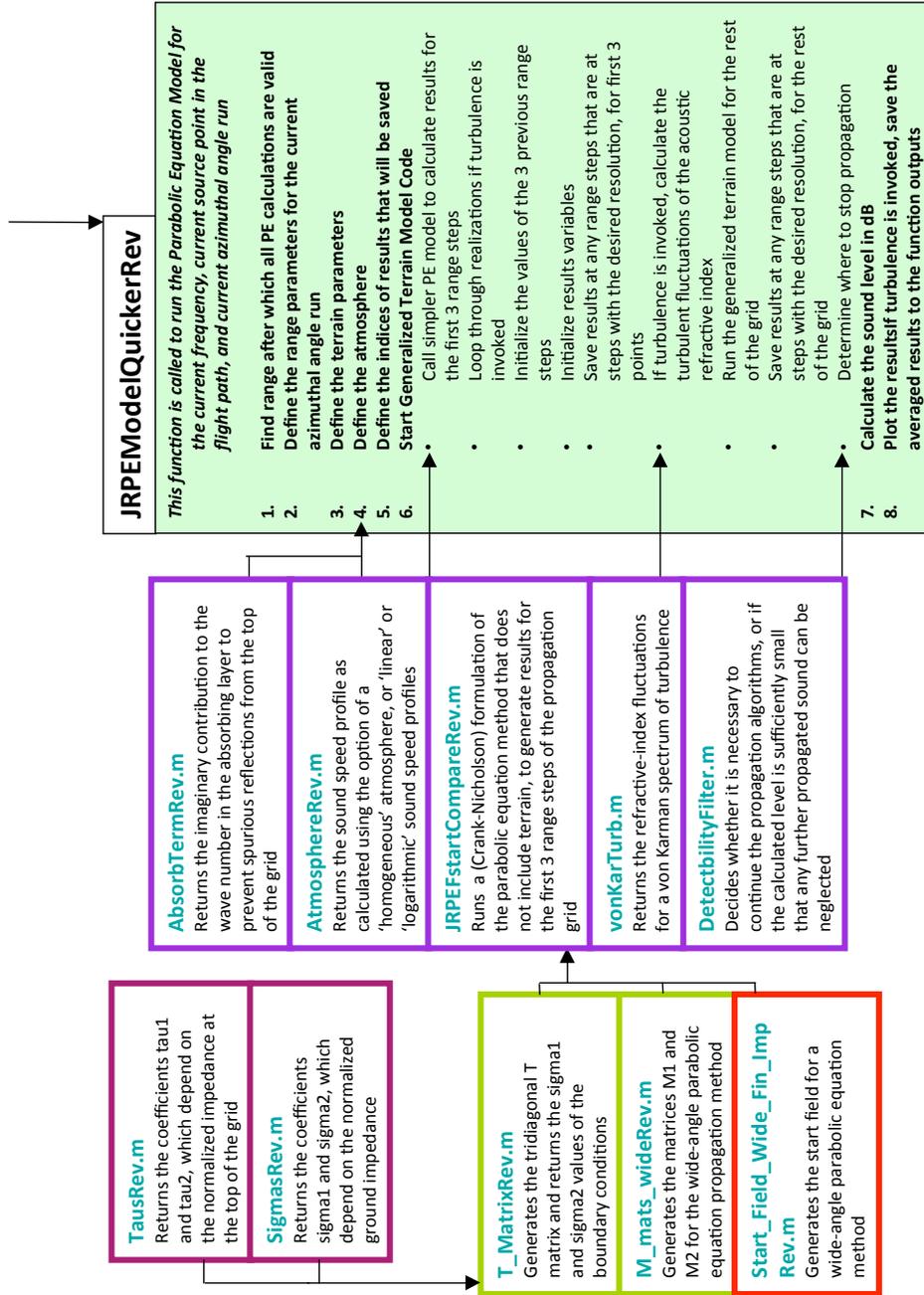
HybridPropagationQuicker.m

This function is called to run the Hybrid Propagation Model

1. Read .txt file with inputs
2. Define user input parameters to be passed through to model
 - Option for model (both PE and FFP) to produce a 2D vertical sound field figure
 - Vector of frequencies
 - Vector of atmospheric absorption parameters (defined only if user-defined option is chosen)
 - Vector of azimuthal angles for runs outward from the source
3. Run the model through all the frequencies in frequency vector
4. If the broadband results calculation is invoked, combine the results for the different frequency runs
5. Email user that code is finished running







Appendix **E**

Hybrid PE-FFP Model Code

E.1 Description of the Code Presentation

This appendix includes the hybrid PE-FFP model code. All functions used in the hybrid model are presented here in published Matlab form. The main functions appear first, followed by the supplemental functions in alphabetical order. Function inputs, processes, and outputs are summarized at the top of the function. The code, with comments, follows.

HybridPropagationQuicker Function

This function is called to run the Hybrid Propagation Model.

Inputs:

- *fileString* - the full name of the folder in which the user-input files are located (including an extra slash at the end). (For example, 'E:\Hybrid Propagation Model Version 5 All-Inclusive\Run1\')
- *caseNumber* - number of the Test Case being run. This input was added for easier labeling of test case run figures. It does not free the user of the responsibility to correctly input the Test Case run parameters. If the current run is not a test case, it has no real meaning.

Function Process:

1. Calls readText.m to read the user inputs
2. Defines the corresponding necessary parameters based on the model capabilities chosen by the user
3. Calls the MultiFreqQuickerRev.m function to run the rest of the model
4. If broadbandResults option is invoked, broadband levels are calculated, saved, and plotted

The function can catch errors and/or send an email to the user at the completion of the run.

(Note: 'Quicker' in the function name is a remnant of the original division of the code into 2 versions, a slower version that was able to plot the 2D vertical sound field, and a quicker version that would save only data at the specified receiver height. These capabilities have been combined into the current version which provides the user this choice in the 'Hybrid Model Parameters.txt' input file.)

Start Code Description:	1
Read in inputs from .txt file.....	2
Define user input parameters to be passed through to model.....	2
Run the model through all the frequencies in freqvector.....	3
If the broadband results calculation is invoked, combine the results for the different frequency runs	3
Email user that code is finished running.....	5

Start Code Description:

```
function [] = HybridPropagationQuicker(fileString,caseNumber)

%try
tic
```

Allow for the PC file naming convention that uses '\' and Mac convention of using '/',

```
indexFileString= strfind(fileString,'\');
if isempty(indexFileString)
    indexFileString= strfind(fileString, '/');
end
```

Identify the file path 'fileStringModel' that is used to locate the 'Auxiliary Functions' folder, and add that folder path so Matlab can access the necessary functions

```
fileStringModel= fileString(1:indexFileString(length(indexFileString)-1));

[fileStringModel 'Auxiliary Functions']
addpath([fileStringModel 'Auxiliary Functions']);
```

Read in inputs from .txt file

Read off the inputs from a correctly structured text file and assign appropriate variable input names:

```
[propagationModel,gridGeometry,axisymmetric,verticalPlot2D,levelat1m,directionalSource,discrete_or_bands,...
freqvector,startandstop,broadbandResults,range,height,deg,deltatheta,receiverHeight,deltarPlot,terrain,...

impedances,inputZorEFR,ZorEFR,lengthfr,atmosphere,czero,ctype,ssThermal,ssInWindDirection,windDirection,z
0,turbulence,...
realizN,A,K0,CTovT0,Cvovc0,user_or_code_AtmAbs,relativeHumidity,absTemperature]...
= readText([fileString 'Hybrid Model Parameters.txt']);
```

Define user input parameters to be passed through to model

These parameters can be defined at the highest level, using only user inputs. They are passed through the model so that they are calculated only once

Define the logical argument that converts user input to 1 (yes) 0 (no) dictating if model (both PE and FFP) should produce a 2D vertical sound field figure

```
if strcmp(verticalPlot2D,'yes')
    verticalPlot2D= 1;
elseif strcmp(verticalPlot2D,'no')
    verticalPlot2D= 0;
else error("The entry for the 'Vertical_2D_Plot' input category is invalid. Please enter either 'yes' or 'no'.")
end
```

Form the vector of specified frequencies to be run in the model

```
% Set of valid frequencies for model runs: 1/3 octave band center frequencies
% from 10 Hz to 10 kHz
fullbands= [10 12.5 16 20 25 31.5 40 50 63 80 100 125 160 200 250 315 400 500 ...
630 800 1000 1250 1600 2000 2500 3150 4000 5000 6300 8000 10000];
if strcmp(discrete_or_bands,'bands')
    startFreqIndex= find(fullbands==startandstop(1));
    stopFreqIndex= find(fullbands==startandstop(2));
    freqvector= fullbands(startFreqIndex:stopFreqIndex);
    if isempty(freqvector)
        error('Either the entered start or stop frequency is not a valid frequency band. Please enter the center frequency of a 1/3 octave band.')
    end
elseif strcmp(discrete_or_bands,'discrete')==0
```

```

    error('Please enter either "discrete" or "bands" for "discrete_or_bands:" input category');
end

```

Form the vector of atmospheric absorption parameters, defined only if the user chooses to define these parameters, rather than relying on the model to calculate them

```

betaAll= NaN;
if strcmp(user_or_code_AtmAbs,'user')
    [betaAll]= readAtmAbsText(fileString,freqvector);
end

```

Form the vector of azimuthal angles at which runs will be performed outward from the source, based on the user provided angle increment (only one angle is run if axisymmetry is assumed,

```

if or(strcmp(axisymmetric,'yes'),strcmp(gridGeometry,'vertical'))
    deltatheta= 2*pi;
elseif strcmp(axisymmetric,'no')
    deltatheta= deltatheta*(pi/180);
else error('The entry for the "Axisymmetric" input category is invalid. Please enter either "yes" or "no".')
end
thetas= 0:deltatheta:2*pi;

```

Run the model through all the frequencies in freqvector

```

for (freqnumber= 1:length(freqvector))
    freq= freqvector(freqnumber);

    MultiFreqQuickerRev(freq,freqnumber,propagationModel,gridGeometry,...
        verticalPlot2D,levelat1m,directionalSource,range,height,deg,deltatheta,thetas,...
        receiverHeight,deltaPlot,terrain,impedances,inputZorEFR,ZorEFR,...
        lengtfr,atmosphere,czero,ctype,ssThermal,ssInWindDirection,windDirection,z0,turbulence,realizN,A,...
        K0,CtoVt0,Cvove0,user_or_code_AtmAbs,relativeHumidity,absTemperature,...
        betaAll,fileString,caseNumber);
end

```

If the broadband results calculation is invoked, combine the results for the different frequency runs

The following section of code currently assumes that the spectrum is flat over all frequencies that were run. In more advanced versions of the code, this section could be made into a function that accepts a more realistic spectrum.

```

if strcmp(broadbandResults,'yes')
    bandwidths= [2 3 4 5 6 7 9 11 15 19 22 28 40 44 56 75 95 110 150 190 220 280 400 440 560 750 950 1100 1500 1900
2200]; % bandwidths for 10-10,000 Hz
    freqIndices= zeros(1,length(freqvector)); % initialize the vector that will contain the indices of the frequencies
used in this code run in the full
        % 1/3 octave band frequency vector from 10-10,000 Hz

```

Loop through the 1/3 octave band frequencies

```

for freqnumber= 1:length(freqvector)

    % Find the index of the current frequency in the full frequency
    % band

```

```

freqIndices(freqnumber)= find(fullbands==freqvector(freqnumber));

% Form the name of the file the results are stored in for the
% current frequency
if rem(freqvector(freqnumber),1)==0
    levelPlotName= strcat(gridGeometry,'grid_',propagationModel,'_' num2str(freqvector(freqnumber)), 'Hz');
    levelPlotName= levelPlotName{1,1};
else freqString= num2str(freqvector(freqnumber));
    indexfreqString= strfind(freqString,'. ');
    freqString= [freqString(1:indexfreqString-1) '_' freqString(indexfreqString+1)];
    levelPlotName= strcat(gridGeometry,'grid_',propagationModel,'_' freqString,'Hz');
    levelPlotName= levelPlotName{1,1};
end

```

Load the results file for the current frequency

```
load([fileString,levelPlotName,'.mat']);
```

Save the (intensity quantity) results for the current 1/3 octave band frequency, weighted by its bandwidth, in a matrix that stores all results (in the matrix's 3rd dimension, to allow for the 2D results of the horizontal plots).

```

% If the horizontal plot option is invoked, the variable
% 'levelPlotName' will contain a matrix. If only 1 azimuthal run
% was done, 'levelPlotName' will contain a cell array and must
% be accessed accordingly.
if strcmp(gridGeometry,'horizontal')
    str2evaluate= ['allFreq(:, :, freqnumber)=
bandwidths(freqIndices(freqnumber))*10.^(' ,levelPlotName, '/10);'];
else str2evaluate= ['allFreq(:, :, freqnumber)=
bandwidths(freqIndices(freqnumber))*10.^(' ,levelPlotName, '{1,1}/10);'];
end

eval(str2evaluate);
end

```

Calculate the level of the different frequency contributions, averaged together, weighted by their bandwidths.

```
tot= 10*log10(sum(allFreq,3)/sum(bandwidths(freqIndices)));
```

Save and plot the broadband averaged results.

```

% Form name of file to save broadband results in
broadbandName= strcat(gridGeometry,'grid_',propagationModel);
broadbandName= broadbandName{1,1};
% If the 'horizontal' option was employed
if strcmp(gridGeometry,'horizontal')

    % Save the broadband results with the xVals and yVals variables to
    % allow for easier future plotting
    save([fileString,broadbandName,'broadband.mat'],'tot','xVals','yVals','freqvector');

    % Show results in a 2D imagesc plot
    figure(89); imagesc(xVals,yVals,tot)
    axis equal;
    axis xy;
    xlabel('x-axis, range [m]', 'FontSize',32); ylabel('y-axis, range [m]', 'FontSize',32);
    if strcmp(discrete_or_bands,'bands')
        title(['Level [dB], ' num2str(freqvector(1)) '-' num2str(freqvector(end)) ' Hz in 1/3 Octave
Bands'], 'FontSize',32)
    else title(['Level [dB], ' mat2str(freqvector) ' Hz'], 'FontSize',32)
    set(gca,'FontSize',32);
end

```

```

% Otherwise, the 'vertical' option was employed
else
% Define the range vector the results are returned for
plotr= 0:deltarPlot:range-deltar;

% Save the broadband results with the plotr variable to
% allow for easier future plotting
save([fileString,broadbandName,'broadband.mat'],'tot','plotr','freqvector');

% Show results in a line plot
figure(89); plot(plotr,tot)
xlabel('range [m]',FontSize,32)
ylabel('Level [dB]',FontSize,32)
if strcmp(discrete_or_bands,'bands')
    title(['Level [dB], ' num2str(freqvector(1)) '-' num2str(freqvector(end)) ' Hz in 1/3 Octave
Bands'],FontSize,32)
else title(['Level [dB], ' mat2str(freqvector) ' Hz'],FontSize,32)
set(gca,FontSize,32);
end

end

end

toc

```

Email user that code is finished running

```

% Define these variables appropriately:
mail = 'jerosenbaum@gmail.com'; %Your GMail email address
password = '####'; %Your GMail password

% Then this code will set up the preferences properly:
setpref('Internet','E_mail',mail);
setpref('Internet','SMTP_Server','smtp.gmail.com');
setpref('Internet','SMTP_Username',mail);
setpref('Internet','SMTP_Password',password);
props = java.lang.System.getProperties();
props.setProperty('mail.smtp.auth','true');
props.setProperty('mail.smtp.socketFactory.class','javax.net.ssl.SSLSocketFactory');
props.setProperty('mail.smtp.socketFactory.port','465');

% Send the email
sendmail('jerosenbaum@gmail.com','Code is done','Hello! The MATLAB code is done running!!')

% Uncomment to send email if there is an error:
% catch
% mail = 'jerosenbaum@gmail.com'; %Your GMail email address
% password = '####'; %Your GMail password
%
% % Then this code will set up the preferences properly:
% setpref('Internet','E_mail',mail);
% setpref('Internet','SMTP_Server','smtp.gmail.com');
% setpref('Internet','SMTP_Username',mail);
% setpref('Internet','SMTP_Password',password);
% props = java.lang.System.getProperties();
% props.setProperty('mail.smtp.auth','true');
% props.setProperty('mail.smtp.socketFactory.class','javax.net.ssl.SSLSocketFactory');
% props.setProperty('mail.smtp.socketFactory.port','465');
%
% % Send the email
% sendmail('jerosenbaum@gmail.com','Uh oh...error','Bummer, there was an error')
% end

```

MultiFreqQuickerRev Function

This function is called to set up the single frequency propagation runs. It calculates model parameters from user inputs compiled in 'HybridPropagationQuicker' and passes them to the PE and FFP models

Inputs:

- *freq* - frequency used in this run of the model
- *freqnumber* - index of frequency used in this run in full frequency vector

- *propagationModel* - propagation model to be used ('PE', 'FFP', or 'hybrid')
- *gridGeometry* - final desired grid geometry ('vertical' or 'horizontal')
- *verticalPlot2D* - option for producing vertical 2D plot ('yes' or 'no')
- *levelat1m* - reference source level at 1 m distance
- *directionalSource* - option for using a source directivity pattern ('yes') or assuming an omnidirectional source ('no')

- *range* - grid range (used only for *gridGeometry* 'vertical'; the range is determined using subfunction *ImportHorizontalTerrainParameters* for *gridGeometry* 'horizontal'.)
- *height* - grid height
- *deg* - elevation angle limit used for PE
- *deltatheta* - increment angle for azimuth
- *thetas* - vector of azimuthal angles at which runs will be performed outward from the source
- *receiverHeight* - receiver height at which horizontal contour maps are calculated
- *deltarPlot* - range step for desired resolution of results

- *terrain* - characteristic of the terrain ('flat' or 'uneven')

- *impedances* - number of different impedances of the ground (1, 2, or 3)
- *inputZorEFR* - option to input impedance or the effective flow resistivity ('Z' or 'EFR')
- *ZorEFR* - vector with different impedances
- *lengthefr* - vector with range values at which the discontinuities occur

- *atmosphere* - characteristic of the atmosphere ('homogeneous' or 'refractive')
- *czero* - sound speed at 0 m height
- *ctype* - type of sound speed profile ('linear' or 'ln')
- *ssThermal* - sound speed parameter from thermal profile
- *ssInWindDirection* - sound speed parameter from wind profile, in the direction of the wind
- *windDirection* - direction of the wind in degrees (corresponding to the azimuthal angle toward which the wind vector points)
- *z0* - aerodynamic roughness (ignored if ssp is linear)

- *turbulence* - option to include turbulence ('yes' or 'no')
- *realizN* - number of realizations to include
- *A* , *K0* , *CTovT0* , *Cvovc0* - turbulence parameters A, K0, CTovT0, and Cvovc0

- *user_or_code_AtmAbs* - option for user to define parameter or allow model to calculate it ('user' or 'code')
- *relativeHumidity* - relative humidity
- *absTemperature* - absolute temperature
- *betaAll* - vector of atmospheric absorption parameters, empty if model-calculated, or full if user-calculated

- *fileString* - the full name of the folder in which the user-input files are located (including an extra slash at the end). (For example, 'E:\Hybrid Propagation Model Version 5 All-Inclusive\Run1\')
- *caseNumber* - number of the Test Case being run (this does not free the user of the responsibility to correctly input the Test Case run parameters. It is used mostly for figure labeling purposes)

Function Process:

1. Calculate constants and model parameters
2. Define the Flight Path
3. Compute and/or Assign the Ground Impedance
4. Compute and/or Assign the Atmospheric Absorption Coefficient
5. Define the name of the data file to save to at the end of the function
6. If the contouring option is chosen:
 - Define the contouring grid
 - Run each source along the flight path separately
 - Import info about terrain and impedance geometries specific to the current source along the path
 - Calculate the sound speed profile for all azimuthal angles, including contributions of both temperature and wind
 - Run the model for the current source
 - Interpolate the sound field on the polar coordinate system to a Cartesian coordinate grid, so that sound field from different sources can be combined later
 - Logarithmically add the sound levels for each source along the flight path
 - Save and plot the sound field data for the flight path at the current frequency

7. If the contouring options is not chosen and the model is run for one 2D vertical sound field corresponding to a single source:

- Import terrain data
- Calculate the sound speed profile, including contributions of both temperature and wind
- Run the model for the current frequency
- Save the data (at receiver height)

8. Save the current figure as a .fig file

This function saves the 2D contouring grid matrix or the 2D vertical run data at the specified receiver height with the name:

[gridGeometry,'grid_',propagationModel,'_',freqString,'Hz' '.mat'] where gridGeometry can be 'vertical' or 'horizontal', propagationModel can be 'hybrid', 'PE', or 'FFP', and 'freqString' is the frequency.

Start Code Description:	3
Calculate constants and model parameters.....	3
Define the flight path, reading in 'User Defined Flight Path.xls'.....	4
Compute and/or assign the ground impedance.....	4
Compute and/or assign the atmospheric absorption coefficient.....	5
Define the name of the data file to save to at the end of the function	5
Contouring (horizontal sound field) runs.....	5
2D Vertical sound field runs (single source, no contouring).....	8
Save the current figure as a .fig file.....	9
ImportHorizontalTerrainParameters Subfunction	9

Start Code Description:

```
function []=
MultiFreqQuickerRev(freq,freqnumber,propagationModel,gridGeometry,verticalPlot2D,levelat1m,directionalSource,range,height,deg,deltatheta,thetas,receiverHeight,deltaPlot,terrain,impedances,...

inputZorEFR,ZorEFR,lengthfr,atmosphere,czero,ctype,ssThermal,ssInWindDirection,windDirection,z0,turbulence,realizN,A,K0,CTovT0,Cvovc0,user_or_code_AtmAbs,...
relativeHumidity,absTemperature,betaAll,fileString,caseNumber)
```

Calculate constants and model parameters

Angular frequency, reference wave number, reference wavelength, range grid spacing, height grid spacing, the range vector, number of range grid points, height vectors for both PE and FFP, and number of height grid points for both PE and FFP are defined. The height is redefined to satisfy the requirements of the PE in terms of number of height points and upper absorbing layer.

Constraints:

- The range spacing must be a divisor of deltarPlot (spacing must be able to multiply an integer to get deltarPlot). This allows each frequency to be defined at common points so that results of different frequencies can be easily added together. However, if deltarPlot is too small as compared to a wavelength, numerical errors may result for uneven terrain cases (at lower frequencies).
- There must be at least 1000 vertical grid points, including the absorbing layer.
- The absorbing layer, at the top of the grid, must be at least 50 wavelengths high.

```

omega= 2*pi*freq;
kzero= omega/czero;
lambda= 2*pi/kzero;

deltarlim = .1*lambda;           %deltar must be less than 1/10 of a wavelength
deltar= deltarPlot/ceil(deltarPlot/deltarlim);           %determine deltar that will be divisor of deltarPlot
clear deltarlim

deltazlim = .1*lambda;           %deltaz must be less than 1/10 of a wavelength
deltaz= receiverHeight/ceil(receiverHeight/deltazlim); %determine deltaz that will be divisor of receiver height

height= ceil(height/deltaz)*deltaz;           %corrected height so that height is integer multiple of deltaz and
deltaz is divisor of receiver height
clear deltazlim

% Set number of points in r and z directions:
absorbLength= ceil(50*lambda/deltaz)*deltaz; %length of absorbing layer, larger than 50 wavelengths
if (height+absorbLength)/deltaz<1000; %if total height with absorbing layer has 1000 points or less
    heightwabs= 1000*deltaz; %make the height with absorbing layer have 1001 points
    height= heightwabs-absorbLength; %and redefine the height without the absorbing layer
else heightwabs= height+absorbLength; %otherwise, leave dimensions as they are
end

% Define the z-axis
z= [0:deltaz:heightwabs];
zN= length(z); %number of height points for PE (includes absorbing layer)

z_FFP= [0:deltaz:height];
zN_FFP= length(z_FFP); %number of height points for FFP (excludes absorbing layer)

```

Define the flight path, reading in 'User Defined Flight Path.xls'

A matrix with 3 columns, and sourceN rows, where sourceN is the number of points along the flight path, is read into the variable flightTrack. The first, second, and third columns of flightTrack correspond to the x, y, and z coordinates of each represented point along the flight path.

```

[flightTrack]= xlsread([fileString 'User Defined Flight Path.xls']);
xflightPositions= flightTrack(:,1).';
yflightPositions= flightTrack(:,2).';
zflightPositions= flightTrack(:,3).';

```

Compute and/or assign the ground impedance

```

if strcmp(inputZorEFR,'EFR')
    [Z]= DandB(freq,ZorEFR);
elseif strcmp(inputZorEFR,'Z')
    Z= ZorEFR;
else error('Please enter either "Z" or "EFR" for "Input_Z_or_EFR:" input category')
end

```

Compute and/or assign the atmospheric absorption coefficient

If the user chooses the option to manually assign the atmospheric absorption coefficients, the defined coefficient corresponding to the current frequency is used. If the option of letting the model calculate the coefficient is used, the model will calculate using a script based on equations in Bass, et. al, JASA, vol 97, 680, 1995 for the current frequency, and the specified relative humidity and absolute temperature in K

```

if strcmp(user_or_code_AtmAbs,'user')
    betaAtmAbs= betaAll(freqnumber);
elseif strcmp(user_or_code_AtmAbs,'code')
    [betaAtmAbs]= airatten_2006_Function(freq,relativeHumidity,absTemperature);
else error('The entry for the "User_or_code_defined" input category is invalid. Please enter either "user" or "code".')
end
end

```

Define the name of the data file to save to at the end of the function

Name is defined as [gridGeometry,'grid_',propagationModel,'_',freqString,'Hz' '.mat']. For example, 'verticalgrid_PE_500Hz.mat'

```

if rem(freq,1)==0
    levelPlotName= strcat(gridGeometry,'grid_',propagationModel,'_',num2str(freq),'Hz');
    levelPlotName= levelPlotName{1,1};
else freqString= num2str(freq);
    indexfreqString= strfind(freqString,'. ');
    freqString= [freqString(1:indexfreqString-1) '_' freqString(indexfreqString+1)];
    levelPlotName= strcat(gridGeometry,'grid_',propagationModel,'_',freqString,'Hz');
    levelPlotName= levelPlotName{1,1};
end

```

Contouring (horizontal sound field) runs

For gridGeometry= 'horizontal'

```

if strcmp(gridGeometry,'horizontal')

```

Define 2D horizontal (Cartesian) grid for contours

```

% Define the ranges of x and y values the contouring grid will cover
[xminTerrain,xmaxTerrain,yminTerrain,ymaxTerrain]= ImportHorizontalTerrainParameters();

% Set the x and y vectors
xVals= xminTerrain:deltarPlot:xmaxTerrain;
xValsN= length(xVals);
yVals= yminTerrain:deltarPlot:ymaxTerrain;
yValsN= length(yVals);

%Construct the contouring grid
[Xgrid,Ygrid]= meshgrid(xVals,yVals);

sourceN= length(xflightPositions); % number of represented points along the flight path
runN= length(thetas); % number of runs needed to cover all azimuthal angles from source

LevelAll= zeros(yValsN,xValsN,sourceN); % initialize matrix to contain contour

```

% data for each source

Run the model for all the represented points along the flight path

for (l=1:sourceN)

Import information about the terrain and impedance geometries in the contouring grid, broken down for each azimuthal angle run for the current point along the flight path

Call to TerrainTransformV3 returns

- a vector specifying the necessary range for runs at each azimuthal angle (distance from source to boundary will change with angle) *rangesAtThetas*
- a cell array of terrain heights that includes different cells for each azimuthal angle *polarTerrain*
- a vector of the number of impedances to be included for each azimuthal run *impedancesAtThetas*
- a matrix with the values of the different impedances for each azimuthal run *ZAtThetas*
- a matrix with the ranges at which the impedance transitions occur for each azimuthal run *lengthfrAtThetas*

```
[rangesAtThetas,polarTerrain,impedancesAtThetas,ZAtThetas,lengthfrAtThetas]=
TerrainTransformV3(thetas,runN,xflightPositions(l),yflightPositions(l),freq,deltar,deltarPlot,inputZorEFR,xminTerrain,xmaxTerrain,yminTerrain,ymaxTerrain,terrain);
```

Calculate the sound speed profile, including contributions of both temperature and wind

The sound speed parameter for the effective sound speed profile, including both temperature and wind contributions are calculated. Currently it is calculated for a non-range-dependent sound speed profile. If a range-dependent profile is needed, a function similar to 'TerrainTransformV3' should be constructed to interpolate the wind vector (magnitude--*ssInWindDirection* and direction--*windDirection*) broken down at each range point for each azimuthal run for the current point along the flight path. (These lines of code were placed inside the flight path 'for' loop with the consideration that range dependence may be added in. For non-range-dependent conditions, it will be the same for each source and could be calculated once for all points, outside the loop.)

```
ssWind= ssInWindDirection*cos(thetas-(windDirection*pi/180)); % vector of sound speed profile parameters due
to wind for each azimuthal angle
ssGradientAtThetas= ssWind+ssThermal; % vector of the combined contribution of wind and thermal profiles--
assumes use of the same profile type (linear or ln)
```

Run the model for the current frequency and source point along the flight path

```
[receiver,strArraySSP]=
Combine_PE_FFP_QuickerRev(propagationModel,verticalPlot2D,directionalSource,terrain,atmosphere,impedance
sAtThetas,freq,czero,rangesAtThetas,height,z0,...
zflightPositions(l),deltarPlot,deg,ZAtThetas,lengthfrAtThetas,ctype,ssGradientAtThetas,receiverHeight,...
```

```
runN,omega,kzero,deltar,deltaz,heightwabs,zN,z,zN_FFP,z_FFP,betaAtmAbs,turbulence,realizN,A,K0,CTovT0,Cv
ovc0,levelat1m,caseNumber,polarTerrain);
```

Prepare grid and matrices necessary for combining sound fields for different sources, and plotting

```
[thetaGrid,rGrid]= cart2pol(Xgrid-xflightPositions(l),Ygrid-yflightPositions(l));
%convert plotting grid to polar coordinate system

thetaGrid(1:ceil(abs(yminTerrain-yflightPositions(l))/deltarPlot),:)= (2*pi)-
abs(thetaGrid(1:ceil(abs(yminTerrain-yflightPositions(l))/deltarPlot),:)); %convert thetaGrid so that it goes from
0 to 2*pi, instead of including positive and negative theta values

Level= zeros(yValsN,xValsN); %sound field for the current source on the flight path at the current frequency
```

Interpolate from polar coordinate grid to Cartesian coordinate grid

```
for mm= 10:yValsN-10 %represent the y axis. Start 30 late and end 10 early because of transformation from
radial interpolated terrain and impedance input. 10 is relatively arbitrary.
for n= 10:xValsN-10 %represent the x axis
```

```
if rem(rGrid(mm,n),deltarPlot)==0
if rem(thetaGrid(mm,n),deltatheta)==0
%if the cartesian grid point falls exactly on a polar
%gridpoint, then the level at that cartesian gridpoint
%equals the level at the corresponding polar gridpoint
Level(mm,n)= receiver{round(thetaGrid(mm,n)/deltatheta+1)}(round(rGrid(mm,n)/deltarPlot+1));
else
%otherwise, if the cartesian gridpoint falls exactly on the
%r value, but between two theta values of the polar
%coordinate system:
%set the two polar coordinate system theta values that this theta is between
floorTheta= deltatheta*floor(thetaGrid(mm,n)/deltatheta);
ceilTheta= floorTheta+deltatheta;
%find the coordinates A and B where A and B can be found in the
%Numerical Recipes book, p. 107
Ainterp= (ceilTheta-thetaGrid(mm,n))/deltatheta;
Binterp= 1-Ainterp;
%interpolate the level at the r value between to two values
%of theta
Level(mm,n)=
10*log10(Ainterp*10^(receiver{round(floorTheta/deltatheta+1)}(round(rGrid(mm,n)/deltarPlot+1))/10)+...
Binterp*10^(receiver{round(ceilTheta/deltatheta+1)}(round(rGrid(mm,n)/deltarPlot+1))/10));
end
elseif rem(thetaGrid(mm,n),deltatheta)==0
%if the cartesian gridpoint falls between two r values in the
%polar coordinate system but exactly on a theta value:
%set the two polar coordinate system r values that this r is between
floorR= deltarPlot*floor(rGrid(mm,n)/deltarPlot);
ceilR= floorR+deltarPlot;
%find the coordinates A and B where A and B can be found in the
%Numerical Recipes book, p. 107
Ainterp= (ceilR-rGrid(mm,n))/deltarPlot;
Binterp= 1-Ainterp;
%interpolate the level at the theta value between to two values
%of r
Level(mm,n)=
10*log10(Ainterp*10^(receiver{round(thetaGrid(mm,n)/deltatheta+1)}(round(floorR/deltarPlot+1))/10)+...
Binterp*10^(receiver{round(thetaGrid(mm,n)/deltatheta+1)}(round(ceilR/deltarPlot+1))/10));
else
%if the cartesian gridpoint falls between both two r values and
%two theta values in the polar coordinate system:
%set the two polar coordinate system theta values that this theta is between
floorTheta= deltatheta*floor(thetaGrid(mm,n)/deltatheta);
ceilTheta= floorTheta+deltatheta;
%set the two polar coordinate system r values that this r is between
floorR= deltarPlot*floor(rGrid(mm,n)/deltarPlot);
```

```

    ceilR= floorR+deltarPlot;

    %find the coordinates A and B along an r where A and B can be found in the
    %Numerical Recipes book, p. 107
    AinterpR= (ceilTheta-thetaGrid(mm,n))/deltatheta;
    BinterpR= 1-AinterpR;
    %interpolate the level at the lower r value between to two values
    %of theta

    LevelLR=
    10*log10(AinterpR*10^(receiver{round(floorTheta/deltatheta+1),1}(1,round(floorR/deltarPlot+1))/10)+...
    BinterpR*10^(receiver{round(ceilTheta/deltatheta+1),1}(1,round(floorR/deltarPlot+1))/10));
    %interpolate the level at the upper r value between to two values
    %of theta
    LevelUR=
    10*log10(AinterpR*10^(receiver{round(floorTheta/deltatheta+1),1}(1,round(ceilR/deltarPlot+1))/10)+...
    BinterpR*10^(receiver{round(ceilTheta/deltatheta+1),1}(1,round(ceilR/deltarPlot+1))/10));
    %find the coordinates A and B between the r's at the correct theta value
    %where A and B can be found in the Numerical Recipes book, p. 107
    AinterpTheta= (ceilR-rGrid(mm,n))/deltarPlot;
    BinterpTheta= 1-AinterpTheta;
    %interpolate the level at the correct theta value between to
    %two values of r
    Level(mm,n)= 10*log10(AinterpTheta*10^(LevelLR/10)+...
    BinterpTheta*10^(LevelUR/10));
end
end
end
LevelAll(:,:,l)= Level;
end

```

Logarithmically add the sound levels for each source along the flight path

```
levelTotal= 10*log10(sum(10.^(LevelAll/10),3));
```

Save the sound field data for the flight path at the current frequency

```
eval([levelPlotName '= levelTotal;']);
save([fileString levelPlotName '.mat'],levelPlotName,'xVals','yVals');
```

Plot the sound field data for the flight path at the current frequency

```
figure(1);
imagesc(xVals,yVals,levelTotal);
xlabel('range [m]')
ylabel('range [m]')
title('Transmission Loss [dB] for Flight Path')
colorbar
%caxis([20,90])
axis xy
axis equal
xlim([xminTerrain+30,xmaxTerrain-30]);ylim([yminTerrain+30,ymaxTerrain-30])

```

2D Vertical sound field runs (single source, no contouring)

```
elseif strcmp(gridGeometry,'vertical')
    %I've reached the call to Combine'

```

Import terrain data

```
if strcmp(terrain,'uneven')
    [terrainVector]= ImportTerrainGrid(range,deltar);
    terrainGrid{1}= terrainVector;

```

```
elseif strcmp(terrain,'flat')
    terrainGrid{1}= zeros(1,range/deltar+1);
else error('Please enter either "uneven" or "flat" in the "Terrain" user-input');
end
```

Calculate the sound speed profile, including contributions of temperature and wind

Assumes the parameter for wind speed, `ssInWindDirection`, reflects the strength of the wind in the direction being considered.

```
ssGradient= ssInWindDirection+ssThermal; % combined contribution of wind and thermal profiles--assumes use
of the same profile type (linear or ln)
```

Run the model for the current frequency

```
[receiver,strArraySSP]=
Combine_PE_FFP_QuickerRev(propagationModel,verticalPlot2D,directionalSource,terrain,atmosphere,impedance
s,freq,czero,range,height,z0,...
zflightPositions(1),deltarPlot,deg,Z,lengtheFr,ctype,ssGradient,receiverHeight,...

2,omega,kzero,deltar,deltaz,heightwabs,zN,z,zN_FFP,z_FFP,betaAtmAbs,turbulence,realizN,A,K0,CTovT0,Cvovc0
,levelat1m,caseNumber,terrainGrid);
```

Save the data (at receiver height) at the current frequency

```
eval([levelPlotName '= receiver;']);
save([fileString levelPlotName '.mat'],levelPlotName,'deltar');

else error('The entry for the "Grid_Geometry" input category is invalid. Please enter either "horizontal" or
"vertical".')

end
```

Save the current figure as a .fig file

Figure is saved in the same folder as the .mat results and the user input files.

```
saveas(gcf,[fileString levelPlotName '_figure.fig'])
```

ImportHorizontalTerrainParameters Subfunction

This subfunction is called to import the boundaries of the contouring grid space on the x-y plane

`xminTerrain` and `xmaxTerrain` are the smallest and largest x-axis values for which terrain and impedance data is available. `yminTerrain` and `ymaxTerrain` are the smallest and largest y-axis values for which terrain and impedance data is available.

```
%=====
function [xminTerrain,xmaxTerrain,yminTerrain,ymaxTerrain]= ImportHorizontalTerrainParameters()

% The body of this function is a placeholder until a better way of
% inputting the data is determined
xminTerrain= -1300;
xmaxTerrain= 500;
yminTerrain= -500;
```

ymaxTerrain= 700;

Combine_PE_FFP_QuickerRev Function

This function is called to run the PE and/or the FFP models, depending on the propagationModel option selected. The function runs for the current frequency and current source point along the flight path. It runs the model for all azimuthal angles, as defined by the user. The results of the PE, FFP, or a combination at the receiverHeight are returned, depending on the propagationModel option

Inputs:

- *propagationModel* - propagation model to be used ('PE', 'FFP', or 'hybrid')
- *verticalPlot2D* - option for producing vertical 2D plot ('yes' or 'no')
- *directionalSource* - option for using a source directivity pattern ('yes') or assuming an omnidirectional source ('no')
- *terrain* - characteristic of the terrain ('flat' or 'uneven')
- *atmosphere* - characteristic of the atmosphere ('homogeneous' or 'refractive')
- *impedancesAtThetas* - a vector of the number of impedance to be included for each azimuthal run for the current source point along the flight path
- *freq* - frequency used in this run of the model
- *czero* - sound speed at 0 m height
- *rangesAtThetas* - a vector specifying the necessary range for runs at each azimuthal angle for the current source point (distance from source to boundary will change with angle)
- *height* - grid height (not including PE absorbing layer)
- *z0* - aerodynamic roughness (ignored if ssp is linear)
- *zflightPositions* - z-coordinate of the current source point along the flight path.
- *deltarPlot* - range step for desired resolution of results
- *deg* - elevation angle limit used for PE
- *ZAtThetas* - a matrix with the values of the different impedances for each azimuthal run for the current source point
- *lengthefrAtThetas* - a matrix with the ranges at which the impedance transitions occur for each azimuthal run for the current source point
- *ctype* - type of sound speed profile ('linear' or 'ln')
- *ssGradientAtThetas* - sound speed parameter at all azimuthal angles
- *receiverHeight* - receiver height at which horizontal contour maps are calculated
- *runN* - number of runs needed to cover all azimuthal angles from source
- *omega* - angular frequency used in this run of the model
- *kzero* - reference wave number used in this run of the model
- *deltar* - horizontal grid spacing used in the PE propagation model, dependent on frequency. Also constrained to be a divisor of number chosen in 'MultiFreqQuickerRev.m'
- *deltaz* - vertical grid spacing used in the PE and FFP propagation models. Constrained to be a divisor of the receiverHeight so that the receiverHeight will fall exactly at a vertical grid point
- *heightwabs* - grid height for PE including PE absorbing layer
- *zN* - number of height points in grid for PE (includes absorbing layer)

- *z* - height vector for PE
- *zN_FFP* - number of height points in grid for FFP (excludes absorbing layer)
- *z_FFP* - height vector for FFP
- *betaAtmAbs* - atmospheric absorption coefficient for current frequency
- *turbulence* - option to include turbulence ('yes' or 'no')
- *realizN* - number of realizations to include
- *A*, *K0*, *CTovT0*, *Cvovc0* - turbulence parameters A, K0, CTovT0, and Cvovc0
- *levelat1m* - reference source level at 1 m distance from the current point source
- *caseNumber* - number of the Test Case being run (this does not free the user of the responsibility to correctly input the Test Case run parameters. It is used mostly for figure labeling purposes)
- *polarTerrain* - a cell array of terrain heights that includes different cells for each azimuthal angle

Function Process and Outputs:

- If 'hybrid' propagationModel option is chosen:
 1. Assign necessary parameters for each azimuthal angle (if the vertical plot option is chosen, return entire vertical run grid, otherwise only return results at the receiver height)
 2. Run the FFP model for all azimuthal angles (if the vertical plot option is chosen, return entire vertical run grid, otherwise only return results at the receiver height)
 3. Combine them as required for valid results at all elevation angles for the specified receiver height (if the vertical plot option is chosen, use the validPEmat function to combine results for the entire grid and plot them)
- If 'PE' or 'FFP' propagationModel options are chosen, run the PE or FFP model for all azimuthal angles respectively
- Return the cell array of sound level results from the propagation(s) at the specified receiver height for each azimuthal angle run, *receiver*
- Return the string array of the sound speed profile parameters (stored as string) for each azimuthal angle run, *strArraySSP*

Start Code Description:	2
Using the hybrid PE-FFP model	3
Using the only the PE model	4
Using the only the FFP model	5
Plot the results	5

Start Code Description:

```
function [receiver,strArraySSP]=
Combine_PE_FFP_QuickerRev(propagationModel,verticalPlot2D,directionalSource,terrain,atmosphere,impedance
sAtThetas,freq,czero,rangesAtThetas,height,z0,...
zflightPositions,deltaPlot,deg,ZAtThetas,lengthfrAtThetas,ctype,ssGradientAtThetas,receiverHeight,...
```

```

runN,omega,kzero,deltar,deltaz,heightwabs,zN,z,zN_FFP,z_FFP,betaAtmAbs,turbulence,realizN,A,K0,CTovT0,Cv
ovc0,levelat1m,caseNumber,polarTerrain)

receiver= cell(runN,1); %initialize the cell array to contain results for the runs at different azimuthal angles

```

Using the hybrid PE-FFP model

```

if strcmp(propagationModel,'hybrid')==1 %if the 'hybrid' propagationModel is invoked

```

Assign necessary parameters for each azimuthal angle

(For the current source along the flight path)

```

for n= 1:runN-1 %only to runN-1 because the first (angle= 0) and last (angle= 2*pi) azimuthal angle runs will be
the same
    n
    % Set necessary input variables
    range= rangesAtThetas(n)-deltar; %Range for model to propagate out to. A
deltar is subtracted because polarTerrain includes one less range value terrain point (due to limits of interpolation).
    H= polarTerrain{n}; %Terrain data for radial slice at theta(n)
    impedances= impedancesAtThetas(n,1); %Number of impedances for radial slice
at theta(n)
    Z= ZAtThetas(n,:); %Vector of impedance values for radial slice at
theta(n)
    lengthfr= lengthfrAtThetas(n,:); %Vector of impedance discontinuity indexes
for radial slice at theta(n)
    ssGradient= ssGradientAtThetas(n); % effective sound speed parameter at angle theta(n)

```

Call the PE model

(For the current azimuthal angle and current source point along the flight path)

If the vertical 2D plot option is invoked, an additional output argument containing the full 2D vertical grid results will be returned

```

if verticalPlot2D==1
    [receiver{n,1},grad,lowPEIntercept,highPEIntercept,cutoffrange,gridsq]=
JRPEModelQuickerRev(verticalPlot2D,directionalSource,terrain,atmosphere,impedances,freq,czero,range,height,.
..
z0,zflightPositions,deltarPlot,deg,Z,lengthfr,ctype,ssGradient,receiverHeight,n,omega,kzero,deltar,deltaz,heightwa
bs,zN,z,betaAtmAbs,turbulence,realizN,A,K0,CTovT0,Cvovc0,levelat1m,H);
    else [receiver{n,1},grad,lowPEIntercept,highPEIntercept,cutoffrange]=
JRPEModelQuickerRev(verticalPlot2D,directionalSource,terrain,atmosphere,impedances,freq,czero,range,height,.
..
z0,zflightPositions,deltarPlot,deg,Z,lengthfr,ctype,ssGradient,receiverHeight,n,omega,kzero,deltar,deltaz,heightwa
bs,zN,z,betaAtmAbs,turbulence,realizN,A,K0,CTovT0,Cvovc0,levelat1m,H);
    end
    strArraySSP(n)= cellstr(grad);

```

Call the FFP model

(For the current azimuthal angle and current source point along the flight path)

If the vertical 2D plot option is invoked, an additional output argument containing the full 2D vertical grid results will be returned

```

if verticalPlot2D==1
    [receiverFFP,receiverFFPgrid]=
    JRFFPModelQuickerRev(verticalPlot2D,atmosphere,freq,czero,min(max(lowPEIntercept,highPEIntercept),range
    ),height,z0,zflightPositions,Z(1),ctype,ssGradient,...
        receiverHeight,n,omega,kzero,deltarPlot,deltar,deltaz,zN_FFP,z_FFP,betaAtmAbs,levelat1m); %with
    range of max(lowPEIntercept,highPEIntercept)
    else [receiverFFP]=
    JRFFPModelQuickerRev(verticalPlot2D,atmosphere,freq,czero,min(max(lowPEIntercept,highPEIntercept),range
    ),height,z0,zflightPositions,Z(1),ctype,ssGradient,...
        receiverHeight,n,omega,kzero,deltarPlot,deltar,deltaz,zN_FFP,z_FFP,betaAtmAbs,levelat1m); %with
    range of max(lowPEIntercept,highPEIntercept)
end

```

Combine the PE and FFP results at the specified receiver height

Use PE where valid and FFP at larger elevation angles

```

validRangeStartN= ceil(abs(zflightPositions-receiverHeight)/(tan(deg*pi/180)*deltarPlot)+1); %number of
points into range at which PE starts to be valid at receiverHeight

receiver{n}(1:min(validRangeStartN,length(receiverFFP)))=
zeros(1,min(validRangeStartN,length(receiverFFP))); %set invalid range steps to zero
receiver{n}(1:min(validRangeStartN,length(receiverFFP)))=
receiver{n}(1:min(validRangeStartN,length(receiverFFP)))+receiverFFP(1:min(validRangeStartN,length(receiver
FFP))); %add in FFP results

```

Combine the PE and FFP results for the entire grid if the vertical sound field option is invoked

Use PE where valid and FFP at larger elevation angles

```

if verticalPlot2D==1

    % Combine the PE and FFP results
    vPE= validPEmat(zflightPositions,deg,deltarPlot,deltaz,range,height);
    grid_PE_FFP= gridsq(1:zN_FFP,:).*vPE+[receiverFFPgrid zeros(zN_FFP,size(gridsq,2)-
    size(receiverFFPgrid,2)).*(1-vPE);

    % Plot the vertical sound field
    figure(n+3)
    imagesc(0:deltarPlot:range-deltar,z_FFP,grid_PE_FFP);
    colormap(jet)
    colorbar;
    axis xy
    ylim([0,height])
    axis square
    xlabel('range [m]','fontsize',22)
    ylabel('height [m]','fontsize',22)
    title(['Hybrid model, source (0,',int2str(zflightPositions),') f=',int2str(freq),' Hz, 'grad'],'fontsize',22);
end
end

```

Using the only the PE model

```
elseif strcmp(propagationModel,'PE')==1
```

Assign necessary parameters for each azimuthal angle

(For the current source along the flight path)

```

    for n= 1:runN-1 %only to runN-1 because the first (angle= 0) and last (angle= 2*pi) azimuthal angle runs will be
    the same
        n
        % Set necessary input variables
        range= rangesAtThetas(n)-deltar; %Range for model to progagate out to. A
        %deltar is subtracted because polarTerrain includes one less range value terrain point (due to limits of interpolation).
        H= polarTerrain{n}; %Terrain data for radial slice at theta(n)
        impedances= impedancesAtThetas(n,1); %Number of impedances for radial slice
        at theta(n)
        Z= ZAtThetas(n,:); %Vector of impedance values for radial slice at
        theta(n)
        lengthfr= lengthfrAtThetas(n,:); %Vector of impedance discontinuity
        distances for radial slice at theta(n)
        ssGradient= ssGradientAtThetas(n); % effective sound speed parameter at angle theta(n)
    end

```

Call the PE model

```

    [receiver{n,1},grad,lowPEIntercept,highPEIntercept,cutoffrange]=
    JRPEModelQuickerRev(verticalPlot2D,directionalSource,terrain,atmosphere,impedances,freq,czero,range,height,
    ..
    z0,zflightPositions,deltarPlot,deg,Z,lengthfr,ctype,ssGradient,receiverHeight,n,omega,kzero,deltar,deltaz,heightwa
    bs,zN,z,betaAtmAbs,turbulence,realizN,A,K0,CTovT0,Cvovc0,levelat1m,H);
    strArraySSP(n)= cellstr(grad);
    end

```

Using the only the FFP model

```

    elseif strcmp(propagationModel,'FFP')==1

```

Assign necessary parameters for each azimuthal angle

```

    for n= 1:runN-1 %only to runN-1 because the first (angle= 0) and last (angle= 2*pi) azimuthal angle runs will be
    the same
        n
        range= rangesAtThetas(n)-deltar; %Range for model to progagate out to. A deltar is subtracted
        %because polarTerrain includes one less range value terrain point (due to limits of interpolation). Even though the FFP
        %does not use terrain data, this is done for consistency.
        %(Range-dependent effects not included)
        ssGradient= ssGradientAtThetas(n); % effective sound speed parameter at angle theta(n)
        strArraySSP(n)= cellstr(' ');
    end

```

Call the FFP model

```

    [receiver{n,1}]=
    JRFFPModelQuickerRev(verticalPlot2D,atmosphere,freq,czero,range,height,z0,zflightPositions,ZAtThetas(1),ctyp
    e,ssGradient,...
    receiverHeight,n,omega,kzero,deltarPlot,deltar,deltaz,zN_FFP,z_FFP,betaAtmAbs,levelat1m);
    end
    else error("The "Propagation_Model" category input you have entered is invalid. Please enter either "hybrid",
    "PE", or "FFP")
    end

```

Plot the results

```

    receiver{runN,1}= receiver{1,1}; %first (angle= 0) and last (angle= 2*pi) azimuthal angle runs are the same
    labels= [1 sqrt((1200 400 630 1000 2000 4000 6300 10000 16000 25000 50000 75000)/3.2808).^2-
    zflightPositions(1)^2)];

```

```
labels= round(labels)+1;
figure(102);
hPlot=semilogx([0:deltarPlot:rangesAtThetas(1)-deltar]*3.2808,receiver{1,:},'LineWidth',2);
xlabel('range [ft]','FontSize',32)
ylabel('Level [dB]','FontSize',32)
title(['Test Case ',num2str(caseNumber), ', ', num2str(freq), ' Hz, source ',
num2str(round(zflightPositions(1)*3.2808)), ' ft height'],'FontSize',32)
set(gca,'FontSize',32);
```

JRPEModelQuickerRev Function

This function is called to run the Parabolic Equation Model for the current frequency, current source point in the flight path, and current azimuthal angle run.

Inputs:

- *verticalPlot2D* - option for producing vertical 2D plot ('yes' or 'no')
- *directionalSource* - option for using a source directivity pattern ('yes') or assuming an omnidirectional source ('no')
- *terrain* - characteristic of the terrain ('flat' or 'uneven')
- *atmosphere* - characteristic of the atmosphere ('homogeneous' or 'refractive')
- *impedances* - number of impedances encountered at the current azimuthal run for the current source point along the flight path
- *freq* - frequency used in this run of the model
- *czero* - sound speed at 0 m height
- *range* - range for model to propagate to
- *height* - grid height (not including PE absorbing layer)
- *z0* - aerodynamic roughness (ignored if ssp is linear)
- *sourceHeight* - z-coordinate of the current source point along the flight path.
- *deltarPlot* - range step for desired resolution of results
- *deg* - elevation angle limit used for PE
- *Z* - a vector with the values of the different impedances for the current azimuthal run and source point
- *lengthefr* - vector of the ranges at which the impedance transitions occur, for the current azimuthal run and source point
- *ctype* - type of sound speed profile ('linear' or 'ln')
- *ssGradient* - sound speed parameter
- *receiverHeight* - receiver height at which results are returned
- *omega* - angular frequency used in this run of the model
- *kzero* - reference wave number used in this run of the model
- *deltar* - horizontal grid spacing used in the PE propagation model, dependent on frequency. Also constrained to be a divisor of number chosen in 'MultiFreqQuickerRev.m'
- *deltaz* - vertical grid spacing used in the PE and FFP propagation models. Constrained to be a divisor of the receiverHeight so that the receiverHeight will fall exactly at a vertical grid point
- *heightwabs* - grid height for PE including PE absorbing layer
- *zN* - number of height points in grid for PE (includes absorbing layer)
- *z* - height vector for PE
- *betaAtmAbs* - atmospheric absorption coefficient for current frequency
- *turbulence* - option to include turbulence ('yes' or 'no')
- *realizN* - number of realizations to include
- *A*, *K0*, *CTovT0*, *Cvovc0* - turbulence parameters A, K0, CTovT0, and Cvovc0
- *levelat1m* - reference source level at 1 m distance from the current point source

- H - a vector of terrain heights at all ranges for the current azimuthal run and source point

Function Process and Outputs:

1. Find range after which all PE calculations are valid to pass to the FFP model by calculating *lowPEIntercept* and *highPEIntercept*
2. Define the range parameters for the current azimuthal angle run
3. Define the terrain parameters
4. Define the atmosphere, returning a string that describes the sound speed profile, used for figure titles, *grad*
5. Define the indices of results that will be saved (separated by *deltarPlot*)
6. Start Generalized Terrain Model Code, saving results at only the receiverHeight if the vertical plot option is not invoked, and saving results at all heights if the option is invoked
7. Call simpler PE model to calculate results for the first 3 range steps
8. Loop through realizations if turbulence is invoked
9. Initialize the values of the 3 previous range steps
10. Initialize result variables
11. Save results at any range steps that are at steps with the desired resolution
12. If turbulence is invoked, calculate the turbulent fluctuation of the acoustic refractive index
13. Run the generalized terrain model for the rest of the grid
14. Save results at any range steps that are at steps with the desired resolution
15. Determine where to stop propagation, *cutoffrange*
16. Calculate the sound level in dB, *receiver*
17. If the vertical sound field option is invoked, calculate the sound field grid, *gridsq*, and plot the results
18. If turbulence is invoked, save the averaged results to the function outputs

Start Code Description:	3
Find range after which all PE calculations are valid	3
Define the range parameters for the current azimuthal angle run	3
Define the derivatives of terrain	3
Define the atmosphere	3
Define the indices of results that will be saved	4
Start Generalized Terrain Model Code	4
Call simpler PE model to calculate results for the first 3 range steps	5
If use of turbulence is not invoked, only one realization will be calculated	5
If turbulence is invoked, initialize a vector to store the results [dB], averaged over the realizations, at the receiver height. If the vertical 2D plot option is invoked, initialize a matrix to store all vertical grid results	5
Loop through the realizations	6
Initialize the values of the 3 previous range steps	6
Initialize result variables	6
Save results at any range steps that are at steps with the desired resolution	6

If turbulence is invoked, calculate the turbulent fluctuation of the acoustic refractive index, mu	6
Run the generalized terrain model for the rest of the grid	6
Save results at any range steps that are at steps with the desired resolution.....	8
Determine where to stop propagation	8
Calculate the sound level in dB	9
Define a string representing the frequency that can be stored in a file name.....	9
If the turbulence option is invoked, save the results for all the realizations separately, and calculate the level averaged over realizations.....	9
Plot the results.....	9
If turbulence is invoked store the averaged results in the function outputs and, if the vertical 2D plot option is invoked plot them.....	10

Start Code Description:

```
function [receiver,grad,lowPEIntercept,highPEIntercept,cutoffrange,gridsq]=
JRPEModelQuickerRev(verticalPlot2D,directionalSource,terrain,atmosphere,impedances,freq,czero,range,height,
..
z0,sourceHeight,deltaPlot,deg,Z,lengthfr,ctype,ssGradient,receiverHeight,runNumber,...
omega,kzero,deltar,deltaz,heightwabs,zN,z,betaAtmAbs,turbulence,realizN,A,K0,CTovT0,Cvovc0,levelat1m,H)
```

Find range after which all PE calculations are valid

This is used as an input to the FFP model, so propagation can be stopped after it is no longer needed

```
lowPEIntercept= sourceHeight/tan(deg*pi/180);           %This is the range value at which valid PE triangle
intersects the ground
highPEIntercept= min(range,(height-sourceHeight)/tan(deg*pi/180)); %This is the range value at which valid PE
triangle intersects the top of the grid
```

Define the range parameters for the current azimuthal angle run

```
r= 0:deltar:range; %the range is defined here because it is specific to each azimuthal angle run
rN= length(r);
```

Define the derivatives of terrain

```
dH= |0 diff(H)|/deltar;
d2H= |0 diff(dH)|/deltar;
```

Define the atmosphere

Calculate the sound speed profile and apply the absorbing layer at the top of the grid, and atmospheric absorption.

```
absorbz= height+deltaz;           %height of the start of absorbing layer start in meters
absorbN= absorbz/deltaz+1;        %height of the start of absorbing layer in number of points
absorb= AbsorbTermRev(freq,deltaz,absorbN,zN,absorbz,heightwabs); %absorptive term to be added to
wavenumber

if strcmp(atmosphere,'homogeneous')
```

```

% cz is the sound speed profile from 0 to the top of the absorbing
% layer
% grad is a string for plotting labels
[cz,grad]= AtmosphereRev(czero,zN);
elseif strcmp(ctype,'linear')
    [cz,grad]= AtmosphereRev(czero,zN,ssGradient,deltaz);
elseif strcmp(ctype,'ln')
    [cz,grad]= AtmosphereRev(czero,zN,ssGradient,deltaz,z0);
else [cz,grad]= AtmosphereRev(czero,zN,a,deltaz,z0,a2);
end

% Add the atmospheric attenuation and absorbing layer attenuation
keta= omega./cz.+j*betaAtmAbs;
keta(absorbN:zN)= keta(absorbN:zN)+absorb;

```

Define the indices of results that will be saved

In order to conserve memory, not all calculated results at all ranges will be saved to the results variables

```

rindex= round(1:deltarPlot/deltar:length(r)); %indices of each range value where results will be saved
groundN= round(H(rindex)/deltaz); %height of the ground at each plotted range value in number of points
(defined for plotting purposes only)

```

Start Generalized Terrain Model Code

Generalized Terrain code follows closely to the description of the Generalized Terrain PE method in Section M.3 in E. M. Salomons, "Computational Atmospheric Acoustics," 2001.

```

ktop= keta(zN);
k0= keta(1);

deltaeta= deltaz;
deltaxi= deltar;

% Form delta^2 matrix
D=sparse(1:zN-1,1:zN-1,-2*ones(1,zN-1),zN-1,zN-1);
E=sparse(2:zN-1,1:zN-2,1*ones(1,zN-2),zN-1,zN-1);
T3=D+E+E';
S3= T3;

% Form delta matrix
E=sparse(2:zN-1,1:zN-2,1*ones(1,zN-2),zN-1,zN-1);
T2=.5*(-E+E');
clear('D','E');
S2= T2;

alphaH= atan(dH);
%alphaH= zeros(1,length(r));

epsilon= 2*deltaeta*cos(alphaH);

%k is dependent on eta, not xi. Therefore, k0 will be the same at every
%ground point as it depends only on how high the receiver is off the ground
%(not how high the ground is).

if impedances == 1
    d= ((-j*k0)/Z(1))+(3./epsilon)-((1.5/deltaxi)+j*kzero)*sin(alphaH);
elseif impedances == 2
    d= ((-j*k0)/Z(1))+(3./epsilon)-((1.5/deltaxi)+j*kzero)*sin(alphaH);

```

```

d(round(lengthfr/deltar)+2:length(d))= ((-j*k0)/Z(2))+(3./epsilon(round(lengthfr/deltar)+2:length(d)))-
((1.5/deltaxi)+j*kzero)*sin(alphaH(round(lengthfr/deltar)+2:length(d)));
elseif impedances == 3
d= ((-j*k0)/Z(1))+(3./epsilon)-((1.5/deltaxi)+j*kzero)*sin(alphaH);
d(round(lengthfr(1)/deltar)+2:round(lengthfr(2)/deltar)+1)= ((-
j*k0)/Z(2))+(3./epsilon(round(lengthfr(1)/deltar)+2:round(lengthfr(2)/deltar)+1))-
((1.5/deltaxi)+j*kzero)*sin(alphaH(round(lengthfr(1)/deltar)+2:round(lengthfr(2)/deltar)+1));
d(round(lengthfr(2)/deltar)+2:length(d))= ((-j*k0)/Z(3))+(3./epsilon(round(lengthfr(2)/deltar)+2:length(d)))-
((1.5/deltaxi)+j*kzero)*sin(alphaH(round(lengthfr(2)/deltar)+2:length(d)));
else error('Please assign either 1, 2, or 3 to the variable impedances')
end

u= 4./(d.*epsilon);
v= -u/4;
w= 2*sin(alphaH)/(d*deltaxi);
y= -w/4;

epsilon= 2*deltaeta*cos(alphaH);

dt= (-j*ktop)+(3./epsilon)+((1.5/deltaxi)+j*kzero)*sin(alphaH);

ut= 4/(dt.*epsilon);
vt= -ut/4;
wt= -2*sin(alphaH)/(dt*deltaxi);
yt= -wt/4;

```

Call simpler PE model to calculate results for the first 3 range steps

The generalized terrain model extrapolates from the previous 3 range steps. Therefore a simpler PE model, assuming flat ground, that extrapolates from only the previous range step is used until enough information is gathered to begin implementation of the GTPE.

```

[gridstart,psivectorstart]= JRPEFstartCompareRev(directionalSource,freq,keta,kzero,deltar,deltaz,zN,...
sourceHeight,Z(1));

```

If use of turbulence is not invoked, only one realization will be calculated

```

if strcmp(turbulence,'no') % if assuming no turbulence
realizN= 1; % run the PE only once for each condition
end

```

If turbulence is invoked, initialize a vector to store the results [dB], averaged over the realizations, at the receiver height. If the vertical 2D plot option is invoked, initialize a matrix to store all vertical grid results.

```

if strcmp(turbulence,'yes')
averageReceiver= -100000000*ones(1,length(rindex));
% If vertical plot is invoked, initialize a matrix to store the results [dB],
% averaged over the realizations, for the vertical grid.
if verticalPlot2D==1
averageVerticalGrid= -100000000*ones(max(groundN)+zN,length(rindex));
end
end
end

```

Loop through the realizations

```
for thisrealiz= 1:realizN
```

Initialize the values of the 3 previous range steps

Used to extrapolate the next range step. Variables are re-initialized for each realization

```
psiMinus3= psivectorstart(:,1);
psiMinus2= psivectorstart(:,2);
psiMinus1= psivectorstart(:,3);
```

Initialize result variables

Variables are re-initialized for each realization

```
receiverPressure= zeros(1,length(rindex)); %slice of the sound level at receiverHeight meters above the ground
level
if verticalPlot2D==1
    grid= zeros(zN,length(rindex)); %matrix of vertical sound field results
end
```

Save results at any range steps that are at steps with the desired resolution

To conserve memory, not all calculated results at all ranges will be saved to the results variables

```
for thisr= 1:3
    if any(rindex==thisr)
        receiverPressure(r(thisr)/deltarPlot+1)= gridstart(round(receiverHeight/deltaz)+1,thisr);
        if verticalPlot2D==1
            grid(:,round(r(thisr)/deltarPlot+1))= gridstart(:,thisr);
        end
    end
end

cutoffrange= range; %user-defined range is the maximum ranged to be propagated to. Model can be stopped
before if a cut-off criteria is reached
```

If turbulence is invoked, calculate the turbulent fluctuation of the acoustic refractive index, mu

Turbulence is applied starting at the fourth range step.

```
if strcmp(turbulence,'yes')
    [mu]= vonKarTurb(deltar,range,rN,deltaz,zN,A,K0,CTovT0,Cvovc0);
end
```

Run the generalized terrain model for the rest of the grid

```
for thisr=4:rN
    if strcmp(turbulence,'yes') % if turbulence is invoked
```

```

    if r(thisr)>cutoffrange % if the cutoff range determined in the first realization has been exceeded
        break % stop propagating
    end
    psiMinus1= psiMinus1.*exp(0.5*j*kzero*mu(:,thisr-1)*deltar); % apply turbulence with the turbulent phase
    factor, as described in E. M. Salomons "Computational Atmospheric Acoustics," (2001) in Eq. J.11 and text below
    end

% The quantities u, v, w, y, ut, vt, wt, and yt in these expressions
% are evaluated at xi= b.

T3(1,1)= -2+u(thisr);
T3(1,2)= 1+v(thisr);
T3(zN-1,zN-1)= -2+ut(thisr);
T3(zN-1,zN-2)= 1+vt(thisr);

k31= w(thisr)*psiMinus1(1)+y(thisr)*psiMinus2(1);
k3top= wt(thisr)*psiMinus1(zN)+yt(thisr)*psiMinus2(zN);
k3= sparse([1,zN-1],[1,1],[k31 k3top],zN-1,1);

T2(1,1)= -u(thisr);
T2(1,2)= 1-v(thisr);
T2(zN-1,zN-1)= ut(thisr);
T2(zN-1,zN-2)= -1+vt(thisr);

k21= -w(thisr)*psiMinus1(1)-y(thisr)*psiMinus2(1);
k2top= wt(thisr)*psiMinus1(zN)+yt(thisr)*psiMinus2(zN);
k2= .5*sparse([1,zN-1],[1,1],[k21 k2top],zN-1,1);

S3(1,1)= -2+u(thisr-1);
S3(1,2)= 1+v(thisr-1);
S3(zN-1,zN-1)= -2+ut(thisr-1);
S3(zN-1,zN-2)= 1+vt(thisr-1);

m31= w(thisr-1)*psiMinus2(1)+y(thisr-1)*psiMinus3(1);
m3top= wt(thisr-1)*psiMinus2(zN)+yt(thisr-1)*psiMinus3(zN);
m3= sparse([1,zN-1],[1,1],[m31 m3top],zN-1,1);

S2(1,1)= -u(thisr-1);
S2(1,2)= 1-v(thisr-1);
S2(zN-1,zN-1)= ut(thisr-1);
S2(zN-1,zN-2)= -1+vt(thisr-1);

m21= -w(thisr-1)*psiMinus2(1)-y(thisr-1)*psiMinus3(1);
m2top= wt(thisr-1)*psiMinus2(zN)+yt(thisr-1)*psiMinus3(zN);
m2= .5*sparse([1,zN-1],[1,1],[m21 m2top],zN-1,1);

% where alpha, beta, and dH are evaluated at xi= b.
alphaB= (dH(thisr))^2+1;
betaB= 2*j*kzero*dH(thisr)+d2H(thisr);
gammaB= keta(2:length(keta)).^2-kzero^2;
chiB= d2H(thisr)-2*j*kzero*dH(thisr);

% where alpha, beta, and dH are evaluated at xi= a.
alphaA= (dH(thisr-1))^2+1;
betaA= 2*j*kzero*dH(thisr-1)+d2H(thisr-1);
gammaA= keta(2:length(keta)).^2-kzero^2;
chiA= d2H(thisr-1)-2*j*kzero*dH(thisr-1);

Balpha= deltaxi*((1/6)*alphaA+(1/3)*alphaB);
Aalpha= deltaxi*((1/3)*alphaA+(1/6)*alphaB);

Bchi= deltaxi*((1/6)*chiA+(1/3)*chiB);
Achi= deltaxi*((1/3)*chiA+(1/6)*chiB);

Bgamma= deltaxi*((1/6)*gammaA+(1/3)*gammaB);
Agamma= deltaxi*((1/3)*gammaA+(1/6)*gammaB);

c3= (1/(2*j*kzero*deltaeta^2))*(((j*alphaB)/(2*kzero))+Balpha);
c2= -(1/(2*j*kzero*deltaeta))*(((j*betaB)/(2*kzero))+2*dH(thisr)-Bchi);
c1= sparse(1:zN-1,1:zN-1,ones(1,zN-1),zN-1,zN-1)+sparse(1:zN-1,1:zN-1,(gammaB/(4*kzero^2))+(Bgamma/(2*j*kzero)),zN-1,zN-1);

```

```

d3= (1/(2*j*kzero*deltaeta^2))*((j*alphaA)/(2*kzero))-Aalpha);
d2= -(1/(2*j*kzero*deltaeta))*((j*betaA)/(2*kzero))+2*dH(thisr-1)+Achi);
d1= sparse(1:zN-1,1:zN-1,ones(1,zN-1),zN-1,zN-1)+sparse(1:zN-1,1:zN-1,(gammaA/(4*kzero^2))-
(Agamma/(2*j*kzero)),zN-1,zN-1);

rhs=(d3*S3+d2*S2+d1)*psiMinus1(2:zN)+d3*m3+d2*m2-c3*k3-c2*k2;

% Solve the linear system of equations
psivector(2:zN,1)=(c3*T3+c2*T2+c1) \ rhs;
psivector(1,1)= u(thisr)*psivector(2)+v(thisr)*psivector(3)+w(thisr)*psiMinus1(1)+y(thisr)*psiMinus2(1);

if strcmp(turbulence,'yes') % if turbulence is invoked
    psivector= psivector.*exp(0.5*j*kzero*mu(:,thisr-1)*deltar); % apply turbulence with the turbulent phase
    factor, as described in E. M. Salomons "Computational Atmospheric Acoustics," (2001) in Eq. J.11 and text below
end

psiMinus3= psiMinus2;
psiMinus2= psiMinus1;
psiMinus1= psivector;

```

Save results at any range steps that are at steps with the desired resolution

```

if any(rindex==thisr)%rem(r(thisr),deltarPlot)<10^-5

    receiverPressure(round(r(thisr)/deltarPlot)+1)=
psivector(round(receiverHeight/deltaz)+1)/sqrt(deltar*(thisr-1));

    if verticalPlot2D==1
        grid(:,round(r(thisr)/deltarPlot+1))= psivector/sqrt(deltar*(thisr-1));
    end

```

Determine where to stop propagation

After cutoff criteria limit in sound level is reached, stop propagation

```

if r(thisr)>=1000 && thisrealiz==1 % if the current range is larger than 1000 m and this is the first
realization (so that all realizations have the same cutoffrange)
    % Find the maximum sound level value in the previous 100m range (to
    % make sure that propagation doesn't stop because the current range is
    % in an interference dip)
    Lsignalband= levelat1m+20*log10(max(abs(receiverPressure((r(thisr)/deltarPlot+1)-
(100/deltarPlot):(r(thisr)/deltarPlot+1))))/abs(exp(j*kzero)));
    % Call the DetectabilityFilter function to decide whether further
    % propagation is necessary or if all significant levels have been
    % calculated.
    continueLoop= DetectabilityFilter(freq,Lsignalband);
    % If continueLoop==1, all significant levels have been calculated, and
    % propagation can stop, so the loop can end.
    cutoffrange= range;
    if continueLoop==0
        cutoffrange= r(thisr);
        ['The cut-off criteria has been reached. Propagation has stopped at ', num2str(r(thisr)), ' m. ']
        break
    end
end
end
end
end
end
end
end

```

Calculate the sound level in dB

Calculated results for all ranges at the specified resolution (deltarPlot) at the receiver height, based on the provided level 1 m from the source.

```
receiver= levelat1m+20*log10(abs(receiverPressure)/abs(exp(j*kzero)));
```

Define a string representing the frequency that can be stored in a file name

```
freqString= num2str(freq);
if rem(freq,1)~=0
    indexfreqString= strfind(freqString,'. ');
    freqString= [freqString(1:indexfreqString-1) '_' freqString(indexfreqString+1)];
end
```

If the turbulence option is invoked, save the results for all the realizations separately, and calculate the level averaged over realizations

```
if strcmp(turbulence,'yes') % if turbulence is invoked
    eval(['receiver_' freqString 'Hz_realiz' num2str(thisrealiz) '= receiver;']);
    save(['receiverdata' freqString 'Hz_realiz' num2str(thisrealiz) '.mat'], ['receiver_' freqString 'Hz_realiz'
num2str(thisrealiz)]); % save results at the receiver height separately for each realization
    averageReceiver= 10*log10(((thisrealiz-1)*10.^(averageReceiver/10)+10.^(receiver/10))/thisrealiz); %
calculate the average results at the receiver height for all realizations run so far
end
```

Plot the results

```
if verticalPlot2D==1
    gridsq= zeros(max(groundN)+zN,length(rindex));
    for thisr= 1:length(rindex)
        gridsq(:,thisr)= levelat1m+20*log10(abs([NaN(groundN(thisr),1); grid(:,thisr); NaN(max(groundN)-
groundN(thisr),1)])/abs(exp(j*kzero)));
    end

    grid= levelat1m+20*log10(abs(grid)/abs(exp(j*kzero)));

    figure(runNumber);
    imagesc(r(rindex),z,grid);
    colormap(jet)
    axis xy
    ylim([0,heightwabs])
    axis square
    xlabel('range [m]', 'fontsize',22)
    ylabel('height [m]', 'fontsize',22)
    title(['PE model, source (0,',int2str(sourceHeight),') f=',int2str(freq),' Hz, 'grad'],'fontsize',22);

    figure(runNumber+1);
    imagesc(r(rindex),z,gridsq);
    colormap(jet)
    axis xy
    ylim([0,heightwabs])
    axis square
    xlabel('range [m]', 'fontsize',22)
    ylabel('height [m]', 'fontsize',22)
    title(['PE model, source (0,',int2str(sourceHeight),') f=',int2str(freq),' Hz, 'grad'],'fontsize',22);

    % Save results at the receiver height (separately for each
    % realization)
```

```

eval(['gridsq' freqString 'Hz_realiz' num2str(thisrealiz) '= gridsq;']);
save(['allverticaldata' freqString 'Hz_realiz' num2str(thisrealiz) '.mat'], ['gridsq' freqString 'Hz_realiz'
num2str(thisrealiz)]);
% Calculate the average results in the vertical grid for all
% realizations run so far
if strcmp(turbulence,'yes')
    averageVerticalGrid= 10*log10(((thisrealiz-1)*10.^(averageVerticalGrid/10)+10.^(gridsq/10))/thisrealiz);
end
end
end

```

If turbulence is invoked store the averaged results in the function outputs and, if the vertical 2D plot option is invoked plot them

```

if strcmp(turbulence,'yes') % if turbulence is invoked

% Define the outputs as the average results over all the realizations
receiver= averageReceiver;
if verticalPlot2D==1
    gridsq= averageVerticalGrid;

    figure(3);
    imagesc(r(rindex),z,gridsq);
    colormap(jet)
    axis xy
    ylim([0,heightwabs])
    axis square
    xlabel('range [m]',fontSize,22)
    ylabel('height [m]',fontSize,22)
    title(['PE model average over ' int2str(realizN) ' realizations, source (0,'int2str(sourceHeight),')
f=',int2str(freq),' Hz, ',grad]',fontSize,22);
end
end

```

JRFFPModelQuickerRev Function

This function is called to run the Fast-Field Program Model for the current frequency, current source point in the flight path, and current azimuthal angle run.

Inputs:

- *atmosphere* - characteristic of the atmosphere ('homogeneous' or 'refractive')
- *freq* - frequency used in this run of the model
- *czero* - sound speed at 0 m height
- *range* - range for FFP model to propagate to (does not need to propagate past the range for which all PE calculations are valid)
- *height* - grid height (not including PE absorbing layer)
- *z0* - aerodynamic roughness (ignored if ssp is linear)
- *sourceHeight* - z-coordinate of the current source point along the flight path.
- *Z* - a vector with the values of the impedances used for the FFP model at the current azimuthal run and source point (the first impedance in the vector is used)
- *ctype* - type of sound speed profile ('linear' or 'ln')
- *ssGradient* - sound speed parameter
- *receiverHeight* - receiver height at which results are returned
- *runNumber* - number of the current azimuthal angle run
- *omega* - angular frequency used in this run of the model
- *kzero* - reference wave number used in this run of the model
- *deltarPlot* - range step for desired resolution of results
- *deltar* - horizontal grid spacing used in the PE propagation model, dependent on frequency. Also constrained to be a divisor of number chosen in 'MultiFreqQuickerRev.m'
- *deltaz* - vertical grid spacing used in the PE and FFP propagation models. Constrained to be a divisor of the receiverHeight so that the receiverHeight will fall exactly at a vertical grid point
- *zN* - number of height points in grid for FFP (excludes absorbing layer)
- *z* - height vector for FFP (excludes absorbing layer)
- *betaAtmAbs* - atmospheric absorption coefficient for current frequency
- *levelat1m* - reference source level at 1 m distance from the current point source

Function Process and Outputs:

1. Redefine range and range step parameters to be appropriate for the FFP
2. Define the atmosphere
3. Run the FFP model saving results at only the receiverHeight if the vertical plot option is not invoked, and saving results at all heights if the option is invoked
4. Transform back to the spatial domain to calculate the pressure
5. Calculate the sound level in dB
6. If the vertical sound field option is invoked, calculate the sound field grid *receiverFFPgrid* and plot the results

Start Code Description:	2
Redefine range and range step parameters to be appropriate for the FFP	2
Define the atmosphere	2
Run the FFP model	2
Transform back to the spatial domain to calculate the pressure	4
Calculate the sound level in dB	4
If the vertical sound field option is invoked, calculate the sound field grid and plot the results	4

Start Code Description:

```
function [receiverFFP,receiverFFPgrid]=
JRFFPModelQuickerRev(verticalPlot2D,atmosphere,freq,czero,range,height,z0,sourceHeight,...
Z,cType,ssGradient,receiverHeight,runNumber,omega,kzero,...
deltarPlot,deltar,deltaz,zN,z,betaAtmAbs,levelat1m)
```

Redefine range and range step parameters to be appropriate for the FFP

```
% The FFP does not require range steps as small as does the PE
if deltarPlot>=2*deltar
    deltarFFP= 2*deltar;
else deltarFFP= deltar;
end

% The FFP requires much larger ranges to be calculated than will be used
range= 3*range;

% Define the range vector
r= 0:deltarFFP:range;
rN= length(r);
```

Define the atmosphere

Calculate the sound speed profile and apply atmospheric absorption.

```
if strcmp(atmosphere,'homogeneous')
    % cz is the sound speed profile from 0 to the top of the absorbing
    % layer
    % grad is a string for plotting labels
    [cz,grad]= AtmosphereRev(czero,zN);
elseif strcmp(cType,'linear')
    [cz,grad]= AtmosphereRev(czero,zN,ssGradient,deltaz);
elseif strcmp(cType,'ln')
    [cz,grad]= AtmosphereRev(czero,zN,ssGradient,deltaz,z0);
else [cz,grad]= AtmosphereRev(czero,zN,a,deltaz,z0,a2);
end

k= omega./cz.'+j*betaAtmAbs;
```

Run the FFP model

Fast-Field Program code follows closely to the description of the FFP method in Appendix F in E. M. Salomons, "Computational Atmospheric Acoustics," 2001.

```
deltak= 2*pi/(deltarFFP*rN);
```

```

ks= .5*deltak:deltak:ceil(3*omega/(czero*deltak))*deltak+.5*deltak;
kr= ks-j*deltak; %Salomons p.159 says to use deltak for kt
krN= length(kr);

KZ1= (k(1)^2-kr.^2).^(.5);

Sdelta= sqrt(2*pi*kr); %Sdelta is 1 x krN
Rkz1= (KZ1-k(1)/Z)./(KZ1+k(1)/Z); %Rkz1 is 1 x krN

m= round(sourceHeight/deltaz+1);
kz0= (k(m)^2-kr.^2).^(.5);

%%
if verticalPlot2D==1
    Pall= zeros(zN,krN);
    Pall(1,:)= Rkz1+1;
end

P= Rkz1+1;
dP= j*KZ1.*(Rkz1-1);

for n= 1:m-1
    %n signifies the level number
    KZn= (k(n)^2-kr.^2).^(.5);
    Plzplus1= cos(KZn*deltaz).*P+(KZn).^(-1).*sin(KZn*deltaz).*dP;
    dPlzplus1= -KZn.*sin(KZn*deltaz).*P+cos(KZn*deltaz).*dP;
    P= Plzplus1.*exp(j*kz0*deltaz);
    if verticalPlot2D==1
        Pall(n+1,:)= P;
    end
    dP= dPlzplus1.*exp(j*kz0*deltaz);
    if abs(z(n+1)-receiverHeight)<.00001 %essentially if they are equal, allowing for roundoff errors
        Preceiver= P;
    end
end

Pml= Plzplus1; %Pml is 1 x krN
dPml= dPlzplus1; %dPml is 1 x krN

P= ones(1,krN);
if verticalPlot2D==1
    Pall(zN,:)= P;
end
dP= j*(k(zN)^2-kr.^2).^(.5);

Puzplus1= P.*exp(j*kz0*deltaz);
dPuzplus1= dP.*exp(j*kz0*deltaz);

for n= zN-1:-1:m
    %n signifies the level number
    KZn= (k(n)^2-kr.^2).^(.5);
    P= cos(KZn*deltaz).*Puzplus1+(KZn).^(-1).*sin(KZn*deltaz).*dPuzplus1;
    if verticalPlot2D==1
        Pall(n,:)= P;
    end
    dP= -KZn.*sin(KZn*deltaz).*Puzplus1+cos(KZn*deltaz).*dPuzplus1;
    if n>0
        Puzplus1= P.*exp(j*kz0*deltaz);
        dPuzplus1= dP.*exp(j*kz0*deltaz);
    end
    if abs(z(n)-receiverHeight)<.00001 %essentially if they are equal, allowing for roundoff errors
        Preceiver= P;
    end
end

Pmu= P; %PmU is 1 x krN
dPmu= dP; %dPmU is 1 x krN

Pm= -Sdelta./((dPmu/Pmu)-(dPml/Pml));

if verticalPlot2D==1

```

```

    Pall(1:m-1,:)= Pall(1:m-1,:).*(((exp(j*abs(z(1:m-1)-sourceHeight).*kz0)).*(ones(m-1,1)*Pm))./(ones(m-
1,1)*Pml)); %Pm is 1 x krN
    Pall(m+1:zN,:)= Pall(m+1:zN,:).*(((exp(j*abs(z(m+1:zN)-sourceHeight).*kz0)).*(ones(zN-
m,1)*Pm))./(ones(zN-m,1)*Pmu));
    Pall(m,:)= Pm;
end

receiverHeightN= receiverHeight/deltaz+1;
if receiverHeightN<m
    Preceiver= Preceiver.*(exp(j*abs(receiverHeight-sourceHeight).*kz0)).*Pm./Pml;
elseif receiverHeightN>m
    Preceiver= Preceiver.*(exp(j*abs(receiverHeight-sourceHeight).*kz0)).*Pm./Pmu;
elseif receiverHeight==m
    Preceiver= Pm;
end

clear dP Plzplus1 dPlzplus1 Puzplus1 dPuzplus1 Pm Pml dPml Pmu dPmu Rkz1 Sdelta ks;

% If you have the Matlab Signal Processing toolbox, you can uncomment the
% next line of code and comment the chunk of code following.
windowInt= tukeywin(length(kr),1/5).';

% % The following definition of the tukey window is after Matlab signal processing
% % toolbox tukeywin .m script:
% rWindow=(1/5);
% tWindow = linspace(0,1,krN)';
% % Defines period of the taper as 1/2 period of a sine wave.
% perWindow = rWindow/2;
% tlWindow = floor(perWindow*(krN-1))+1;
% thWindow = krN-tlWindow+1;
% % Window is defined in three sections: taper, constant, taper
% windowInt = [ ((1+cos(pi/perWindow*(tWindow(1:tlWindow) - perWindow)))/2); ones(thWindow-tlWindow-1,1);
((1+cos(pi/perWindow*(tWindow(thWindow:end) - 1 + perWindow)))/2)].';

```

Transform back to the spatial domain to calculate the pressure

```

pc= (((1-i)/(pi*sqrt(2)))*exp(deltak*r*(1-(i/2)))*(2*pi/deltarFFP)).*ifft(Preceiver.*windowInt,rN)+exp(-
deltak*r*(1-(i/2)))*deltak.*fft(Preceiver.*windowInt,rN))./sqrt(r);

if verticalPlot2D==1
    pcAll= zeros(zN,rN);
    for n= 1:zN
        pcAll(n,:)= (((1-i)/(pi*sqrt(2)))*exp(deltak*r*(1-(i/2)))*(2*pi/deltarFFP)).*ifft(Pall(n,:).*windowInt,rN)+exp(-
deltak*r*(1-(i/2)))*deltak.*fft(Pall(n,:).*windowInt,rN))./sqrt(r);
    end
end

```

Calculate the sound level in dB

Calculated results for all ranges at the specified resolution (deltarPlot) at the receiver height, based on the provided level 1 m from the source.

```

receiverFFP= levelat1m+20*log10(abs(pc(1:floor(range/(3*deltarFFP)+1)))/abs(exp(j*kzero)));
rindex= round(1:deltarPlot/deltarFFP:floor(range/(3*deltarFFP)+1));
receiverFFP= receiverFFP(rindex);

```

If the vertical sound field option is invoked, calculate the sound field grid and plot the results

```

if verticalPlot2D==1
    receiverFFPgrid= levelat1m+20*log10(abs(pcAll(:,rindex))/abs(exp(j*kzero)));

    figure(runNumber+2);

```

```
imagesc(r(rindex),z,receiverFFPgrid);
colorbar(jet)
colorbar;
axis xy
ylim([0,height])
axis square
xlabel('range [m]', 'fontsize', 22)
ylabel('height [m]', 'fontsize', 22)
title(['FFP model, source (0, 'int2str(sourceHeight),') f='int2str(freq), ' Hz, 'grad', 'fontsize', 22);
end
```

AbsorbTermRev Function

This function is called to return the imaginary contribution to the wave number in the absorbing layer so that the sound will be attenuated enough to prevent reflections off the top of the grid

Inputs:

- *freq* - frequency used in this run of the model
- *deltaz* - vertical grid spacing used in the PE and FFP propagation models.
- *absorbN* - height of the start of absorbing layer in number of points
- *zN* - number of height points in grid for PE (includes absorbing layer)
- *absorbz* - height of the start of absorbing layer start in meters
- *_heightwabs* - grid height for PE including PE absorbing layer

Function Process and Outputs:

1. Determine the parameter *At* that depends on frequency (calculated as recommended in Section 6.9 of E. M. Salomons "Computational Atmospheric Acoustics" (2001).
2. Calculate the imaginary term to be added to the wave number in order to prevent spurious reflections from the top of the grid *absorb*

```
*Start Code Description:* ..... 1
Determine parameter _At_ ..... 1
Calculate the imaginary term added to the wave number ..... 1
```

Start Code Description:

```
function [absorb]= AbsorbTermRev(freq,deltaz,absorbN,zN,absorbz,heightwabs)
```

Determine parameter _At_

```
if isequal(freq, 1000)
    At= 1;
elseif freq>500&&freq<1001
    At= (1/1000)*freq;
elseif freq>125
    At= (1/3750)*freq+(11/30);
elseif freq<=125
    At= (1/475)*freq+(13/95);
else 'Please enter a frequency from 30-1000 Hz'
end
```

Calculate the imaginary term added to the wave number

```
absorb= j*At*(deltaz*[absorbN-1:zN-1]-absorbz).^2/(heightwabs-absorbz)^2;
absorb= absorb.;
```

```

function [beta]= airatten_2006_Function(f,hr,t)

% Adapted from airatten_2006.m

% This mfile calculates the absorption coefficient for air using equations
% in Bass, et. al, JASA, vol 97, 680, 1995.

% f is frequency in Hz. The default range is 10 Hz - 10 kHz.
% hr is relative humidity in percent.
% t is absolute temperature in Kelvin. The default value for t is 293.15 K, i.e., 20 deg C.
% ps is the barometric pressure in atmospheres. The default value is 1 atm, i.e. 101325 Pa.
% The output alpha is the absorption coefficient in nepers/m.
% The output adbperkm is the absorption coefficient in dB/km.
% The graph is adbperkm vs f.

ps = 1;
pso = 1;
to = 293.15;
to1 = 273.16;
psat = pso*10^(-6.8346*(to1/t)^1.261 + 4.6151);
ff = f/ps;

h = hr*(psat/pso)/(ps/pso);

fro = (ps/pso)*(24 + 4.04e4*h*((h+0.02)/(h + 0.391)));
frn = (ps/pso)*sqrt(to/t)*(9 + 280*h*exp(-4.17*((to/t)^0.333 - 1)));

aoverps = ff.^2/pso.*(1.84e-11*sqrt(t/to) + (t/to)^-2.5*(0.01278*exp(-2239.1/t)/(fro + ff.^2/fro) + 0.1068*exp(-3352/t)/(frn + ff.^2/frn)));

alpha = aoverps*ps;
a = 8.686*alpha

beta= a/(20*log10(exp(1)));

```

AtmosphereRev Function

This variable number of arguments function is called to return the sound speed profile as calculated using the user option for a 'homogeneous' atmosphere, or a 'linear' or 'ln' (logarithmic) sound speed profile. (The additional option is a remnant of matching the Attenborough, et al. "Benchmark cases for outdoor sound propagation models" Case 4, ducting type of sound profile.)

Inputs:

- *czero* - sound speed at 0 m height
- *zN* - number of height points in grid for PE (includes absorbing layer)
- *param* - sound speed parameter of either the linear (gradient) or logarithmic profile
- *deltaz* - vertical grid spacing used in the PE and FFP propagation models. Constrained to be a divisor of the receiverHeight so that the receiverHeight will fall exactly at a vertical grid point
- *z0* - aerodynamic roughness (used only if the sound speed profile is logarithmic)
- *param2* - for the ducted sound speed profile, this is the second sound speed gradient provided

Function Process and Outputs:

- Depending on the type of user-specified atmosphere to be used, calculate the sound speed profile *cz*
- Form a string that describes the atmosphere, used for figure titles *grad* (For example, 'b= 1 m/s'. 'a' is used for linear profiles, 'b' is used for logarithmic profiles'.)

Start Code Description: 1

Start Code Description:

```
function [cz,grad]= AtmosphereRev(czero,zN,param,deltaz,z0,param2)

if (nargin == 2) %if using a homogeneous atmosphere
    cz= czero*ones(1,zN);
    grad= 'a= 0 m/s';

elseif (nargin == 4) %if using a linear sound speed profile
    a= param;
    cz= czero+a*(0:zN-1]*deltaz);
    grad= ['a= ',num2str(a),' s^{-1}'];

elseif (nargin == 5) %if using a logarithmic sound speed profile
    b= param;
    cz= czero+b*log((0:zN-1]*deltaz/z0)+1);
    grad= ['b= ',num2str(b),' m/s'];

elseif (nargin == 6) %if using a ducting type sound speed profile
    a1= param;
    a2= param2;
    zN100= floor(100/deltaz)+1;
```

```
zN300= floor(300/deltaz)+1;
cz(1:zN100)= czero+a1*(0:zN100-1)*deltaz;
cz(zN100+1:zN300)= cz(zN100)+a2*(1:zN300-zN100)*deltaz;
cz(zN300+1:zN)= cz(zN300)*ones(1,zN-zN300);
grad= ['a1= ',num2str(a1),', a2= ',num2str(a2),', a3= 0 s^{-1}'];
end
```

DandB Function

This function is called to return the normalized acoustic impedance value for a ground type described by an effective flow resistivity in cgs rayls

Inputs:

- *freq* - current frequency
- *efr* - vector of (up to 3) effective flow resistivities in cgs rayls of the ground type

Function Process and Outputs:

1. Convert the effective flow resistivity from cgs rayls to Pa s/m²
2. Compute the normalized acoustic impedance *Z* using Eq C.15 on p. 119 of E. M. Salomons "Computational Atmospheric Acoustics," 2001. (This equation is equivalent to the Delany-Bazley equation, but uses an effective flow resistivity with differen units.)

Start Code Description: 1
 Convert the effective flow resistivity from cgs rayls to Pa s/m²..... 1
 Compute the normalized acoustic impedance with the Delany-Bazley method 1

Start Code Description:

```
function [Z]= DandB(freq,efr)
```

Convert the effective flow resistivity from cgs rayls to Pa s/m²

```
efr= efr*1000; %because 1 cgs rayl = 1000 Pa s/m^2
```

Compute the normalized acoustic impedance with the Delany-Bazley method

```
Z= 1+.0511*(efr/freq).^-.75+i*.0768*(efr/freq).^-.73;
```

DetectabilityFilter Function

This function decides whether it is necessary to continue the propagation algorithms, or if the calculated level is sufficiently small that any further propagated sound can be neglected. It uses the parameters EASNthreshold, tenlogetaband (derived from the 1/3 octave band filter characteristic eta_band), and the bandwidths, as applied in determining detectability level in the INM 7.0 Technical Manual.

Inputs:

- *freq* - frequency used in this run of the model
- *Lsignalband* - maximum sound level value in the previous 100m range (to make sure that propagation doesn't stop because the current range is in an interference dip)

Function Process and Outputs:

1. Determine the frequency band index
2. Calculate the correction term applied to the difference between signal and noise levels
3. Decide whether to continue propagation (for *continueLoop*= 1, propagation continues, for *continueLoop*= 0, propagation stops)

```
*Start Code Description:* ..... 1
Determine the frequency band index and define other necessary parameters..... 1
Calculate the correction term applied to the difference between signal and noise levels2
Decide whether to continue propagation..... 2
```

Start Code Description:

```
function [continueLoop]= DetectabilityFilter(freq,Lsignalband)
```

Determine the frequency band index and define other necessary parameters

```
freqIndex= find([25 31.5 40 50 63 80 100 125 160 200 250 315 400 500 630 800 1000 1250 1600 2000 2500 3150 4000
5000 6300 8000 10000]==freq);

% EASN thresholds as outlined in INM 7.0 Technical Manual for 1/3 octave
% bands from 50-10,000 Hz, extended down to 25 Hz.
EASNthreshold= [40.2 40.2 40.2 40.2 35.0 29.8 25.8 22.2 19.0 16.2 13.4 11.6 9.3 7.8 6.3 6.3 6.1 5.4 5.2 4.0 2.8 2.4 4.0
8.1 13.1 17.0];

% 10*log10(eta_band) values as outlined in INM 7.0 Technical Manual for 1/3 octave
% bands from 50-10,000 Hz, extended down to 25 Hz.
tenlogetaband= [-7 -7 -7 -6.96 -6.26 -5.56 -5.06 -4.66 -4.36 -4.16 -3.96 -3.76 -3.56 -3.56 -3.56 -3.56 -3.56 -3.76 -3.96 -
4.16 -4.36 -4.56 -4.96 -5.36 -5.76 -6.26 -6.86];

% bandwidths as outlined in INM 7.0 Technical Manual for 1/3 octave
% bands from 50-10,000 Hz, extended down to 25 Hz.
bandwidth= [5.6 7.5 9.5 11 15 19 22 28 40 44 56 75 95 110 150 190 220 280 400 440 560 750 950 1100 1500 1900 2200];
```

Calculate the correction term applied to the difference between signal and noise levels

Calculation process outlined in the INM 7.0 Technical Manual. It is based on the $10 \cdot \log_{10}(\text{eta_band})$ values, and bandwidth

```
correctionFilter= (tenlogtaband(freqIndex)+0.5*10*log10(bandwidth(freqIndex)));
```

Decide whether to continue propagation

The max sound level is compared to a threshold value with a correction filter applied. Currently, the detectability criteria is set to 0 dB for each band. This number is relatively arbitrary, chosen in place of 7 dB, as used in INM, to lessen the possibility that the cutoff is applied before the band's contribution to the total detectability is negligible. Because the total detectability level cutoff criteria is also 7 dB, when adding contributions of all the bands, a level of less than 7 dB in one band may still be a significant contribution to the total level.

If `continueLoop == 1`, propagation continues, if `continueLoop == 0`, propagation stops

```
if Lsignalband-EASNthreshold(freqIndex)+correctionFilter>0
    continueLoop= 1;
else continueLoop= 0;
end
```

ImportImpedanceGrid Function

This function is called to return a matrix with impedance data for the space covered by the specified contouring grid

Function Process and Outputs:

This function is meant to be a placeholder. When the form of the input impedance data is known, this function can be rewritten to import the data and pass it through to the model. The output should be:

- *impedanceGrid* - a matrix with impedance defined for the physical location specified in the contouring grid definition.

The physical distance between the points defined in *impedanceGrid* is assumed to be the same as the distance between points defined in the terrainGrid

Start Code Description: 1

Start Code Description:

```
function [impedanceGrid]=
ImportImpedanceGrid(xmaxTerrain,xminTerrain,ymaxTerrain,yminTerrain,deltaTer)

%impedanceGrid= [150*ones(111,(xmaxTerrain-xminTerrain)/deltaTer+1); 150*ones((ymaxTerrain-
yminTerrain)/deltaTer+1-111,(xmaxTerrain-xminTerrain)/deltaTer+1)];
impedanceGrid= 150*ones((ymaxTerrain-yminTerrain),(xmaxTerrain-xminTerrain));
```

ImportTerrainGrid Function

This function is called to return both a matrix or vector with terrain height data for the space covered by the specified contouring grid and the spacing between the defined terrain data grid points

Function Process and Outputs:

This function is meant to be a placeholder. When the form of the input terrain data is known, this function can be rewritten to import the data and pass it through to the model. The user should be aware of the grid extension (currently extended to a square grid with each side equal to twice the longest range side of the original grid) to allow for interpolation of terrain and account for the extended grid when importing terrain. The output should be:

- *terrainGrid* - a matrix (for grid geometry 'horizontal') or vector (for grid geometry 'vertical') with terrain heights defined for the physical location specified in the contouring grid definition.
- *deltarTer* - the physical distance between the points defined in *terrainGrid*. The distance between points in the x and y directions should be equal.

Start Code Description: 1
 Vectical gridGeometry is invoked 1
 Horizontal (contouring) geometry is invoked 2

Start Code Description:

```
function [terrainGrid, deltarTer]=
  ImportTerrainGrid(xmaxTerrain,variableParameter,ymaxTerrain,yminTerrain)
```

Vectical gridGeometry is invoked

For a vertical (non-contouring) geometry, a single vector, *terrainGrid*, defining the terrain heights along the vertical plane of propagation, from a range of 0 to the defined end range, *xmaxTerrain*, with a range step equal to that of the PE grid, is returned

```
if (nargin == 2)
    deltar= variableParameter;

    terrainGrid= zeros(1,xmaxTerrain/deltar+1);

    ater= 10;
    Wter= 100;
    xtopter= 230;
    ster= 2*xtopter/Wter;
    startNter= round((xtopter-(Wter/2))/deltar)+1;
    endNter= startNter+round(Wter/deltar)+1;

    r= 0:deltar:xmaxTerrain;

    terrainGrid(startNter:endNter)= [ater*(cos((pi*ster/2)*(1-r(startNter:endNter)/xtopter))).^2];
```

Horizontal (contouring) geometry is invoked

For a horizontal (contouring) geometry, both a matrix, *terrainGrid*, defining the terrain heights within the contouring grid space, and the range step, *deltarTer*, with which the matrix is defined, are returned

```
elseif (nargin == 4)
    xminTerrain= variableParameter;
    deltarTer= 1;
    gridside= max([(ymaxTerrain-yminTerrain)/deltarTer+1,(xmaxTerrain-xminTerrain)/deltarTer+1]);
    terrainGrid= zeros(2*gridside,2*gridside);
    addedY= gridside-.5*(ymaxTerrain-yminTerrain)/deltarTer;
    addedX= gridside-.5*(xmaxTerrain-xminTerrain)/deltarTer;
    %THIS IS A SPECIFIC CASE OF TERRAIN!
    Hvect= zeros(1,(ymaxTerrain-yminTerrain)/deltarTer+1);
    ater= 10;
    Wter= 100;
    xtopter= 230;
    ster= 2*xtopter/Wter;
    startNter= round((xtopter-(Wter/2))/deltarTer)+1;
    endNter= startNter+round(Wter/deltarTer)+1;
    r= 0:deltarTer:(ymaxTerrain-yminTerrain);
    Hvect(startNter:endNter)= [ater*(cos((pi*ster/2)*(1-r(startNter:endNter)/xtopter))).^2];
    terrainGrid(addedY:size(terrainGrid,1)-addedY,addedX:size(terrainGrid,2)-addedX)=
    Hvect.'*ones(1,(xmaxTerrain-xminTerrain)/deltarTer+1);
end
```

JRPEFstartCompareRev Function

This function is called to run a (Crank-Nicholson) formulation of the parabolic equation method that does not include terrain, to generate results for the first 3 range steps of the propagation grid

Inputs:

- *directionalSource* - option for using a source directivity pattern ('yes') or assuming an omnidirectional source ('no')
- *freq* - frequency used in this run of the model
- *keta* - wave number profile for each vertical point in the grid, including the effect of atmospheric absorption and attenuating term in the upper absorbing layer
- *kzero* - reference wave number used in this run of the model
- *deltar* - horizontal grid spacing used in the PE propagation model, dependent on frequency. Also constrained to be a divisor of number chosen in 'MultiFreqQuickerRev.m'
- *deltaz* - vertical grid spacing used in the PE and FFP propagation models. Constrained to be a divisor of the receiverHeight so that the receiverHeight will fall exactly at a vertical grid point
- *zN* - number of height points in grid for PE (includes absorbing layer)
- *sourceHeight* - z-coordinate of the current source point along the flight path.
- *Z* - values of the normalized ground impedance of the first 3 range steps

Function Process and Outputs:

1. Calculate the parameters needed in model
2. Calculate M matrices
3. Form the starting field
4. Run PE over range steps, generating the *grid* and *psivector* outputs

References: Appendix G of E. M. Salomons, "Computational Atmospheric Acoustics," 2001, Dr. Sparrow's VQPE, and Dr. Sparrow's Lecture Notes All units are assumed to be MKS.

Start Code Description:	1
Calculate the parameters needed in model.....	2
Calculate M matrices	2
Form the starting field.....	2
Run PE over range steps.....	2

Start Code Description:

```
function [grid,psivector]= JRPEFstartCompareRev(directionalSource,freq,keta,kzero,deltar,deltaz,...
zN,sourceHeight,Z)
```

Calculate the parameters needed in model

```

maxrnum= 3; %range in number of points
grid=zeros(zN,maxrnum); % Zero out grid:

dk2= keta(2:length(keta)).^2-kzero^2;

gamma=j/(2*kzero*deltaz^2);
beta= j*dk2/(2*kzero); % This is a tiny contrast in k

Ztop= 1; %this value doesn't matter because the sound will have been attenuated before reaching top

[tmatrix,sigma1,sigma2]= T_matrixRev(zN-1,keta(1),deltaz,Z,Ztop);

dmatrix= sparse(1:zN-1,1:zN-1,beta,zN-1,zN-1);

```

Calculate M matrices

```
[M1,M2]= M_mats_wideRev(deltar,kzero,gamma,tmatrix,dmatrix);
```

Form the starting field

```

psivectorzero= Start_Field_Wide_Fin_ImpRev(directionalSource,freq,zN,kzero,deltaz,sourceHeight,Z);

% Put starting field in solution matrix as initial range.
grid(:,1)=psivectorzero;

psivector= grid;

```

Run PE over range steps

```

for thisr=2:maxrnum
% Form right hand side matrix M_1 times Psi at range r
rhs= M1*psivector(2:size(psivector,1),thisr-1);
% Solve the linear system of equations
psivector(2:size(psivector,1),thisr)= M2 \ rhs;
psivector(1,thisr)= sigma1*psivector(2,thisr)+sigma2*psivector(3,thisr);
% Put in grid matrix to be ready for next time.
grid(:,thisr)= psivector(:,thisr)/sqrt(deltar*(thisr-1));
end;

```

M_mats_wideRev Function

This function is called to generate the matrices $M1$ and $M2$ for the wide-angle parabolic equation propagation method as calculated by Eq. G.33 in E. M. Salomons, "Computational Atmospheric Acoustics," (2001).

Inputs:

- *deltar* - horizontal grid spacing used in the PE propagation model, dependent on frequency. Also constrained to be a divisor of number chosen in 'MultiFreqQuickerRev.m'
- *kzero* - reference wave number used in this run of the model
- *gamma* - coefficient of T matrix, defined in section G.6 of Salomons "CAA"
- *dmatrix* - diagonal PE matrix, defined in Eq. G.29 of Salomons, "CAA"

Function Process and Outputs:

1. Calculate $M1$ and $M2$ as in Eq. G.33 in Salomons, "Computational Atmospheric Acoustics," (2001).

Start Code Description: 1

Start Code Description:

```
function [M1,M2]= M_mats_wideRev(deltar,kzero,gamma,tmatrix,dmatrix)
identity=sparse(1:length(tmatrix),1:length(tmatrix),ones(1,length(tmatrix)),length(tmatrix),length(tmatrix));
M1=identity+(0.5*deltar)*(gamma*tmatrix+dmatrix)+((gamma*tmatrix+dmatrix)/(2*j*kzero));
M2=identity-(0.5*deltar)*(gamma*tmatrix+dmatrix)+((gamma*tmatrix+dmatrix)/(2*j*kzero));
```

readAtmAbsText Function

This function is called to read the 'User Defined Atm Abs Coeffs' input TXT file.

Inputs:

- *fileString* - the full name of the folder in which the user-input files are located (including an extra slash at the end). (For example, 'E:\Hybrid Propagation Model Version 5 All-Inclusive\Run1\')
- *freqvector* - vector of frequencies to be run in the model

Function Process and Outputs:

1. Function reads in:

- distance over which attenuation coefficients are defined (for example, 1.1 dB per 305 m.
- frequencies for which user coefficients are defined
- vector of defined atmospheric absorption coefficients

2. Function corrects the coefficient to find the attenuation per 1 m

3. Function creates a vector with only attenuation coefficients for each frequency in the freqvector, in the same order as in freqvector *betaAll*

Start Code Description:	1
Read in attenuation coefficient distance, frequency vector, and attenuation coefficient vector.....	1
Correct the coefficient to find the attenuation per 1 m.....	2
Create a vector with only attenuation coefficients for each frequency in the freqvector, in the same order as in freqvector.....	2

Start Code Description:

```
function [betaAll]= readAtmAbsText(fileString,freqvector)

fid= fopen([fileString 'User Defined Atm Abs Coeffs.txt']);

fgetl(fid); fgetl(fid);
```

Read in attenuation coefficient distance, frequency vector, and attenuation coefficient vector

coefDistance, freqvectorAtmAbs, fullcoefvectorAtmAbs

```
coefDistance= textscan(fgetl(fid),'%*s %n %*s');
```

```

coefDistance= coefDistance{1,1};

fgetl(fid);

freqvectorAtmAbs= textscan(fgetl(fid),'%*s %[^H]');
freqvectorAtmAbs= deblank(char(freqvectorAtmAbs{1,1}));
freqvectorAtmAbs= strread(freqvectorAtmAbs);
fullcoefvectorAtmAbs= textscan(fgetl(fid),'%*s %[^d]');
fullcoefvectorAtmAbs= deblank(char(fullcoefvectorAtmAbs{1,1}));

```

Correct the coefficient to find the attenuation per 1 m

```

fullcoefvectorAtmAbs= strread(fullcoefvectorAtmAbs);
fullcoefvectorCorrected= fullcoefvectorAtmAbs/coefDistance;

```

Create a vector with only attenuation coefficients for each frequency in the freqvector, in the same order as in freqvector

```

a= zeros(1,length(freqvector));
for m= 1:length(freqvector)
    freqIndex= find(freqvectorAtmAbs==freqvector(m));
    if isempty(freqIndex)
        error('There is no defined atmospheric absorption coefficient for this frequency. Please check that you have entered the right frequency or that you have defined a coefficient for this frequency.')
    end
    a(m)= fullcoefvectorCorrected(freqIndex);
end

betaAll= a/(20*log10(exp(1)));

```

readText Function

This function is called to read the 'Hybrid Model Parameter' input TXT file.

Inputs:

- *fileString* - the full name of the folder in which the user-input files are located (including an extra slash at the end). (For example, 'E:\Hybrid Propagation Model Version 5 All-Inclusive\Run1\')

Function Process and Outputs:

Function reads in:

1. Model Description

- propagation model to be used ('PE', 'FFP', or 'hybrid') *propagationModel*
- final desired grid geometry ('vertical' or 'horizontal') *gridGeometry*
- axisymmetry of propagation conditions ('yes' or 'no') *axisymmetric*
- option for producing vertical 2D plot ('yes' or 'no') *verticalPlot2D*
- reference source level at 1 m distance *levelat1m*
- directivity of the source ('yes' or 'no') *directionalSource*

2. Frequency Specification

- run for discrete frequencies or 1/3-octave bands ('discrete' or 'bands') *discrete_or_bands*
- vector of discrete frequencies, if 'discrete' was entered *freqvector*
- start and stop frequencies, if 'bands' was entered *startandstop*
- option for saving broadband level data and producing broadband plot ('yes' or 'no') with 1/3 octave bands used in this code run *broadbandResults*

3. Propagation Grid Definition

- grid range *range*
- grid height *height*
- elevation angle limit used for PE *deg*
- increment angle for azimuth *deltatheta*
- receiver height *receiverHeight*
- range step to be saved for plotting purposes *deltarPlot*

4. Terrain

- characteristic of the terrain ('flat' or 'uneven') *terrain*

5. Impedance

- number of different impedances of the ground (1, 2, or 3) *impedances*
- option to input impedance or the effective flow resistivity ('Z' or 'EFR') *inputZorEFR*
- vector with different impedances *ZorEFR*
- vector with range values at which the discontinuities occur *lengthefr*

6. Meteorology--Sound Speed Profile

- characteristic of the atmosphere ('homogeneous' or 'refractive') *atmosphere*
- sound speed at 0 m height *czero*
- type of sound speed profile ('linear' or 'ln') *ctype*
- sound speed parameter for the thermal profile *ssThermal*
- sound speed parameter for the wind profile, in the direction of the wind *ssInWindDirection*
- direction of the wind in degrees where 0 degrees is defined as pointing horizontally right (or East) *windDirection*
- aerodynamic roughness (ignored if ssp is linear) *z0*

7. Meteorology--Turbulence

- option to include turbulence ('yes' or 'no') *turbulence*
- number of realizations to include *realizN*
- turbulence parameters A, K0, CTovT0, and Cvovc0 *A , K0 , CTovT0 , Cvovc0*

8. Meteorology--Atmospheric Absorption

- option for user to define parameter or allow model to calculate it ('user' or 'code') *user_or_code_AtmAbs*
- relative humidity *relativeHumidity*
- absolute temperature *absTemperature*

Start Code Description:	2
Model Description	3
Frequency Specification	3
Propagation Grid Definition	3
Terrain	3
Impedance	4
Meteorology--Sound Speed Profile	4
Meteorology--Turbulence	4
Meteorology--Atmospheric Absorption	5

Start Code Description:

```
function
[propagationModel,gridGeometry,axisymmetric,verticalPlot2D,levelat1m,directionalSource,discrete_or_bands,freq
vector,startandstop,broadbandResults,range,height,deg,deltatheta,receiverHeight,deltaPlot,terrain,impedances,...
```

```

inputZorEFR,ZorEFR,lengthfr,atmosphere,czero,ctype,ssThermal,ssInWindDirection,windDirection,z0,turbulence
,realizN,A,K0,CTovT0,Cvovc0,user_or_code_AtmAbs,...
  relativeHumidity,absTemperature]= readText(fileString)

fid= fopen(fileString);

fgetl(fid);

```

Model Description

```

propagationModel= textscan(fgetl(fid),'%*s %s');
propagationModel= propagationModel{1,1};
gridGeometry= textscan(fgetl(fid),'%*s %s');
gridGeometry= gridGeometry{1,1};
axisymmetric= textscan(fgetl(fid),'%*s %s');
axisymmetric= axisymmetric{1,1};
verticalPlot2D= textscan(fgetl(fid),'%*s %s');
verticalPlot2D= verticalPlot2D{1,1};
levelat1m= textscan(fgetl(fid),'%*s %n %*s');
levelat1m= levelat1m{1,1};
directionalSource= textscan(fgetl(fid),'%*s %s');
directionalSource= directionalSource{1,1};

fgetl(fid); fgetl(fid);

```

Frequency Specification

```

discrete_or_bands= textscan(fgetl(fid),'%*s %s');
discrete_or_bands= discrete_or_bands{1,1};
freqvector= textscan(fgetl(fid),'%*s %[^H]');
freqvector= deblank(char(freqvector{1,1}));
freqvector= strread(freqvector);
startandstop= textscan(fgetl(fid),'%*s %n %n %*s');
startandstop= [startandstop{1,1} startandstop{1,2}];
broadbandResults= textscan(fgetl(fid),'%*s %s');
broadbandResults= broadbandResults{1,1};

fgetl(fid); fgetl(fid);

```

Propagation Grid Definition

```

range= textscan(fgetl(fid),'%*s %n %*s');
range= range{1,1};
height= textscan(fgetl(fid),'%*s %n %*s');
height= height{1,1};
deg= textscan(fgetl(fid),'%*s %n %*s');
deg= deg{1,1};
deltatheta= textscan(fgetl(fid),'%*s %n %*s');
deltatheta= deltatheta{1,1};
receiverHeight= textscan(fgetl(fid),'%*s %n %*s');
receiverHeight= receiverHeight{1,1};
deltarPlot= textscan(fgetl(fid),'%*s %n %*s');
deltarPlot= deltarPlot{1,1};

fgetl(fid);

```

Terrain

```

terrain= textscan(fgetl(fid),'%*s %s');
terrain= terrain{1,1};

fgetl(fid); fgetl(fid);

```

Impedance

```

impedances= textscan(fgetl(fid),'%*s %n');
impedances= impedances{1,1};
inputZorEFR= textscan(fgetl(fid),'%*s %s');
inputZorEFR= inputZorEFR{1,1};
ZorEFR= zeros(1,impedances);
for n= 1:impedances
    ZorEFRText= textscan(fgetl(fid),'%*s %n %*s');
    ZorEFR(n)= ZorEFRText{1,1};
end
for n= impedances+1:3
    textscan(fgetl(fid),'%*s %n %*s');
end
lengthfr= zeros(1,impedances-1);
for m= 1:impedances-1
    lengthfrText= textscan(fgetl(fid),'%*s %n %*s');
    lengthfr(m)= lengthfrText{1,1};
end
for m= impedances:2
    textscan(fgetl(fid),'%*s %n %*s');
end

fgetl(fid); fgetl(fid);

```

Meteorology--Sound Speed Profile

```

atmosphere= textscan(fgetl(fid),'%*s %s');
atmosphere= atmosphere{1,1};
czero= textscan(fgetl(fid),'%*s %n %*s');
czero= czero{1,1};
ctype= textscan(fgetl(fid),'%*s %s');
ctype= ctype{1,1};
ssThermal= textscan(fgetl(fid),'%*s %n %*s %*%s %*%s');
ssThermal= ssThermal{1,1};
ssInWindDirection= textscan(fgetl(fid),'%*s %n %*%s %*%s %*%s');
ssInWindDirection= ssInWindDirection{1,1};
windDirection= textscan(fgetl(fid),'%*s %n %*%s');
windDirection= windDirection{1,1};
z0= textscan(fgetl(fid),'%*s %n %*%s');
z0= z0{1,1};

fgetl(fid); fgetl(fid);

```

Meteorology--Turbulence

```

turbulence= textscan(fgetl(fid),'%*%s %s');
turbulence= turbulence{1,1};
realizN= textscan(fgetl(fid),'%*s %n');
realizN= realizN{1,1};
A= textscan(fgetl(fid),'%*%s %n');
A= A{1,1};
K0= textscan(fgetl(fid),'%*%s %n %*%s');
K0= K0{1,1};
CTovT0= textscan(fgetl(fid),'%*%s %n %*%s');
CTovT0= CTovT0{1,1};
Cvove0= textscan(fgetl(fid),'%*%s %n %*%s');
Cvove0= Cvove0{1,1};

fgetl(fid); fgetl(fid);

```

Meteorology--Atmospheric Absorption

```
user_or_code_AtmAbs= textscan(fgetl(fid),'%*s %s');
user_or_code_AtmAbs= user_or_code_AtmAbs{1,1};
relativeHumidity= textscan(fgetl(fid),'%*s %n %*s');
relativeHumidity= relativeHumidity{1,1};
absTemperature= textscan(fgetl(fid),'%*s %n %*s');
absTemperature= absTemperature{1,1};

fclose(fid);
```

SigmatasRev Function

This function returns the coefficients σ_1 and σ_2 , which depend on the normalized ground impedance, as used in Eq. G.25 in Salomons, "Computation Atmospheric Acoustics," (2001).

Inputs:

- k_0 - wavenumber at the ground
- dz - vertical grid spacing used in the PE and FFP propagation models. Constrained to be a divisor of the receiverHeight so that the receiverHeight will fall exactly at a vertical grid point
- Z - values of the normalized ground impedance of the first 3 range steps

Function Process and Outputs:

1. Calculate σ_1 and σ_2 as in Eq. G.39 in Salomons, "Computational Atmospheric Acoustics," (2001).

Start Code Description: 1

Start Code Description:

```
function [sigma1,sigma2]= SigmasRev(k0,dz,Z)

sigma1= ((3-(2*j*k0*dz)/Z).^-1)^4;
sigma2= -(3-(2*j*k0*dz)/Z).^-1;
```

Start_Field_Wide_Fin_ImpRev Function

This function is called to calculate the starting field for the wide-angle PE model either for an omnidirectional or directional source

Inputs:

- *directionalSource* - option for using a source directivity pattern ('yes') or assuming an omnidirectional source ('no')
- *freq* - frequency used in this run of the model
- *zN* - number of height points in grid for PE (includes absorbing layer)
- *kzero* - reference wave number used in this run of the model
- *deltaz* - vertical grid spacing used in the PE and FFP propagation models. Constrained to be a divisor of the receiverHeight so that the receiverHeight will fall exactly at a vertical grid point
- *sourceHeight* - z-coordinate of the current source point along the flight path.
- *Z* - values of the normalized ground impedance of the first 3 range steps

Function Process and Outputs:

1. Define constants used for determining Gaussian function for omnidirectional sources
2. If the user chooses the omnidirectional source option, use the standard, wide-angle Gaussian representation
3. If the user chooses the directional source option, use array theory and windowing to calculate the starting field
4. Create directivity pattern vector
5. Define the wave number window function to smoothly taper the evanescent wave numbers
6. Apply the window function to the directivity profile
7. Prepare directivity pattern for Matlab ifft algorithm
8. Calculate array weightings $A_{\text{shift}}(z)$
9. Assign the array values at points above the ground to *psivectorzerosource* and array values at points below the ground to *psivectorzeroimage*, accounting for array point spacing
10. Calculate the starting field *psivectorzero* including the effect of reflections off the ground

Start Code Description:	2
Start_Field_Wilde_Fin_ImpRev_window.m file (for Approach 2).	2
Define constants used for determining Gaussian function for omnidirectional sources,	2
If the user chooses the omnidirectional source option, use the standard, wide-angle Gaussian representation.....	2
If the user chooses the directional source option, use array theory and windowing to calculate the starting field.....	2
Create directivity pattern vector	3

Define the wave number window function to smoothly taper the evanescent wave numbers 3
 Apply the window function to the directivity profile 3
 Prepare directivity pattern for Matlab ifft algorithm 3
 Calculate array weightings $A_{\text{shift}}(z)$ 3
 The array values at points above the ground to $\text{psivectorzerosource}$ and array values at points below the ground to $\text{psivectorzeroimage}$, accounting for array point spacing 4
 Calculate the starting field including the effect of reflections off the ground..... 4

Start Code Description:

```
function [psivectorzero]= Start_Field_Wide_Fin_ImpRev(directionalSource,freq,zN,kzero,deltaz,sourceHeight,Z)
```

Start_Field_Wilde_Fin_ImpRev_window.m file (for Approach 2).

This function calculates the start field for a wide-angle parabolic equation with a given directivity.

```
% It uses a vector that describes the user-input directivity profile at the necessary elevation angles to
% achieve array point weightings  $A_{\text{shift}}(z)$ , related to pressure, at the vertical initial field
% gridpoints (as dictated by  $\text{deltaz}$ ).
```

Define constants used for determining Gaussian function for omnidirectional sources,

as given in E. M. Salomons "Computational Atmospheric Acoustics," (2001), p. 179, under Ex. G.75

```
A0=1.3717;
A2= -.3701;
B= 3;

z=0:deltaz:zN*deltaz-deltaz;    %z vector
R2= z+sourceHeight;
```

If the user chooses the omnidirectional source option, use the standard, wide-angle Gaussian representation

```
if strcmp(directionalSource,'no') % if the source is assumed to be omnidirectional
psivectorzerosource= sqrt(j*kzero)*(A0+A2*kzero^2*(z-sourceHeight).^2).*exp(-kzero^2*(z-sourceHeight).^2/B);
psivectorzeroimage= sqrt(j*kzero)*(A0+A2*kzero^2*R2.^2).*exp(-kzero^2*R2.^2/B);
```

If the user chooses the directional source option, use array theory and windowing to calculate the starting field

```
elseif strcmp(directionalSource,'yes') % if the source has a directivity pattern

lambda= 2*pi/kzero;                %wavelength
N= 1025;                            %number of elements in approximated array

%Determine the full length and span of the FFT, with zero-padding
```

```

spanZeroPad= lambda/deltaz; %span ( S= N*sin(phi) ) needed to acheive correct
array spacing of deltax %sin(phi) increment based on necessary span
deltasinephi= spanZeroPad/N;

sinphi= -deltasinephi*((N-1)/2):deltasinephi:deltasinephi*((N-1)/2); %define the sin(phi) vector--this will
have abs(sin(phi))>1, which is non-physical, but necessary to achieve the deltax
phi= asin(sinphi);
nonZeroArrayIndices= find(abs(sinphi)<=1);%define the elevation angles at which to define the directivity pattern

```

Create directivity pattern vector

```

% Initialized the directivity vector
dirPattern= zeros(1,N);

% Define the directivity pattern at designated elevation angles (in the
% non-evanescent wave number range
dirPattern(nonZeroArrayIndices)= cos(phi(nonZeroArrayIndices)); %define the directivity pattern at
designated elevation angles

% Assign the value of the lowest non-evanescent wave number to all negative evanescent wave numbers
dirPattern(1:nonZeroArrayIndices(1)-1)= dirPattern(nonZeroArrayIndices(1));
% Assign the value of the highest non-evanescent wave number to all positive evanescent wave numbers
dirPattern(nonZeroArrayIndices(end)+1:end)= dirPattern(nonZeroArrayIndices(end));

```

Define the wave number window function to smoothly taper the evanescent wave numbers

```

alpha= 0.1;
dirFilter= .5*exp((1-abs(sinphi))/alpha);
dirFilter(nonZeroArrayIndices)= 1-.5*exp(-(1-abs(sinphi(nonZeroArrayIndices)))/alpha);

```

Apply the window function to the directivity profile

```

dirPattern= (dirPattern./(1-sin(phi).^2/4)).*dirFilter;

```

Prepare directivity pattern for Matlab ifft algorithm

```

dirPatternshift= [dirPattern((N+1)/2:N) dirPattern(1:(N-1)/2)];

% Define the heights of the array sources
zArrayIndex= (deltaz*round(sourceHeight/deltaz))-(deltaz*(N-1)/2):deltaz:(deltaz*round(sourceHeight/deltaz)+(deltaz*(N-1)/2));

```

Calculate array weightings Ashift(z)

```

% Account for a phase shift caused by the source height not falling
% directly on a gridpoint
delta= deltax*round(sourceHeight/deltaz)-sourceHeight;
expterm= exp(j*2*pi*delta*deltasinephi*[0:(N-1)]/lambda);
expshift= [expterm((N+1)/2:N) expterm(1:(N-1)/2)];

% Apply the phase shift and determine the array source values
A= ifft(dirPatternshift.*expshift)/deltaz;
Ashift= [A((N+1)/2+1:N) A(1:(N+1)/2)]*sqrt(i*lambda); %get A(z) back into form user
expects

```

The array values at points above the ground to psivectorzerosource and array values at points below the ground to psivectorzeroimage, accounting for array point spacing

```

% Initialize vectors
psivectorzerosource= zeros(1,zN);
psivectorzeroimage= zeros(1,zN);

indicesAboveGround= find(zArrayIndex>=0,1,'first'):find(zArrayIndex<=z(end),1,'last');
indicesBelowGround= find(zArrayIndex<=0);
psivectorzerosource(round(zArrayIndex(indicesAboveGround)/deltaz)+1)= ((Ashift(indicesAboveGround)));
psivectorzeroimage(-round(zArrayIndex(indicesBelowGround)/deltaz)+1)= ((Ashift(indicesBelowGround)));

else error('The entry for the "Directional_Source" input category is invalid. Please enter either "yes" or "no".')
end

if freq<80
expVal= kzero*((Z+1)/(Z))*(R2/j);
Cs= 1-2*(kzero/Z)*(R2/j).*exp(expVal).*expint(expVal);
else
Cp= (Z-1)/(Z+1);

n= 1:2;
Cs= zeros(1,zN);
for thisz= 1:zN
Cs(thisz)= Cp-(2/(Z+1))*sum((-1).^n.*factorial(n).*(i./(kzero*R2(thisz)))*(Z/(Z+1)).^n);
end
end

```

Calculate the starting field including the effect of reflections off the ground

```
psivectorzero= psivectorzerosource+Cs.*psivectorzeroimage;
```

T_MatrixRev Function

This function is called to generate the tridiagonal T matrix, as described by Eq. G.28 in E. M. Salomons, "Computational Atmospheric Acoustics," (2001). It also returns the *sigma1* and *sigma2* values as used in Eq. G.25 in Salomons, "CAA."

Inputs:

- *Nz* - number of elements in both dimensions of the T matrix. It is 1 smaller than the number of vertical grid points *zN*
- *k0* - wavenumber at the ground
- *deltaz* - vertical grid spacing used in the PE and FFP propagation models. Constrained to be a divisor of the receiverHeight so that the receiverHeight will fall exactly at a vertical grid point
- *Z* - values of the normalized ground impedance of the first 3 range steps
- *Ztop* - value of the normalized impedance at the top of the grid

Function Process and Outputs:

1. Form the basis of the T matrix
2. Run functions to get boundary condition parameters (including *sigma1* and *sigma2*, outputs of this function)
3. Finish forming tridiagonal matrix T by adding in boundary conditions

```
*Start Code Description:* ..... 1
Form the basis of the T matrix..... 1
Run functions to get boundary condition parameters ..... 1
Finish forming tridiagonal matrix T by adding in boundary conditions..... 2
```

Start Code Description:

```
function [tmatrix,sigma1,sigma2]= T_matrixRev(Nz,k0,deltaz,Z,Ztop)
```

Form the basis of the T matrix

```
D=sparse(1:Nz,1:Nz,-2*ones(1,Nz),Nz,Nz);
E=sparse(2:Nz,1:Nz-1,1*ones(1,Nz-1),Nz,Nz);
tmatrix=D+E+E';
```

Run functions to get boundary condition parameters

```
[sigma1,sigma2]= SigmasRev(k0,deltaz,Z);
[tau1,tau2]= TausRev(k0,deltaz,Ztop);
```

Finish forming tridiagonal matrix T by adding in boundary conditions.

```
tmatrix(1,1)=tmatrix(1,1)+sigma1;  
tmatrix(1,2)=tmatrix(1,2)+sigma2;  
tmatrix(Nz,Nz)=tmatrix(Nz,Nz)+tau1;  
tmatrix(Nz,Nz-1)=tmatrix(Nz,Nz-1)+tau2;
```

TausRev Function

This function returns the coefficients τ_1 and τ_2 , which depend on the normalized impedance at the top of the grid, as used in Eq. G.26 in Salomons, "Computation Atmospheric Acoustics," (2001).

Inputs:

- k_0 - wavenumber at the ground
- dz - vertical grid spacing used in the PE and FFP propagation models. Constrained to be a divisor of the receiverHeight so that the receiverHeight will fall exactly at a vertical grid point
- Z_{top} - value of the normalized impedance at the top of the grid

Function Process and Outputs:

1. Calculate τ_1 and τ_2 as in Eq. G.40 in Salomons, "Computational Atmospheric Acoustics," (2001).

Start Code Description: 1

Start Code Description:

```
function [tau1,tau2]= TausRev(k0,dz,Ztop)
```

```
tau1= ((3+(2*j*k0*dz)/Ztop).^-1)*4;
```

```
tau2= -((3+(2*j*k0*dz)/Ztop).^-1);
```

TerrainTransformV3 Function

This function is called to return information about the terrain and impedance geometries in the contouring grid, broken down for each azimuthal angle run for the current point along the flight path

Inputs:

- *thetas* - vector of azimuthal angles at which runs will be performed outward from the source
- *runN* - number of runs needed to cover all azimuthal angles from source
- *xflightPositions* - vector defining the x-coordinate of the current represented point along the flight path
- *yflightPositions* - vector defining the y-coordinate of the current represented point along the flight path
- *freq* - frequency used in this run of the model
- *deltar* - horizontal grid spacing at current frequency
- *inputZorEFR* - option to input impedance or the effective flow resistivity ('Z' or 'EFR')
- *xminTerrain* - lowest value of the x-axis to which the contouring grid extends
- *xmaxTerrain* - highest value of the x-axis to which the contouring grid extends
- *yminTerrain* - lowest value of the y-axis to which the contouring grid extends
- *ymaxTerrain* - highest value of the y-axis to which the contouring grid extends
- *terrain* - characteristic of the terrain ('flat' or 'uneven')

Function Process and Outputs:

1. Initialize matrices to be used
2. Find terrain and impedance parameters for each azimuthal angle run
3. Calculate propagation range *rangesAtThetas* and associated parameters at the different azimuthal angle value
4. Interpolate the terrain and impedance values from data points on the rectangular terrain and impedance grids to polar points on the source propagation run grid (interpolated terrain stored in *polarTerrain*). The number of different impedances at each azimuthal angle run *impedancesAtThetas*, the values of the different impedances *ZAtThetas*, and the ranges at which the impedance transitions occur *lengthefrAtThetas* are also calculated.

Start Code Description:	2
Import the terrain and impedance data	2
Initialize matrices to be used	2
Find terrain and impedance parameters for each azimuthal angle run	2
Plot terrain and impedance features	7

Start Code Description:

```
function [rangesAtThetas,polarTerrain,impedancesAtThetas,ZAtThetas,lengthefrAtThetas]=
TerrainTransformV3(thetas,runN,xflightPositions,yflightPositions,freq,deltar,deltarPlot,inputZorEFR,xminTerrain,
xmaxTerrain,yminTerrain,ymaxTerrain,terrain)
```

Import the terrain and impedance data

```
if strcmp(terrain,'flat')
deltarTer= 1; % assume a deltarTer=1. This assignment is arbitrary.
gridside= max((ymaxTerrain-yminTerrain)/deltarTer+1,(xmaxTerrain-xminTerrain)/deltarTer+1);
terrainGrid= zeros(2*gridside,2*gridside);
addedY= gridside-.5*(ymaxTerrain-yminTerrain)/deltarTer;
addedX= gridside-.5*(xmaxTerrain-xminTerrain)/deltarTer;

elseif strcmp(terrain,'uneven')
[terrainGrid,deltarTer]= ImportTerrainGrid(xmaxTerrain,xminTerrain,ymaxTerrain,yminTerrain);
gridside= max((ymaxTerrain-yminTerrain)/deltarTer+1,(xmaxTerrain-xminTerrain)/deltarTer+1);
addedY= gridside-.5*(ymaxTerrain-yminTerrain)/deltarTer;
addedX= gridside-.5*(xmaxTerrain-xminTerrain)/deltarTer;
else error('Please enter either "uneven" or "flat" in the "Terrain" user-input')
end

[impedanceGrid]= ImportImpedanceGrid(gridside,-gridside,gridside,-gridside,deltarTer);
```

Initialize matrices to be used

```
% Define the different ranges for the different azimuthal angles
rangesAtThetasShort= zeros(1,runN); %unadjusted vector of range values at each theta angle from source
rangesAtThetas= zeros(1,runN); %vector of range values at each theta angle from source

% Define the terrain and impedance cell arrays
polarTerrain= cell(runN,1);
polarImpedance= cell(runN,1);

% Define the x and y values of the polar points
xRangeValsArray= cell(runN,1);
yRangeValsArray= cell(runN,1);

% Define the impedance parameters for the different azimuthal angles
impedancesAtThetas= zeros(runN,1); %number of different impedances outward propagation will see for each
azimuthal angle run
ZorEFR= zeros(runN,3); %impedance parameter values for each of the different impedances (could be an
impedance value or an effective flow resistivity) for each azimuthal run. A max of 3 impedances can be included
ZAtThetas= zeros(runN,3); %impedance (not efr) values for each of the different impedances for each
azimuthal run
lengthefrAtThetas= zeros(runN,2); %distances at which impedance transitions occur for each azimuthal run

thetaUR= atan((ymaxTerrain-yflightPositions)/(xmaxTerrain-xflightPositions));
thetaUL= atan((ymaxTerrain-yflightPositions)/(xminTerrain-xflightPositions))+pi;
thetaLL= atan((yminTerrain-yflightPositions)/(xminTerrain-xflightPositions))+pi;
thetaLR= atan((yminTerrain-yflightPositions)/(xmaxTerrain-xflightPositions))+2*pi;
```

Find terrain and impedance parameters for each azimuthal angle run

```
for thistheta= 1:runN

% Define range values for different thetas at the specific theta values
% of 0, pi/2, pi, 3pi/2, and 2pi
if thetas(thistheta)==0
rangesAtThetasShort(thistheta)= deltar*ceil(abs(xmaxTerrain-xflightPositions)/deltar); %distance (using and
integer number of range grid points) to the max x boundary
```

```

elseif thetas(thistheta)==pi/2
    rangesAtThetasShort(thistheta)= deltar*ceil(abs(ymaxTerrain-yflightPositions)/deltar); %distance (using and
integer number of range grid points) to the max y boundary
elseif thetas(thistheta)==pi
    rangesAtThetasShort(thistheta)= deltar*ceil(abs(xflightPositions-xminTerrain)/deltar); %distance (using and
integer number of range grid points) to the min x boundary
elseif thetas(thistheta)==3*pi/2
    rangesAtThetasShort(thistheta)= deltar*ceil(abs(yflightPositions-yminTerrain)/deltar); %distance (using and
integer number of range grid points) to the min y boundary
elseif thetas(thistheta)==2*pi
    rangesAtThetasShort(thistheta)= deltar*ceil(abs(xmaxTerrain-xflightPositions)/deltar); %distance (using and
integer number of range grid points) to the max x boundary (again)
else
    % Define important parameters for finding the ranges at different
    % thetas

    % For first quadrant
    if 0<thetas(thistheta) && thetas(thistheta)<pi/2
        extreme_xVal= xmaxTerrain-xflightPositions;
        extreme_yVal= ymaxTerrain-yflightPositions;

    % For second quadrant
    elseif pi/2<thetas(thistheta) && thetas(thistheta)<pi
        extreme_xVal= xminTerrain-xflightPositions;
        extreme_yVal= ymaxTerrain-yflightPositions;

    % For third quadrant
    elseif pi<thetas(thistheta) && thetas(thistheta)<3*pi/2
        extreme_xVal= xminTerrain-xflightPositions;
        extreme_yVal= yminTerrain-yflightPositions;

    % For fourth quadrant
    elseif 3*pi/2<thetas(thistheta) && thetas(thistheta)<2*pi
        extreme_xVal= xmaxTerrain-xflightPositions;
        extreme_yVal= yminTerrain-yflightPositions;

    else error('Something is wrong with the azimuthal angles, out of the range of 0 to 2*pi?')
    end

    %% Calculate propagation range and associated parameters at the current azimuthal angle value, making sure
the range value is a multiple of deltar.
    % Range will be dependent on which boundary--top, bottom, left,
    % right--is encountered for each angle.)
    rangesAtThetasShort(thistheta)=
deltar*ceil((min(abs(extreme_yVal/sin(thetas(thistheta))),abs(extreme_xVal/cos(thetas(thistheta)))))/deltar);
    end

end

for thistheta= 1:runN

    if thistheta==1 || thistheta==runN
        rangesAtThetas(thistheta)= max([rangesAtThetasShort(runN-1) rangesAtThetasShort(1:2)])+deltarPlot;
        if (thetaUR >= thetas(thistheta) && thetaUR<= thetas(thistheta+1)) ||(thetaUR >= thetas(runN-1) &&
thetaUR<= thetas(thistheta))
            rangesAtThetas(thistheta)= max([sqrt((ymaxTerrain-yflightPositions)^2+(xmaxTerrain-
xflightPositions)^2)+deltarPlot,rangesAtThetas(thistheta)]);
            elseif (thetaUL >= thetas(thistheta) && thetaUL<= thetas(thistheta+1)) ||(thetaUL >= thetas(runN-1) &&
thetaUL<= thetas(thistheta))
            rangesAtThetas(thistheta)= max([sqrt((ymaxTerrain-yflightPositions)^2+(xminTerrain-
xflightPositions)^2)+deltarPlot,rangesAtThetas(thistheta)]);
            elseif (thetaLL >= thetas(thistheta) && thetaLL<= thetas(thistheta+1)) ||(thetaLL >= thetas(runN-1) &&
thetaLL<= thetas(thistheta))
            rangesAtThetas(thistheta)= max([sqrt((yminTerrain-yflightPositions)^2+(xminTerrain-
xflightPositions)^2)+deltarPlot,rangesAtThetas(thistheta)]);
            elseif (thetaLR >= thetas(thistheta) && thetaLR<= thetas(thistheta+1)) ||(thetaLR >= thetas(runN-1) &&
thetaLR<= thetas(thistheta))
            rangesAtThetas(thistheta)= max([sqrt((yminTerrain-yflightPositions)^2+(xmaxTerrain-
xflightPositions)^2)+deltarPlot,rangesAtThetas(thistheta)]);
            end
        else rangesAtThetas(thistheta)= max([rangesAtThetasShort(thistheta-1:thistheta+1)])+deltarPlot;

```

```

    if (thetaUR >= thetas(thistheta) && thetaUR<= thetas(thistheta+1)) ||(thetaUR >= thetas(thistheta-1) &&
thetaUR<= thetas(thistheta))
        rangesAtThetas(thistheta)= max([sqrt((ymaxTerrain-yflightPositions)^2+(xmaxTerrain-
xflightPositions)^2)+deltarPlot,rangesAtThetas(thistheta)]);
    elseif (thetaUL >= thetas(thistheta) && thetaUL<= thetas(thistheta+1)) ||(thetaUL >= thetas(thistheta-1) &&
thetaUL<= thetas(thistheta))
        rangesAtThetas(thistheta)= max([sqrt((ymaxTerrain-yflightPositions)^2+(xminTerrain-
xflightPositions)^2)+deltarPlot,rangesAtThetas(thistheta)]);
    elseif (thetaLL >= thetas(thistheta) && thetaLL<= thetas(thistheta+1)) ||(thetaLL >= thetas(thistheta-1) &&
thetaLL<= thetas(thistheta))
        rangesAtThetas(thistheta)= max([sqrt((yminTerrain-yflightPositions)^2+(xminTerrain-
xflightPositions)^2)+deltarPlot,rangesAtThetas(thistheta)]);
    elseif (thetaLR >= thetas(thistheta) && thetaLR<= thetas(thistheta+1)) ||(thetaLR >= thetas(thistheta-1) &&
thetaLR<= thetas(thistheta))
        rangesAtThetas(thistheta)= max([sqrt((yminTerrain-yflightPositions)^2+(xmaxTerrain-
xflightPositions)^2)+deltarPlot,rangesAtThetas(thistheta)]);
    end
end

% Define the rho vector (called 'r' in other areas of propagation
% code, but rho here keeping with notation for polar coordinate
% systems) using deltar and the appropriate range for the given theta

rho= 0:deltar:deltar*ceil(rangesAtThetas(thistheta)/deltar);
rhoValsN= length(rho);

% Define cell arrays storing x and y values for the polar coordinate
% system with origin at the source and the rectangular
% coordinate system boundaries of the given terrain grid. Cell arrays
% are constructed by forming vectors of values for each azimuthal angle
% run, starting at the lowest angle of the theta vector, and storing
% each vector in a cell (making runN number of cells).
xRangeValsArray{thistheta}= rho*cos(thetas(thistheta));
yRangeValsArray{thistheta}= rho*sin(thetas(thistheta));

%% Interpolate the terrain and impedance values from data points on the rectangular terrain and impedance
grids to polar points on the source propagation run grid.
% Loop only goes to rhoValsN-1 because interpolation requires data from
% points at larger rho values, which we do not have for rho(rhoValsN).
% (Used equations from p. 116-117 in Numerical Recipes in FORTRAN,
% The Art of Scientific Computing, Second Edition, Press, Teukolsky,
% Vetterling, Flannery)
for thisrho= 1:rhoValsN-1

    if rem(xRangeValsArray{thistheta}(thisrho),deltarTer)==0 %if the x-value of the current range point along
the current azimuthal angle run falls exactly at the x-value of a point on the terrain grid
        if rem(yRangeValsArray{thistheta}(thisrho),deltarTer)==0 %if the y-value of the current range point along
the current azimuthal angle run falls exactly at the y-value of a point on the terrain grid
            polarTerrain{thistheta}(thisrho)= terrainGrid(-
yminTerrain/deltarTer+yflightPositions/deltarTer+round(yRangeValsArray{thistheta}(thisrho)/deltarTer)+1,-
xminTerrain/deltarTer+xflightPositions/deltarTer+round(xRangeValsArray{thistheta}(thisrho)/deltarTer)+1);
            %set the terrain height at the current range point equal to the terrain height at the terrain grid point it falls on
            polarImpedance{thistheta}(thisrho)= impedanceGrid(-
yminTerrain/deltarTer+yflightPositions/deltarTer+round(yRangeValsArray{thistheta}(thisrho)/deltarTer)+1,-
xminTerrain/deltarTer+xflightPositions/deltarTer+round(xRangeValsArray{thistheta}(thisrho)/deltarTer)+1);
            %set the impedance parameter at the current range point equal to the terrain height at the terrain grid point it falls
on
        else
            floorY= deltarTer*floor(yRangeValsArray{thistheta}(thisrho)/deltarTer); %find the y-value of the
nearest point on the terrain grid with a lower y-value
            ceilY= floorY+deltarTer; %find the y-value of the nearest point on the terrain grid with a higher y-value

            u= (yRangeValsArray{thistheta}(thisrho)-floorY)/deltarTer;

            z1= terrainGrid(-yminTerrain/deltarTer+yflightPositions/deltarTer+round(floorY/deltarTer)+1+addedY,-
xminTerrain/deltarTer+xflightPositions/deltarTer+round(xRangeValsArray{thistheta}(thisrho)/deltarTer)+1+adde
dX);
            z4= terrainGrid(-yminTerrain/deltarTer+yflightPositions/deltarTer+round(ceilY/deltarTer)+1+addedY,-
xminTerrain/deltarTer+xflightPositions/deltarTer+round(xRangeValsArray{thistheta}(thisrho)/deltarTer)+1+adde
dX);

```

```

    polarTerrain{thistheta}(thisrho)= (1-u)*z1+u*z4; %interpolate the terrain height linearly between the
    terrain heights at the 2 nearest terrain grid points (will have same x-values)

    if u>.5 && u<1 %if the current range point is closer to the impedance grid point with a higher y-value
        polarImpedance{thistheta}(thisrho)= impedanceGrid(-
        yminTerrain/deltaTer+yflightPositions/deltaTer+round(ceilY/deltaTer)+1+addedY,-
        xminTerrain/deltaTer+xflightPositions/deltaTer+round(xRangeValsArray{thistheta}(thisrho)/deltaTer)+1+adde
        dX); %set the impedance parameter at the current range point equal to the impedance parameter at the impedance
        grid point with the higher y-value
        elseif (u<.5 && u>0) || u==.5 %if the current range point is closer to the impedance grid point with a lower
        y-value, or equally spaced between the upper and lower points
            polarImpedance{thistheta}(thisrho)= impedanceGrid(-
            yminTerrain/deltaTer+yflightPositions/deltaTer+round(floorY/deltaTer)+1+addedY,-
            xminTerrain/deltaTer+xflightPositions/deltaTer+round(xRangeValsArray{thistheta}(thisrho)/deltaTer)+1+adde
            dX); %set the impedance parameter at the current range point equal to the impedance parameter at the impedance
            grid point with the lower y-value
        else error('The u interpolation value is not in the correct range of 0<u<1')
        end
    end
    elseif rem(yRangeValsArray{thistheta}(thisrho),deltaTer)==0 %else if the y-value of the current range point
    along the current azimuthal angle run falls exactly at the y-value of a point on the terrain grid (but not exactly on the
    x)
        floorX= deltaTer*floor(xRangeValsArray{thistheta}(thisrho)/deltaTer); %find the x-value of the nearest
        point on the terrain grid with a lower x-value
        ceilX= floorX+deltaTer; %find the x-value of the nearest point on the terrain grid with a higher x-value

        t= (xRangeValsArray{thistheta}(thisrho)-floorX)/deltaTer;

        z1= terrainGrid(-
        yminTerrain/deltaTer+yflightPositions/deltaTer+round(yRangeValsArray{thistheta}(thisrho)/deltaTer)+1+adde
        dY,-xminTerrain/deltaTer+xflightPositions/deltaTer+round(floorX/deltaTer)+1+addedX);
        z2= terrainGrid(-
        yminTerrain/deltaTer+yflightPositions/deltaTer+round(yRangeValsArray{thistheta}(thisrho)/deltaTer)+1+adde
        dY,-xminTerrain/deltaTer+xflightPositions/deltaTer+round(ceilX/deltaTer)+1+addedX);

        polarTerrain{thistheta}(thisrho)= (1-t)*z1+t*z2; %interpolate the terrain height linearly between the
        terrain heights at the 2 nearest terrain grid points (will have same y-values)

        if t>.5 && t<1
            polarImpedance{thistheta}(thisrho)= impedanceGrid(-
            yminTerrain/deltaTer+yflightPositions/deltaTer+round(yRangeValsArray{thistheta}(thisrho)/deltaTer)+1+adde
            dY,-xminTerrain/deltaTer+xflightPositions/deltaTer+round(ceilX/deltaTer)+1+addedX); %set the impedance
            parameter at the current range point equal to the impedance parameter at the impedance grid point with the higher
            x-value
            elseif (t<.5 && t>0) || t==.5
                polarImpedance{thistheta}(thisrho)= impedanceGrid(-
                yminTerrain/deltaTer+yflightPositions/deltaTer+round(yRangeValsArray{thistheta}(thisrho)/deltaTer)+1+adde
                dY,-xminTerrain/deltaTer+xflightPositions/deltaTer+round(floorX/deltaTer)+1+addedX); %set the impedance
                parameter at the current range point equal to the impedance parameter at the impedance grid point with the lower
                x-value
            else error('The t interpolation value is not in the correct range of 0<t<1')
            end
        else
            floorX= deltaTer*floor(xRangeValsArray{thistheta}(thisrho)/deltaTer); %find the x-value of the nearest
            point on the terrain grid with a lower x-value
            ceilX= floorX+deltaTer; %find the x-value of the nearest point on the terrain grid with a higher x-value
            floorY= deltaTer*floor(yRangeValsArray{thistheta}(thisrho)/deltaTer); %find the y-value of the nearest
            point on the terrain grid with a lower y-value
            ceilY= floorY+deltaTer; %find the y-value of the nearest point on the terrain grid with a higher y-value

            t= (xRangeValsArray{thistheta}(thisrho)-floorX)/deltaTer;
            u= (yRangeValsArray{thistheta}(thisrho)-floorY)/deltaTer;

            z1= terrainGrid(-yminTerrain/deltaTer+yflightPositions/deltaTer+round(floorY/deltaTer)+1+addedY,-
            xminTerrain/deltaTer+xflightPositions/deltaTer+round(floorX/deltaTer)+1+addedX);
            z2= terrainGrid(-yminTerrain/deltaTer+yflightPositions/deltaTer+round(floorY/deltaTer)+1+addedY,-
            xminTerrain/deltaTer+xflightPositions/deltaTer+round(ceilX/deltaTer)+1+addedX);
            z3= terrainGrid(-yminTerrain/deltaTer+yflightPositions/deltaTer+round(ceilY/deltaTer)+1+addedY,-
            xminTerrain/deltaTer+xflightPositions/deltaTer+round(ceilX/deltaTer)+1+addedX);
            z4= terrainGrid(-yminTerrain/deltaTer+yflightPositions/deltaTer+round(ceilY/deltaTer)+1+addedY,-
            xminTerrain/deltaTer+xflightPositions/deltaTer+round(floorX/deltaTer)+1+addedX);

```

```

    polarTerrain{thistheta}(thisrho) = (1-t)*(1-u)*z1+t*(1-u)*z2+t*u*z3+(1-t)*u*z4; %interpolate the terrain
    height linearly between the terrain heights at the 4 nearest terrain grid points

    impedValue1= impedanceGrid(-
    yminTerrain/deltaTer+yflightPositions/deltaTer+round(floorY/deltaTer)+1+addedY,-
    xminTerrain/deltaTer+xflightPositions/deltaTer+round(floorX/deltaTer)+1+addedX);
    impedValue2= impedanceGrid(-
    yminTerrain/deltaTer+yflightPositions/deltaTer+round(floorY/deltaTer)+1+addedY,-
    xminTerrain/deltaTer+xflightPositions/deltaTer+round(ceilX/deltaTer)+1+addedX);
    impedValue3= impedanceGrid(-
    yminTerrain/deltaTer+yflightPositions/deltaTer+round(ceilY/deltaTer)+1+addedY,-
    xminTerrain/deltaTer+xflightPositions/deltaTer+round(ceilX/deltaTer)+1+addedX);
    impedValue4= impedanceGrid(-
    yminTerrain/deltaTer+yflightPositions/deltaTer+round(ceilY/deltaTer)+1+addedY,-
    xminTerrain/deltaTer+xflightPositions/deltaTer+round(floorX/deltaTer)+1+addedX);

    imped1= 1;
    if impedValue2==impedValue1
        imped2= 1;
        if impedValue3==impedValue2
            imped3= 1;
            if impedValue4==impedValue3
                imped4=1;
            else imped4= -1;
                otherImpedValue= impedValue4;
            end
        else imped3= -1;
            otherImpedValue= impedValue3;
            if impedValue4==impedValue1
                imped4= 1;
            elseif impedValue4==impedValue3
                imped4= -1;
            else error('Cannot have 3 different impedance values in one elementary grid square');
            end
        end
    else imped2= -1;
        otherImpedValue= impedValue2;
        if impedValue3==impedValue1
            imped3= 1;
            if impedValue4==impedValue3
                imped4= 1;
            elseif impedValue4==impedValue2
                imped4= -1;
            else error('Cannot have 3 different impedance values in one elementary grid square');
            end
        elseif impedValue3==impedValue2
            imped3= -1;
            if impedValue4==impedValue1
                imped4= 1;
            elseif impedValue4==impedValue2
                imped4= -1;
            else error('Cannot have 3 different impedance values in one elementary grid square');
            end
        end
    else error('Cannot have 3 different impedance values in one elementary grid square');
    end

    if (1-t)*(1-u)*imped1+t*(1-u)*imped2+t*u*imped3+(1-t)*u*imped4>0
        polarImpedance{thistheta}(thisrho)= impedValue1;
    else polarImpedance{thistheta}(thisrho)= otherImpedValue;
    end

    end
end

    impedIndexes= find([0 diff(polarImpedance{thistheta})]); %find the impedance transition points along the current
    azimuthal angle run
    thislengthfr= (impedIndexes-1)*deltaTer; %find the impedance transition point ranges for the current azimuthal
    angle run
    if length(thislengthfr)>2
        error('Impedance grid can have a maximum of 2 impedance discontinuities for each radial run from any source
    point');

```

```

end
impedancesAtThetas(thistheta,1)= length(thislengthfr)+1; %set the number of impedance transitions
if impedancesAtThetas(thistheta,1)==1
    ZorEFR(thistheta,1:3)= [polarImpedance{thistheta}(1) NaN(1,2)]; %set the impedance parameter values
    lengthfrAtThetas(thistheta,1:2)= NaN(1,2); %set the impedance transition point ranges
elseif impedancesAtThetas(thistheta,1)==2
    ZorEFR(thistheta,1:3)= [polarImpedance{thistheta}(1) polarImpedance{thistheta}(impedIndexes(1))
NaN(1,1)];
    lengthfrAtThetas(thistheta,1:2)= [thislengthfr(1) NaN(1,1)];
    else ZorEFR(thistheta,1:3)= [polarImpedance{thistheta}(1) polarImpedance{thistheta}(impedIndexes(1))
polarImpedance{thistheta}(impedIndexes(2))];
    lengthfrAtThetas(thistheta,1:2)= thislengthfr;
end
if strcmp(inputZorEFR,'EFR') %if the impedance parameter used is the effective flow resistivity

    [ZAtThetas(thistheta,1:3)]= DandB(freq,ZorEFR(thistheta,1:3)); %calculate and set the impedance value

elseif strcmp(inputZorEFR,'Z')
    ZAtThetas(thistheta,1:3)= ZorEFR(thistheta,1:3); %set the impedance values

else error('Please enter either "Z" or "EFR" for "Input_Z_or_EFR:" input category')

end

end

```

Plot terrain and impedance features

Following matrices are only created for the purpose of plotting in order to compare the polar coordinate system terrain profile with the given rectangular coordinate system terrain profile. They reparate the different theta value vectors into separate rows in a matrix.

```

largestRange= max(rangesAtThetas);
xPlotVals= zeros(runN,largestRange/deltar);
yPlotVals= zeros(runN,largestRange/deltar);
HPlotVals= zeros(runN,largestRange/deltar);
ImpedPlotVals= zeros(runN,largestRange/deltar);

for thistheta= 1:runN

    xPlotVals(thistheta,1:ceil(rangesAtThetas(thistheta)/deltar))= xRangeValsArray{thistheta}(1:end-1); %minus
one because propagating to 1 step less than the range
    yPlotVals(thistheta,1:ceil(rangesAtThetas(thistheta)/deltar))= yRangeValsArray{thistheta}(1:end-1); %minus
one because propagating to 1 step less than the range
    HPlotVals(thistheta,1:ceil(rangesAtThetas(thistheta)/deltar))= polarTerrain{thistheta};
    ImpedPlotVals(thistheta,1:ceil(rangesAtThetas(thistheta)/deltar))= polarImpedance{thistheta};

end

% Surface plots to compare the given rectangular terrain profile with the calculated
% polar terrain profile
figure(3); surf(xPlotVals+xflightPositions,yPlotVals+yflightPositions,HPlotVals)

xlabel('range [m]')
ylabel('range [m]')
title('Terrain')
xlim([xminTerrain+30,xmaxTerrain-30]);ylim([yminTerrain+30,ymaxTerrain-30])

```

validPEmat Function

This function returns a grid whose components equal 0 where the PE model is invalid (outside a certain angle given in degrees from a source) and 1 where the PE model is valid (inside the angle)

Inputs:

- *sourceHeight* - z-coordinate of the current source point along the flight path
- *deg* - elevation angle limit used for PE
- *deltarPlot* - range step to be saved for plotting purposes
- *deltaz* - vertical grid spacing used in the PE and FFP propagation models. Constrained to be a divisor of the receiverHeight so that the receiverHeight will fall exactly at a vertical grid point
- *range* - range for model to propagate to
- *height* - grid height (not including PE absorbing layer)

Function Process and Outputs:

1. Determine the range indices at which the PE starts being valid for each height point in the grid
2. Set the matrix element equal to 1 for all points that fall at or after the first valid point in range for each height

```
*Start Code Description:* ..... 1
Determine the range indices at which the PE starts being valid for each height point in
the grid..... 1
Set the matrix element equal to 1 for all points that fall at or after the first valid point in
range for each height..... 1
```

Start Code Description:

```
function [vPE]= validPEmat(sourceHeight,deg,deltarPlot,deltaz,range,height)

rN= floor(range/deltarPlot+1);
zN= round(height/deltaz+1);
```

Determine the range indices at which the PE starts being valid for each height point in the grid

```
index= ceil(abs(sourceHeight-[0:deltaz:height])/(deltarPlot*tan(deg*pi/180)))+1;

vPE= zeros(zN,rN);
```

Set the matrix element equal to 1 for all points that fall at or after the first valid point in range for each height

```
%These points, set to 1, are all within the valid elevation angle range
```

```
for m= 1:zN
    vPE(m,index(m):rN)= ones(1,rN-index(m)+1);
end
```

vonKarTurb Function

This function is called to return the refractive-index fluctuations μ , for a von Karman spectrum, in a matrix spanning the range and height of the vertical grid.

Inputs:

- *deltar* - horizontal grid spacing used in the PE propagation model, dependent on frequency. Also constrained to be a divisor of number chosen in 'MultiFreqQuickerRev.m'
- *range* - range for model to propagate to
- *rN* - number of range points in grid for PE
- *deltaz* - vertical grid spacing used in the PE and FFP propagation models. Constrained to be a divisor of the receiverHeight so that the receiverHeight will fall exactly at a vertical grid point
- *zN* - number of height points in grid for PE (includes absorbing layer)
- *A* - turbulence constant
- *K0* - turbulence constant related to the size of L of the largest eddies (as $2\pi/L$)
- *CTovT0* - structure parameter of the temperature fluctuations, over $T0^2$ ($CT^2/T0^2$)
- *Cvovc0* - structure parameter of the wind velocity fluctuations over $c0^2$ ($Cv^2/c0^2$)

Function Process and Outputs:

1. Define necessary parameters (number of modes, wave number spacing, etc.)
2. Define the random polar (thetan) and phase (alphan) angles and calculate the wave number components
3. Calculate the refractive-index fluctuations matrix μ based on the von Karman spectrum

Start Code Description:	1
Define necessary parameters	2
Define the random polar (thetan) and phase (alphan) angles.....	2
Calculate the wave number components.....	2
Calculate the von Karman spectrum.....	2
Initialize the refractive-index fluctuations matrix	2
Calculate the refractive-index fluctuations matrix mu.....	2

Start Code Description:

```
function [mu]= vonKarTurb(deltar,range,rN,deltaz,zN,A,K0,CTovT0,Cvovc0)
```

Define necessary parameters

As specified in E. M. Salomons "Computational Atmospheric Acoustics," (2001), Appendix J

```
N= 100; %number of modes, p.227, Salomons
deltak= .1; %wave number spacing, p.277 Salomons, [m^-1]
kn= [1:N]*deltak;
r= [0:deltak:range+deltak];
```

Define the random polar (thetan) and phase (alphan) angles

```
thetan= rand(1,N)*2*pi; %random angle between 0 and 2*pi
alphan= rand(1,N)*2*pi; %random angle between 0 and 2*pi
```

Calculate the wave number components

```
knr= kn.*cos(thetan);
knz= kn.*sin(thetan);
```

Calculate the von Karman spectrum

```
F= (A./(kn.^2+K0^2).^(8/6)).*((gamma(.5)*gamma(8/6)/gamma(11/6))*(CTovT0/4)...
+((gamma(3/2)*gamma(8/6)/gamma(17/6))+((knz.^2/(kn.^2+K0^2))*(gamma(.5)*gamma(14/6)/gamma(17/6))))*(
22*Cvovc0/12));
G= sqrt(4*pi*deltak.*F.*kn);
```

Initialize the refractive-index fluctuations matrix

```
mu= zeros(zN,rN+1); %rN+1 so that the last mu can be calculated, though not used (makes code easier)
```

Calculate the refractive-index fluctuations matrix mu

```
for k= 1:rN+1
    costermLast= exp(i*knr*r(k)+i*alphan);
    for l= 1:zN
        costermLast= costermLast.*exp(i*knz*deltaz);
        mu(l,k)= sum(G.*real(costermLast));
    end
end
```

Benchmarking the PE and FFP Models

F.1 Introduction

Extensive verification of the PE and FFP models was performed to confirm that the methods were coded correctly. This appendix includes the comparison of the PE and FFP model results to the published benchmark cases in reference [7]. It also includes a benchmarking of the included turbulence effect in the PE method with a comparison to results in reference [1]. Good agreement is found for all conditions.

F.2 Comparisons for the PE and FFP Under Different Sound Speed Profile Conditions

The benchmark cases from reference [7] explore four different atmospheric conditions. Case 1 uses a homogeneous atmosphere, Case 2 includes a linear sound speed profile with a constant gradient of 0.1 s^{-1} , Case 3 includes a linear sound speed profile with a constant gradient of -0.1 s^{-1} , and Case 4 includes a ducting sound speed profile with a linear sound speed profile with a gradient of 0.1 s^{-1} from the ground to 100 m height, a gradient of -0.1 s^{-1} from 100 m height to 300 m height, and a constant sound speed from 300 m height to infinite height

[7]. The reader is directed to the paper for a complete description of all the input conditions. Comparison with the PE results is presented first, followed by FFP comparisons, for all four cases. Three different frequencies—10 Hz, 100 Hz, and 1000 Hz—are run for each case. Figures in the first column show results for ranges between 0 m and 200 m, and figures in the second column show results out to 10 km range. Results produced by the PE and FFP models coded for this research match the results of reference [7] closely for the three different frequencies in the four different atmospheric conditions.

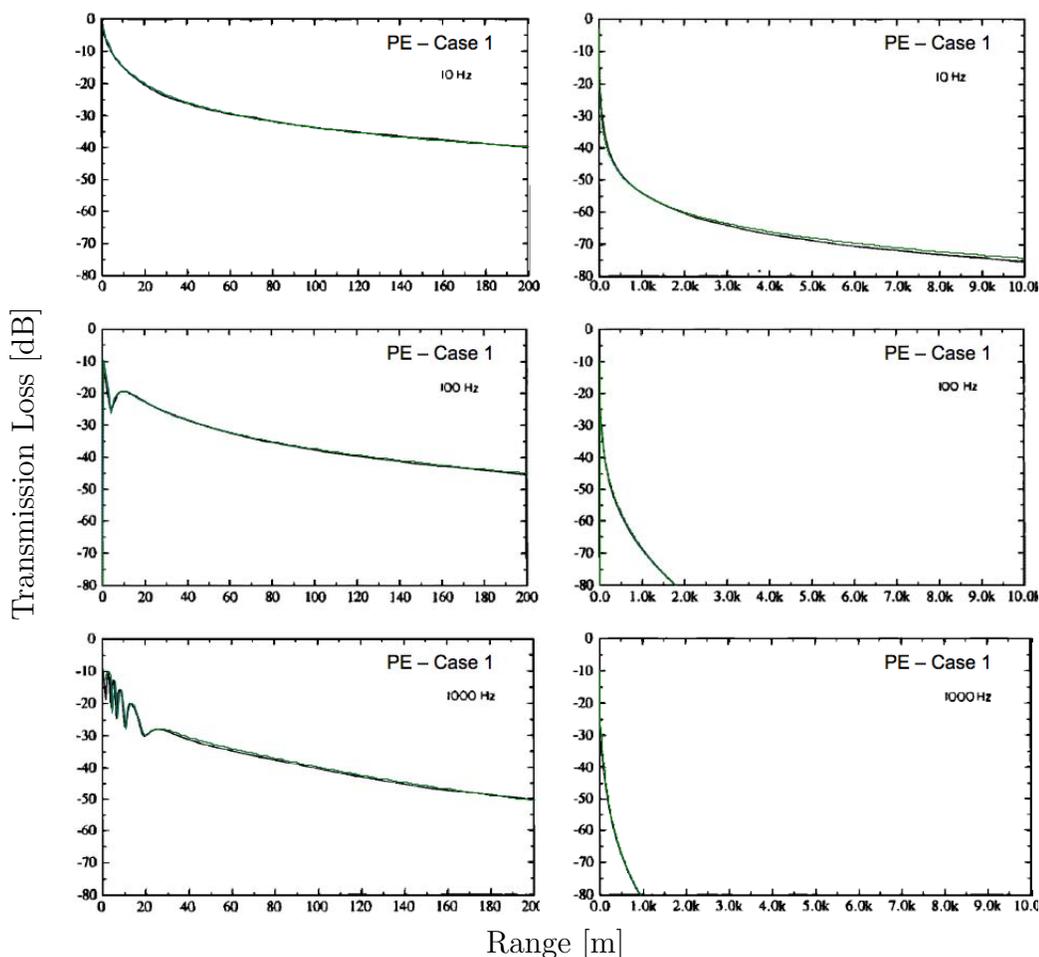


Figure F.1. Comparison of the PE module with benchmark cases from reference [7] for Case 1, a homogeneous atmosphere. The PE results of this research, shown in green, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.

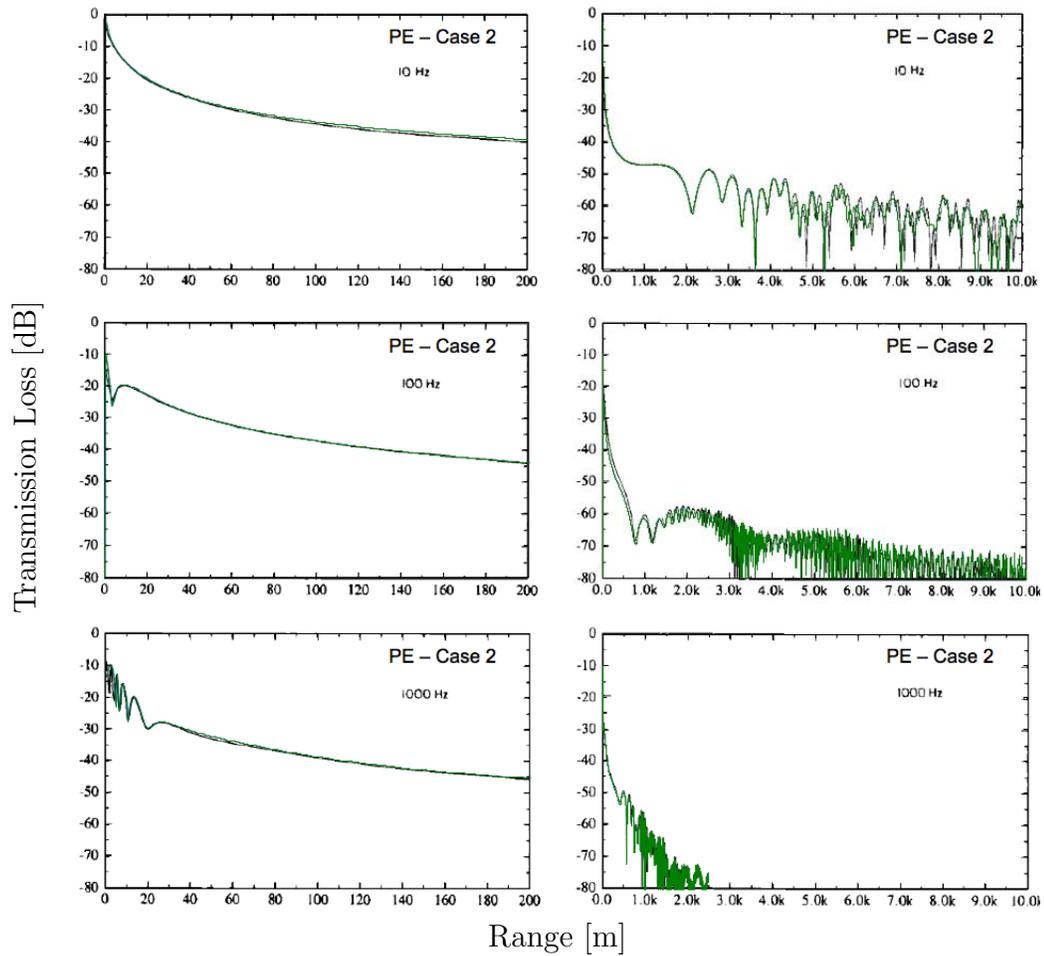


Figure F.2. Comparison of the PE module with benchmark cases from reference [7] for Case 2, a downward refracting atmosphere. The PE results of this research, shown in green, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.

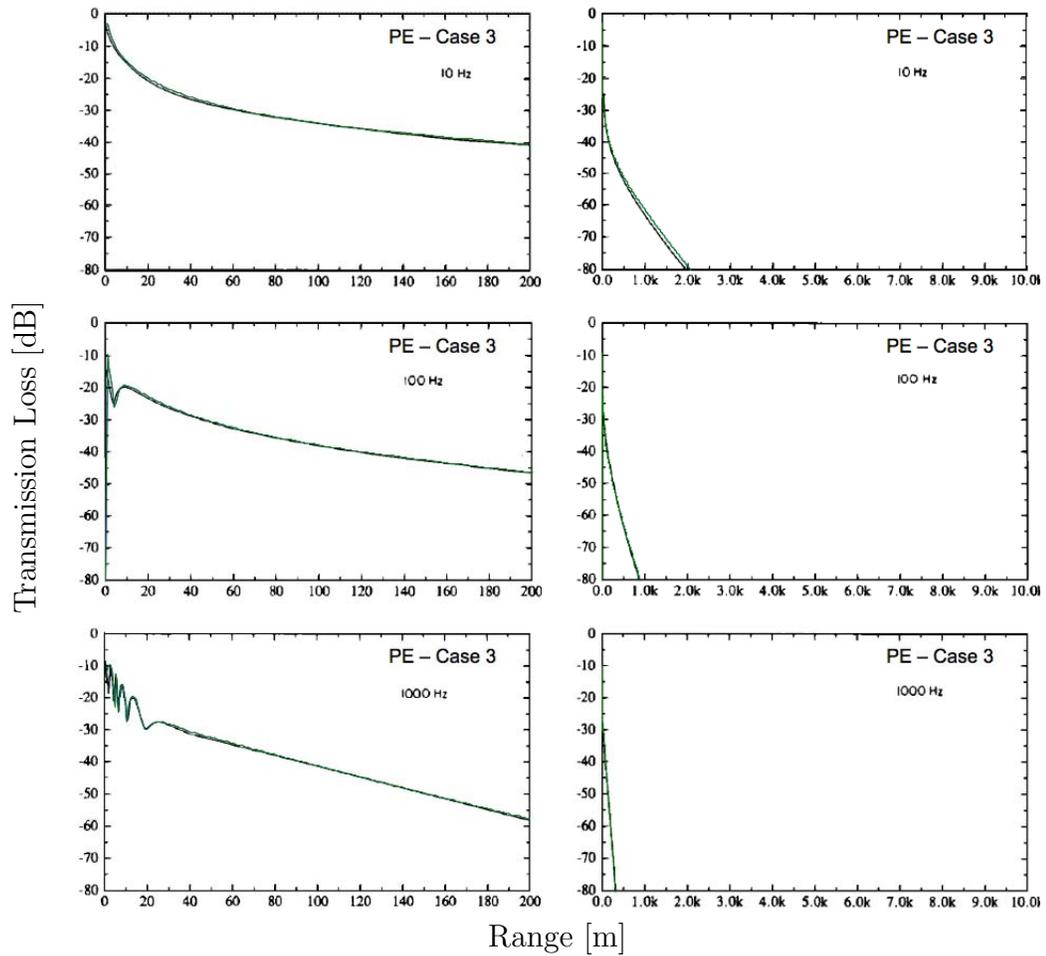


Figure F.3. Comparison of the PE module with benchmark cases from reference [7] for Case 3, an upward refracting atmosphere. The PE results of this research, shown in green, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.

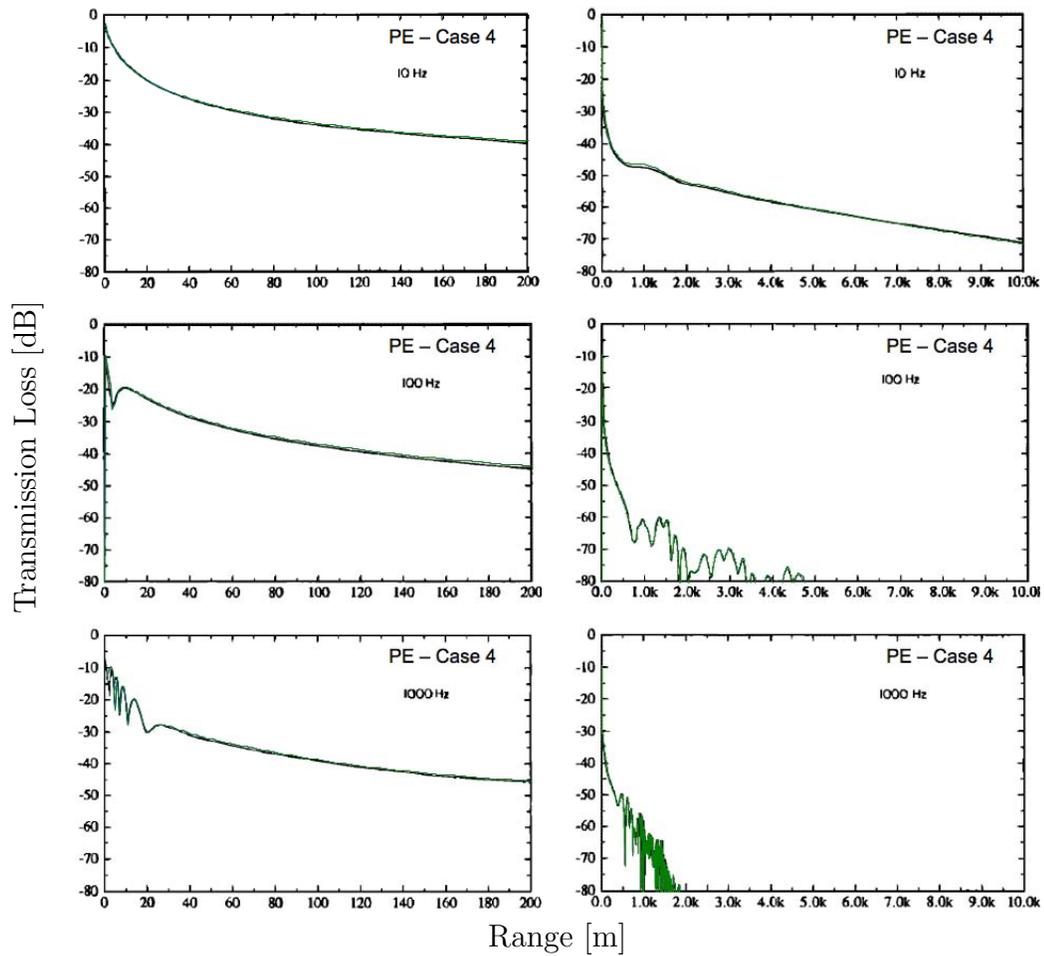


Figure F.4. Comparison of the PE module with benchmark cases from reference [7] for Case 4, a ducting atmosphere. The PE results of this research, shown in green, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.

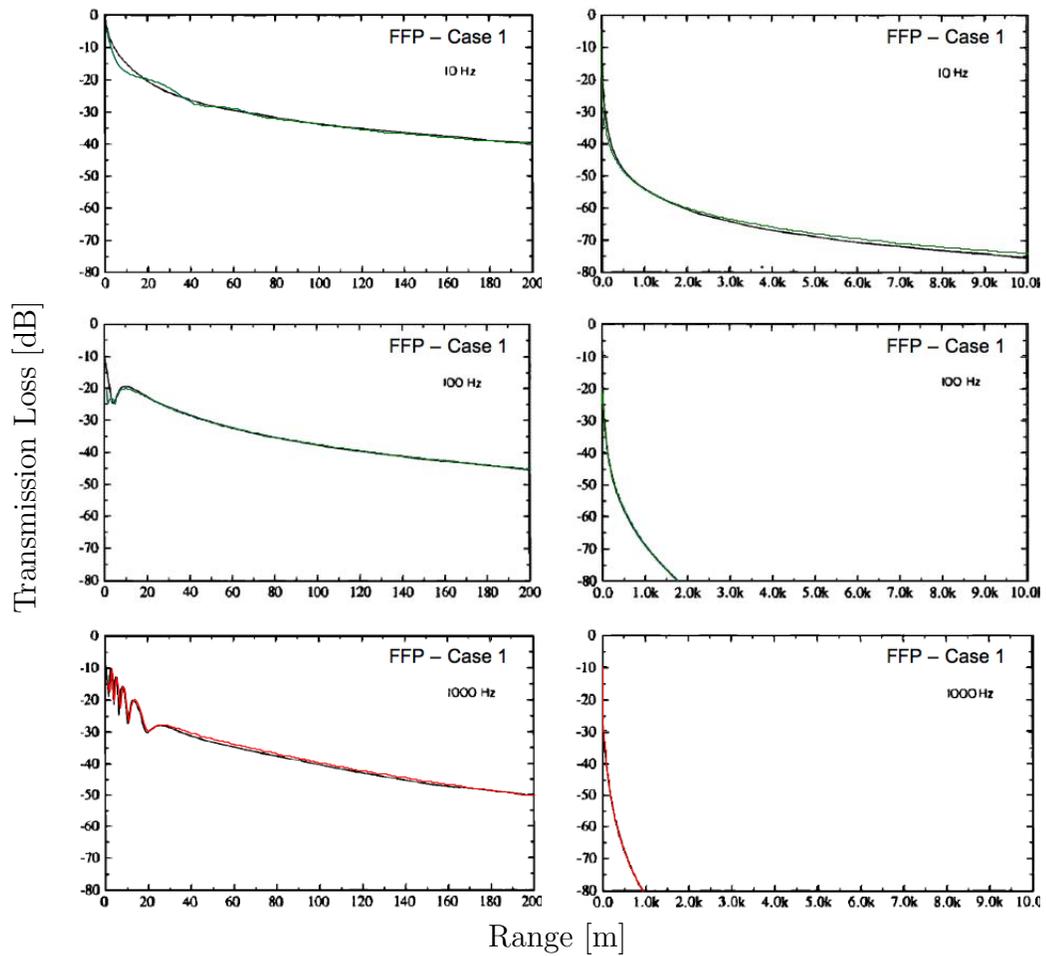


Figure F.5. Comparison of the FFP module with benchmark cases from reference [7] for Case 1, a homogeneous atmosphere. The FFP results of this research, shown in green or red, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.

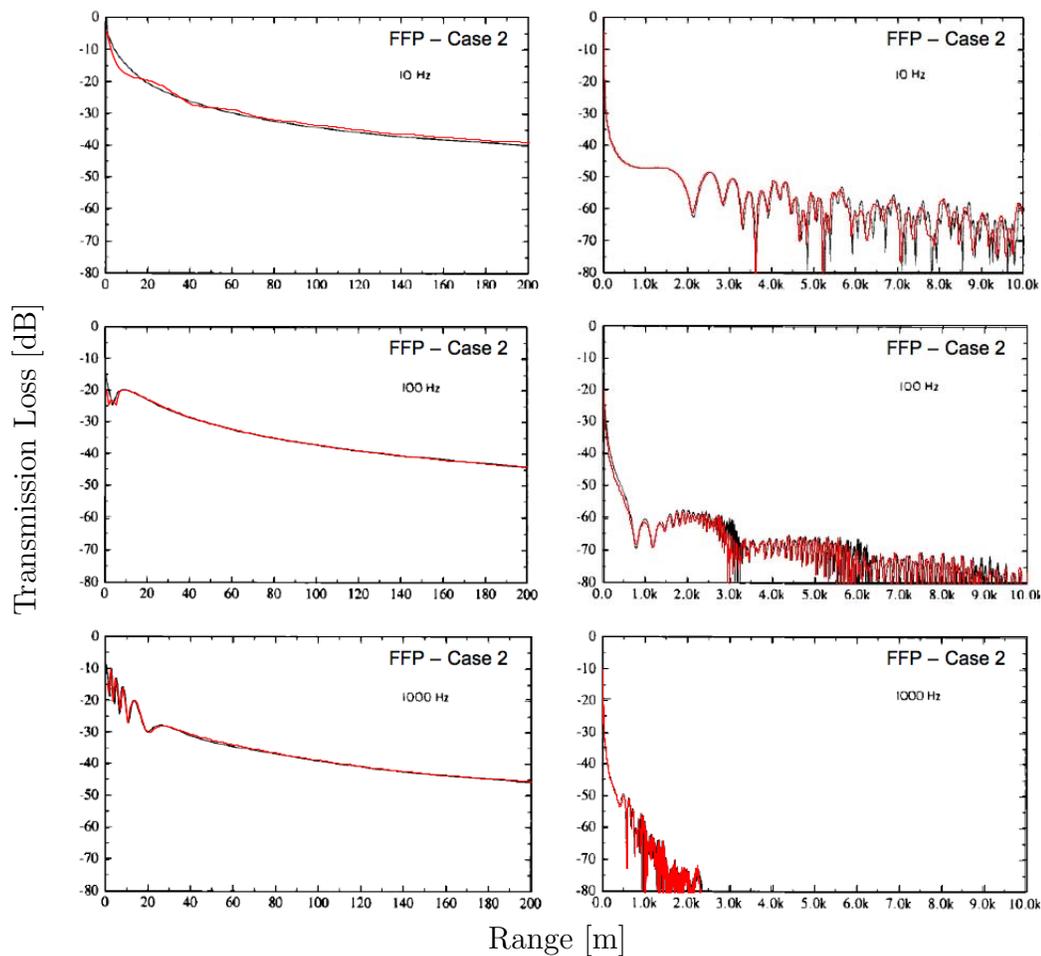


Figure F.6. Comparison of the FFP module with benchmark cases from reference [7] for Case 2, a downward refracting atmosphere. The FFP results of this research, shown in red, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.

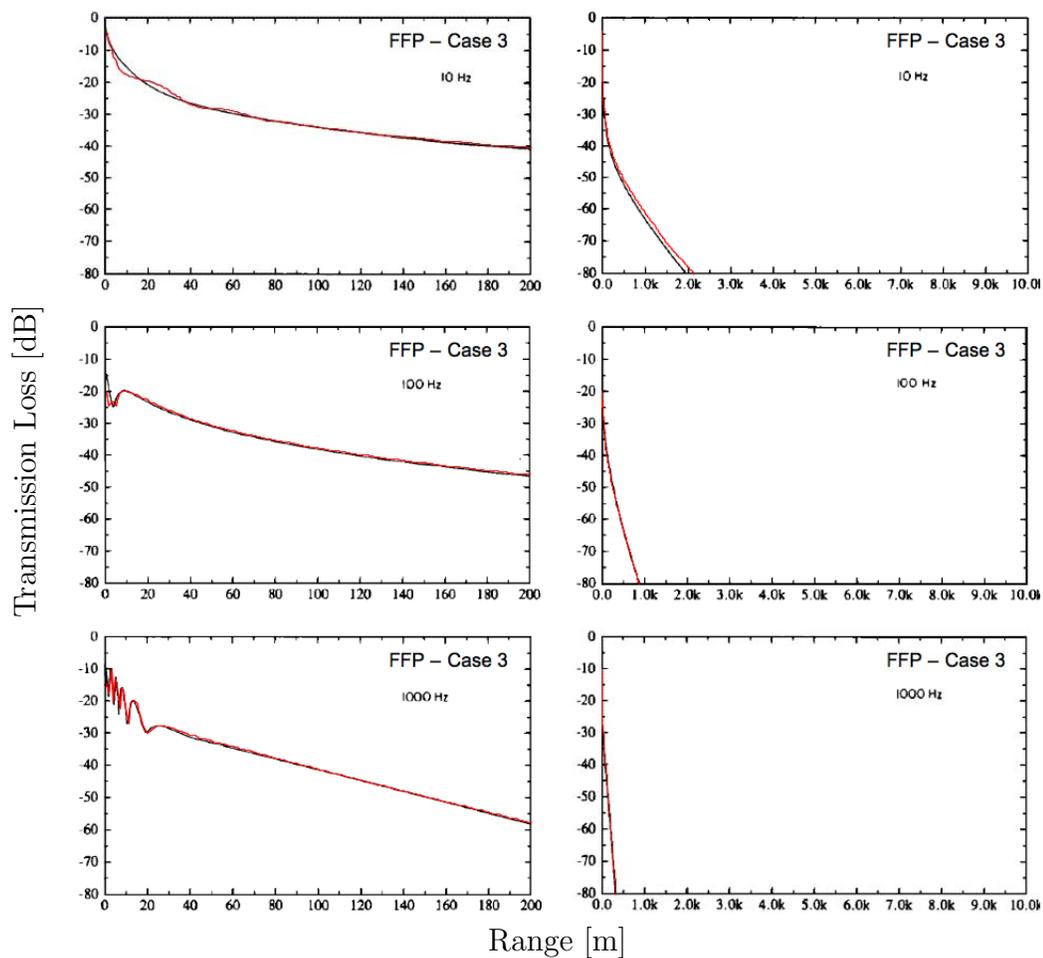


Figure F.7. Comparison of the FFP module with benchmark cases from reference [7] for Case 3, an upward refracting atmosphere. The FFP results of this research, shown in red, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.

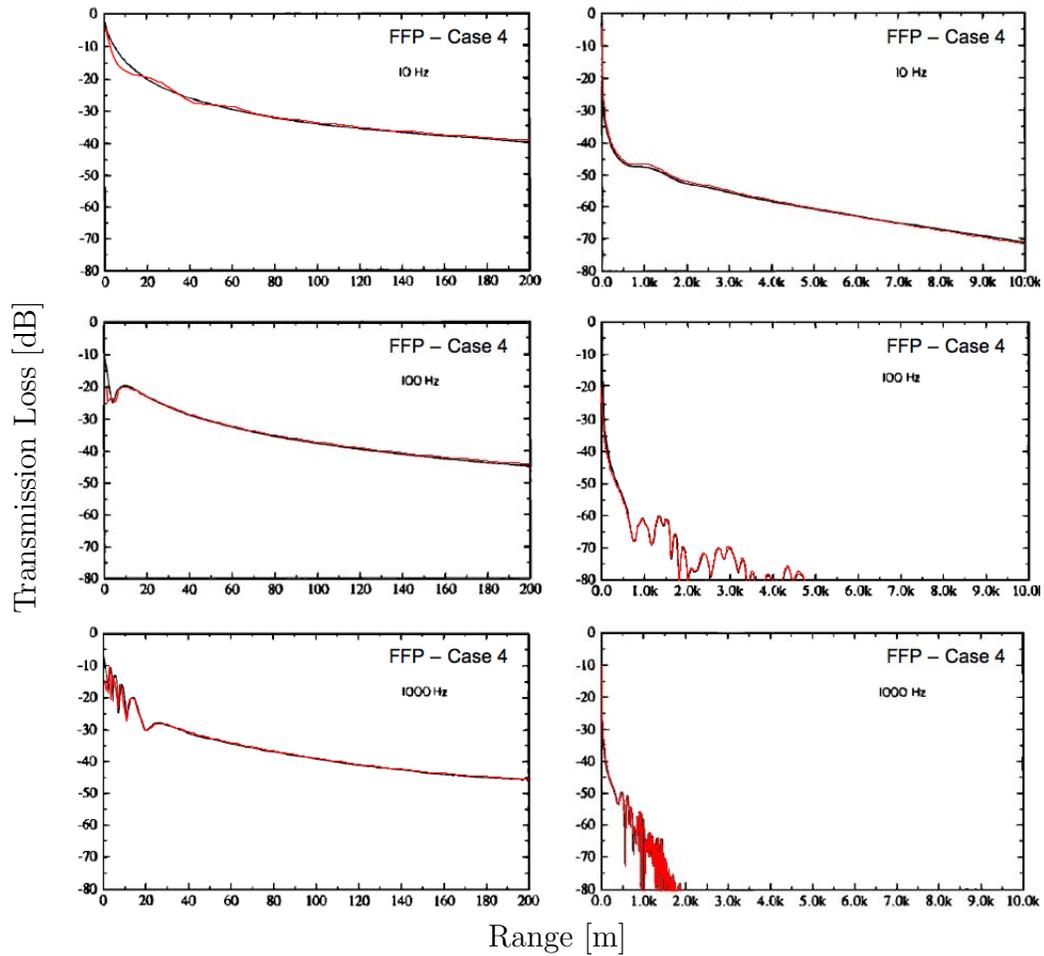


Figure F.8. Comparison of the FFP module with benchmark cases from reference [7] for Case 4, a ducting atmosphere. The FFP results of this research, shown in red, are overlaid on reference [7] results, reproduced with permission. Much of the time the lines are indistinguishable.

F.3 Comparison of PE Under Turbulent Conditions

In this section a comparison is made between results presented in reference [1] and results calculated by the PE model coded for this research including the effects of atmospheric turbulence. The von Kármán spectrum parameters used to represent the turbulence are $K_0^{-1} = 10$ m, $C_T^2/T_0^2 = 2.5 \times 10^{-7} \text{ m}^{-2/3}$, and $C_v^2/c_0^2 = 1 \times 10^{-6} \text{ m}^{-2/3}$. A source height of 2 m, receiver height of 2 m, frequency of 500 Hz, and logarithmic sound speed parameter of $b = -1$ m/s were used for propagation over a rigid ground for 50 realizations of turbulence.

The rise in sound level at larger ranges due to scattering from turbulence into the shadow zone is quite similar between the two sets of results, indicating that turbulence is included correctly in the PE model coded for this research.

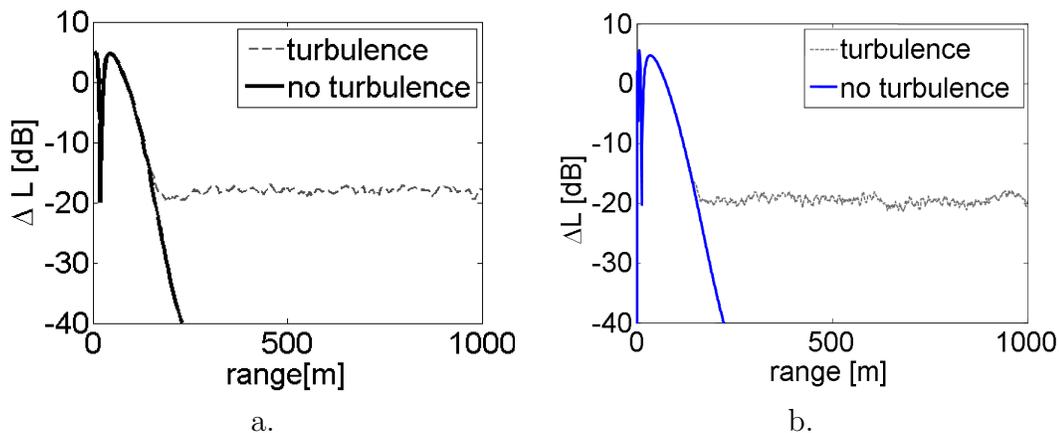


Figure F.9. Comparison of PE results of sound pressure level relative to a source in the free field including the effects of turbulence from reference [1] and the PE model used in this research. Source height = 2 m, receiver height = 2 m, frequency = 500 Hz, $b = -1$ m/s, rigid ground. Average over 50 realizations of turbulence, $K_0^{-1} = 10$ m, $C_T^2/T_0^2 = 2.5 \times 10^{-7} \text{ m}^{-2/3}$, and $C_v^2/c_0^2 = 1 \times 10^{-6} \text{ m}^{-2/3}$ (a) Results after reference [1] Figure 5.7. (b) Results of coded PE model used in this research.

Bibliography

- [1] SALOMONS, E. M. (2001) *Computational Atmospheric Acoustics*, Kluwer Academic Publishers.
- [2] LAMANCUSA, J. S. (2009) “10. Outdoor Sound Propagation,” in *ME 458: Engineering Noise Control Notes*, The Pennsylvania State University.
- [3] EMBLETON, T. F. W. (1996) “Tutorial on sound propagation outdoors,” *The Journal of the Acoustical Society of America*, **100**(1), pp. 31–48.
- [4] BOEKER, E. R., E. DINGES, B. HE, G. G. FLEMING, C. J. ROOF, P. J. GERBI, A. S. RAPOZA, and J. HEMANN (2008) *Integrated Noise Model (INM) Version 7.0 Technical Manual, Tech. rep.*, Report FAA-AEE-08-01, Office of Environment and Energy, Federal Aviation Administration, Washington, D.C.
- [5] SALOMONS, E. M., F. H. A. VAN DEN BERG, and H. E. A. BRACKENHOFF (1994) “Long-term average sound transfer through the atmosphere: predictions based on meteorological statistics and numerical computations of sound propagation,” in *Proceedings of the 6th International Symposium on Long Range Sound Propagation, Ottawa, Canada*, pp. 209–228.
- [6] SALOMONS, E. M. and F. H. A. VAN DEN BERG (1999) “Interpolation model for long-range propagation of shooting noise over heterogeneous ground,” in *Inter-Noise Conference Proceedings*, vol. 106, INCE, pp. 393–398.
- [7] ATTENBOROUGH, K., S. TAHERZADEH, H. E. BASS, X. DI, R. RASPET, G. R. BECKER, A. GÜDESEN, A. CHRESTMAN, G. A. DAIGLE, A. L’ESPÉRANCE, ET AL. (1995) “Benchmark cases for outdoor sound propagation models,” *The Journal of the Acoustical Society of America*, **97**(1), pp. 173–191.
- [8] EMBLETON, T. F. W., J. E. PIERCY, and G. A. DAIGLE (1983) “Effective flow resistivity of ground surfaces determined by acoustical measurements,” *The Journal of the Acoustical Society of America*, **74**(4), pp. 1239–1244.

- [9] SOCIETY OF AUTOMOTIVE ENGINEERS, COMMITTEE A-21, AIRCRAFT NOISE (1975) *Standard Values of Atmospheric Absorption as a Function of Temperature and Humidity*, Tech. rep., Aerospace Research Report No. 866A.
- [10] ANSI S1.26-1995 (R2009) (2009) *American National Standard Method for Calculation of the Absorption of Sound by the Atmosphere*, Tech. rep., ANSI.
- [11] FEDERAL AVIATION ADMINISTRATION (2010) “Forecast Links Aviation Activity and National Economic Growth,” *Washington Headquarters Press Release*.
URL http://www.faa.gov/news/press_releases/news_story.cfm?newsId=11232
- [12] BOEKER, E. R. (2008) *Meteorological and Environmental Effects on INM Sound Propagation*, Memo, USDOT Research & Innovative Administration, Environmental Measurement and Modeling Division, Volpe Center Acoustics Facility.
- [13] PAGE, J. A., C. WILMER, T. SCHULTZ, K. J. PLOTKIN, and J. CZECH (2008) *Advanced Acoustic Model Technical Reference And User Manual*, Tech. Rep. WR 08-12, Wyle Laboratories, Inc.
- [14] ATTENBOROUGH, K. (2002) “Sound propagation close to the ground,” *Annual Review of Fluid Mechanics*, **34**(1), pp. 51–82.
- [15] PIERCY, J. E., T. F. W. EMBLETON, and L. C. SUTHERLAND (1977) “Review of noise propagation in the atmosphere,” *The Journal of the Acoustical Society of America*, **61**(6), pp. 1403–1418.
- [16] BASS, H. E., L. C. SUTHERLAND, A. J. ZUCKERWAR, D. T. BLACKSTOCK, and D. M. HESTER (1995) “Atmospheric absorption of sound: Further developments,” *The Journal of the Acoustical Society of America*, **97**(1), pp. 680–683.
- [17] SOCIETY OF AUTOMOTIVE ENGINEERS, COMMITTEE A-21, AIRCRAFT NOISE (1986) *Procedure for the Computation of Airplane Noise in the Vicinity of Airports*, Tech. rep., Aerospace Information Report No. 1845.
- [18] INTERNATIONAL STANDARD ORGANISATION (1993), “ISO 9613-1. Attenuation of sound during propagation outdoors. Part 1: Calculation of the absorption of sound by the atmosphere.” .
- [19] DELANY, M. E. and E. N. BAZLEY (1970) “Acoustical properties of fibrous absorbent materials,” *Applied Acoustics*, **3**(2), pp. 105–116.

- [20] CHESSELL, C. I. (1977) "Propagation of noise along a finite impedance boundary," *The Journal of the Acoustical Society of America*, **62**(4), pp. 825–834.
- [21] ATTENBOROUGH, K. (1985) "Acoustical impedance models for outdoor ground surfaces," *Journal of Sound and Vibration*, **99**(4), pp. 521–544.
- [22] DAIGLE, G. A., J. NICOLAS, and J.-L. BERRY (1985) "Propagation of noise above ground having an impedance discontinuity," *The Journal of the Acoustical Society of America*, **77**(1), pp. 127–137.
- [23] BOULANGER, P., T. WATERS-FULLER, K. ATTENBOROUGH, and K. M. LI (1997) "Models and measurements of sound propagation from a point source over mixed impedance ground," *The Journal of the Acoustical Society of America*, **102**(3), pp. 1432–1442.
- [24] DE JONG, B. A., A. MOERKERKEN, and J. D. VAN DER TOORN (1983) "Propagation of sound over grassland and over an earth barrier," *Journal of Sound and Vibration*, **86**(1), pp. 23–46.
- [25] KOERS, P. (1983) "Diffraction by an absorbing barrier or by an impedance transition," in *Proceedings of Inter-Noise*, vol. 83, pp. 311–314.
- [26] RASMUSSEN, K. B. (1982) "A note on the calculation of sound propagation over impedance jumps and screens," *Journal of Sound and Vibration*, **84**, pp. 598–602.
- [27] ATTENBOROUGH, K., K. M. LI, and K. HOROSHENKOV (2007) *Predicting outdoor sound*, Taylor & Francis, London and New York.
- [28] CRADDOCK, J. N. and M. J. WHITE (1992) "Sound propagation over a surface with varying impedance: A Parabolic Equation approach," *The Journal of the Acoustical Society of America*, **91**(6), pp. 3184–3191.
- [29] SCHULTZ, T. (2007) *Aircraft Noise Simulation*, M. Eng., The Pennsylvania State University.
- [30] LIDTHILL, M. J. (1952) "On sound generated aerodynamically. I. General theory," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, **211**(1107), pp. 564–587.
- [31] RUIJGROK, G. J. J. (2007) *Elements of aviation acoustics*, 2 ed., VSSD, Leegwaterstraat 42, 2628 CA Delft, The Netherlands.
- [32] LI, K. M., S. TAHERZADEH, and K. ATTENBOROUGH (1997) "Sound propagation from a dipole source near an impedance plane," *The Journal of the Acoustical Society of America*, **101**(6), pp. 3343–3352.

- [33] LI, K. M. and S. TAHERZADEH (1997) “The sound field of an arbitrarily oriented quadrupole near ground surfaces,” *The Journal of the Acoustical Society of America*, **102**(4), pp. 2050–2057.
- [34] FLEMING, G. G., D. A. SENZIG, and J. P. B. CLARKE (2002) “Lateral attenuation of aircraft sound levels over an acoustically hard water surface: Logan airport study,” *Noise Control Engineering Journal*, **50**(1), pp. 19–29.
- [35] SPARROW, V. W. (2006) “Outdoor Sound Propagation Class Notes,” The Pennsylvania State University’s Graduate Program in Acoustics.
- [36] KANKEY, A. T. (2008) *A method for focusing sound in harbor environments at low frequencies: Theory and experiment*, Ph.D. thesis, The Pennsylvania State University.
- [37] GILBERT, K. E. and X. DI (1993) “A fast Green’s function method for one-way sound propagation in the atmosphere,” *The Journal of the Acoustical Society of America*, **94**(4), pp. 2343–2352.
- [38] LEVY, M. (2000) *Parabolic equation methods for electromagnetic wave propagation*, The Institution of Electrical Engineers, London.
- [39] WEST, M., K. GILBERT, and R. A. SACK (1992) “A tutorial on the parabolic equation (PE) model used for long range sound propagation in the atmosphere,” *Applied Acoustics*, **37**(1), pp. 31–49.
- [40] ROBERTSON, J. S., P. J. SCHLATTER, and W. L. SIEGMANN (1996) “Sound propagation over impedance discontinuities with the parabolic approximation,” *The Journal of the Acoustical Society of America*, **99**(2), pp. 761–767.
- [41] GILBERT, K. E. and M. J. WHITE (1989) “Application of the parabolic equation to sound propagation in a refracting atmosphere,” *The Journal of the Acoustical Society of America*, **85**(2), pp. 630–637.
- [42] CHEVRET, P., P. BLANC-BENON, and D. JUVÉ (1996) “A numerical model for sound propagation through a turbulent atmosphere near the ground,” *The Journal of the Acoustical Society of America*, **100**(6), pp. 3587–3599.
- [43] OSTASHEV, V. E., E. M. SALOMONS, S. F. CLIFFORD, R. J. LATAITIS, D. K. WILSON, P. BLANC-BENON, and D. JUVÉ (2001) “Sound propagation in a turbulent atmosphere near the ground: A parabolic equation approach,” *The Journal of the Acoustical Society of America*, **109**(5), pp. 1894–1908.
- [44] SACK, R. A. and M. WEST (1995) “A parabolic equation for sound propagation in two dimensions over any smooth terrain profile: the generalised terrain parabolic equation (GT-PE),” *Applied Acoustics*, **45**(2), pp. 113–129.

- [45] ROBERTSON, J. S., W. L. SIEGMANN, and M. J. JACOBSON (1995) “Low-frequency sound propagation modeling over a locally reacting boundary with the parabolic approximation,” *The Journal of the Acoustical Society of America*, **98**(2), pp. 1130–1137.
- [46] LEONTOVICH, M. A. and V. A. FOCK (1946) “Solution of the problem of propagation of electromagnetic waves along the earth’s surface by the method of parabolic equation,” *J. Phys. USSR*, **10**(1), pp. 13–23.
- [47] FOCK, V. A. (1965) *Electromagnetic diffraction and propagation problems*, Pergamon Press, New York.
- [48] TAPPERT, F. D. (1977) “The parabolic approximation method,” *Wave propagation and underwater acoustics, Lecture Notes in Physics*, **70**, pp. 224–287.
- [49] CLAERBOUT, J. F. (1985) *Fundamentals of geophysical data processing*, Blackwell, Oxford.
- [50] WHITE, M. J. and K. E. GILBERT (1989) “Application of the parabolic equation to the outdoor propagation of sound,” *Applied Acoustics*, **27**(3), pp. 227–238.
- [51] LEE, D., A. D. PIERCE, and E. C. SHANG (2000) “Parabolic equation development in the twentieth century,” *Journal of Computational Acoustics*, **8**(4), pp. 527–637.
- [52] GREENE, R. R. (1984) “The rational approximation to the acoustic wave equation with bottom interaction,” *The Journal of the Acoustical Society of America*, **76**(6), pp. 1764–1773.
- [53] COOPER, J. L. and D. C. SWANSON (2007) “Parameter selection in the Green’s function parabolic equation,” *Applied Acoustics*, **68**(4), pp. 390–402.
- [54] ABRAMOWITZ, M. and I. A. STEGUN (1972) *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, U.S. Department of Commerce.
- [55] WILSON, D. K. (1992) *Acoustic tomographic monitoring of the atmospheric boundary layer*, Ph.D. thesis, The Pennsylvania State University, University Park, PA.
- [56] FRANKE, S. J. and G. W. SWENSON, JR. (1989) “Brief tutorial on the Fast Field Program (FFP) as applied to sound propagation in the air,” *Applied Acoustics*, **27**(3), pp. 203–215.

- [57] WEST, M., R. A. SACK, and F. WALKDEN (1991) “The Fast Field Program (FFP). A second tutorial: Application to long range sound propagation in the atmosphere,” *Applied Acoustics*, **33**(3), pp. 199–228.
- [58] KUTSCHALE, H. W. (1970) *The integral solution of the sound field in a multilayered liquid-solid half-space with numerical computations for low-frequency propagation in the Arctic Ocean, Tech. Rep. 1 (CU-1-70, ONR Contract N00014-67-A-0108-0016)*, Lamont-Doherty Geological Observatory, Columbia University, Palisades, New York.
- [59] DINAPOLI, F. R. (1971) *Fast field program for multilayered media, Tech. Rep. 4103*, Naval Underwater Systems Center.
- [60] KUTSCHALE, H. W. (1973) *Rapid computation by wave theory of propagation loss in the Arctic Ocean, Tech. Rep. 8*, Lamont-Doherty Geological Observatory, Columbia University, Palisades, New York.
- [61] RASPET, R., S. W. LEE, E. KUESTER, D. C. CHANG, W. F. RICHARDS, R. GILBERT, and N. BONG (1985) “A fast-field program for sound propagation in a layered atmosphere above an impedance ground,” *The Journal of the Acoustical Society of America*, **77**(2), pp. 345–352.
- [62] LEE, S. W., N. BONG, W. F. RICHARDS, and R. RASPET (1986) “Impedance formulation of the fast field program for acoustic wave propagation in the atmosphere,” *The Journal of the Acoustical Society of America*, **79**(3), pp. 628–634.
- [63] NIJS, L. and C. P. A. WAPENAAR (1990) “The influence of wind and temperature gradients on sound propagation, calculated with the two-way wave equation,” *The Journal of the Acoustical Society of America*, **87**(5), pp. 1987–1998.
- [64] WILSON, D. K. (1993) “Sound field computations in a stratified, moving medium,” *The Journal of the Acoustical Society of America*, **94**(1), pp. 400–407.
- [65] LI, Y. L., S. J. FRANKE, and C. H. LIU (1991) “Numerical implementation of an adaptive fast-field program for sound propagation in layered media using the chirp z transform,” *The Journal of the Acoustical Society of America*, **89**(5), pp. 2068–2075.
- [66] RASPET, R. and W. WU (1995) “Calculation of average turbulence effects on sound propagation based on the fast field program formulation,” *The Journal of the Acoustical Society of America*, **97**(1), pp. 147–153.

- [67] IKELHEIMER, B. and K. J. PLOTKIN (2005) *Noise Model Simulation (NM-Sim) User's Manual, Tech. Rep. WR 03-09*, Wyle Laboratories, 2001 Jefferson Davis Highway Suite 701.
- [68] ROSENBAUM, J. E., A. A. ATCHLEY, and V. W. SPARROW (2009) "Enhanced sound propagation modeling of aviation noise using a hybrid Parabolic Equation-Fast Field Program method," in *Inter-Noise Conference Proceedings*, INCE, Ottawa, ON.
- [69] ROSENBAUM, J. E., A. A. ATCHLEY, AND V. W. SPARROW (2010) "Modeling propagation of aviation noise under range-dependent conditions, using a Parabolic Equation method," in *Noise-Con Conference Proceedings*, Baltimore, MD.
- [70] DOCKERY, G. D. (1988) "Modeling electromagnetic wave propagation in the troposphere using the parabolic equation," *IEEE Transactions on Antennas and Propagation*, **36**(10), pp. 1464–1470.
- [71] WILLIAMS, E. G. (1999) *Fourier acoustics: sound radiation and nearfield acoustical holography*, Academic Press, San Diego.
- [72] AF CENTER FOR ENGINEERING AND THE ENVIRONMENT. US AIR FORCE., "Aircraft Noise Related Software," .
- [73] CZECH, J. J. and K. J. PLOTKIN (1998) *NMAP 7.0 User's Manual, Tech. Rep. Report No. AFRL-HE-WP-TR-2002-0132*, Wyle Laboratories, Arlington, VA.
- [74] EUROCONTROL, "The Aircraft Noise and Performance (ANP) database," . URL www.aircraftnoisemodel.org
- [75] HEIMANN, D. and E. M. SALOMONS (2004) "Testing meteorological classifications for the prediction of long-term average sound levels," *Applied Acoustics*, **65**(10), pp. 925–950.
- [76] SALOMONS, E. M. and D. HEIMANN (2004) *Description of the Reference model: Deliverable 16 of the Harmonoise project, Work Package HAR29TR-041118-TNO10*, Harmonoise.
- [77] VAN DEN BERG, F. H. A., N. A. KINNEGING, and E. M. SALOMONS (1996) "An overview of a method to predict average propagation of shooting noise in order to create computer-generated noise contours around shooting ranges," in *Inter-Noise Conference Proceedings*, vol. 217, INCE, pp. 579–582.
- [78] VAN DEN BERG, F. H. A. (2009) "Email communication," .

- [79] CIMORELLI, A. J., S. G. PERRY, A. VENKATRAM, J. C. WEIL, R. J. PAINE, R. B. WILSON, R. F. LEE, W. D. PETERS, and R. W. BRODE (2005) “AERMOD: a dispersion model for industrial source applications. Part I: general model formulation and boundary layer characterization,” *Journal of Applied Meteorology*, **44**(5), pp. 682–693.
- [80] WILSON, D. K., J. G. BRASSEUR, and K. E. GILBERT (1999) “Acoustic scattering and the spectrum of atmospheric turbulence,” *The Journal of the Acoustical Society of America*, **105**(1), pp. 30–34.
- [81] DEWAN, E. M. and N. GROSSBARD (2007) “The inertial range ‘outer scale’ and optical turbulence,” *Environmental Fluid Mechanics*, **7**(5), pp. 383–396.
- [82] CUIJPERS, J. W. M. and W. KOHSIEK (1989) “Vertical profiles of the structure parameter of temperature in the stable, nocturnal boundary layer,” *Boundary-Layer Meteorology*, **47**(1), pp. 111–129.

Vita

Joyce E. Rosenbaum

Education

Ph.D. in Acoustics, The Pennsylvania State University, May 2011

B.A. in Physics, Cornell University, May 2006

Experience

Research Assistant, Propagation Modeling of Aviation Noise, The Pennsylvania State University, University Park, PA, 2007-present

Research Assistant, Content Development for *NoiseQuest* outreach website, The Pennsylvania State University, University Park, PA, 2006-2007

Research Assistant, Investigation of preferred acoustic conditions for libraries, Cornell University, Ithaca, NY, 2005-2006

Intern, Sound and Vibration Research Experience for Undergraduates (REU), Colorado State University, Fort Collins, CO, June-July 2005

Papers, Presentations, and Awards

Atlantic City Area Amelia Earhart Fellowship Award winner (2010): Zonta International's Atlantic City Area Club in coordination with the Federal Aviation Administration Air Transportation Centers of Excellence Program Office

Kenneth T. Simowitz Memorial Citation winner (2010): The Pennsylvania State University Graduate Program in Acoustics

Noise-Con Conference Proceedings Paper and Presentation (2010): J.E. Rosenbaum, A. A. Atchley, V. W. Sparrow, Modeling propagation of aviation noise under range-dependent conditions, using a Parabolic Equation method, Baltimore, MD, April 2010.

Inter-Noise Conference Proceedings Paper and Presentation (2009): J.E. Rosenbaum, A. A. Atchley, V. W. Sparrow, Enhanced sound propagation modeling of aviation noise using a hybrid Parabolic Equation-Fast Field Program method, Ottawa, ON, August 2009.

Invited Speaker at American Library Association Annual Conference (2007): J. E. Rosenbaum, C. L. Krumhansl, and M. J. Hove, Fine-tuning the acoustics of libraries at colleges and universities, Washington, D.C., June 2007.

Invited Speaker at 151st Meeting of Acoustical Society of America (2006): J. E. Rosenbaum, C. L. Krumhansl, and M. J. Hove, Fine-tuning the acoustics of libraries at colleges and universities, Providence, RI, J. Acoust. Soc. Am. 119(5), 3401, May 2006.