Outdoor Sound Propagation and the Boundary Element Method

Philip Mc Laughlin

August 21, 2005

Abstract

This project is an extension of the work done previously on the Application of the Boundary Element Method to the design Traffic Noise Barriers. A computer program was written by Chandler-Wilde and D.C. Hothersall in 1995 using Fortran 77, which can calculate numerical solutions, using the Boundary Element Method, to problems of propagation from a line source over one or more noise barriers sitting on a homogeneous flat ground. Much of the work for this project involved rewriting and improving this code using Matlab.

Contents

List of Figures

Declaration

I confirm that this work is my own and the use of all other material from other sources has been properly and fully acknowledged.

Acknowledgments

I would particularly like to thank Professor Simon Chandler-Wilde for his support, encouragement and good humour throughout this project. I would like to acknowledge the other members of the staff in the Mathematics Department for their assistance throughout the year.

I would also like to acknowledge the EPSRC for financially supporting this work.

Finally I would like to thank my friends and family for their support over this time.

Chapter 1

Introduction

With an ever increasing amount of congestion on the roads, the rails, and in the skies, traffic noise is a common problem that most people have experienced at one time or another. Road traffic is by far the most widespread problem and is there is ongoing research in the field of reducing the levels of noise pollution from our roads. There are a number of ways of reducing road traffic noise levels, these include reducing traffic levels, improvements to both road and tyre surface designs, sound proofing of buildings, and the use of traffic noise barriers. One of the simplest and most cost effective solutions is the use of noise barriers, or the improvement of existing noise barriers.

In order to investigate which barrier configurations, or barrier improvements are effective we can employ a numerical model. The boundary element method(BEM) gives a numerical solution to the problem by obtaining accurate solutions to the Helmholtz wave equation. The Helmholtz wave equation governs propagation, reflection and scattering of acoustic waves in a homogeneous atmosphere. The BEM is particularly well suited to this problem as it can deal with barriers of arbitrary shape and surface covering.

In this project we will firstly look at introducing some background acoustical knowledge needed in order to appreciate the problem of modelling outdoor sound in order to investigate traffic noise barriers.

In chapter two the two dimensional BEM used in this project is presented. Then a discussion of work which has been done using the BEM to investigate traffic noise barriers is discussed. Then we describe how the BVP is formulated, how it is converted to an integral equation, and how it is approximated and numerically solved.

In chapter three some of the advantages of programming in MATLAB and the difficulties incurred in re-programming are discussed. The program is described and main functions are outlined in detail. Then some improvements to program are discussed.

Finally the results using the new Matlab BEM code are presented and discussed.

1.1 Background Acoustics

The perturbation in pressure, $P(\mathbf{r}, t)$, is a function of position, $\mathbf{r}$, and time, t, and it satisfies the following homogeneous wave equation

$$\nabla^2 P - \frac{1}{c^2}\frac{\partial^2 P}{\partial t^2} = 0 \tag{1.1}$$

where $c \approx 340m/s$.

The density perturbation, ρ, also satisfies the same wave equation. Provided the wave motion is initially irrotational, the velocity, v, is the gradient of a scalar field, Φ, which is the velocity potential, which also satisfies the wave equation. Φ,P, and ρ are related by the following three equations:

$$\mathbf{v} = \nabla\Phi, \qquad P = -\rho_0\frac{\partial\Phi}{\partial t}, \qquad P = c^2\rho. \tag{1.2}$$

If we consider a mono frequency acoustic wave, with angular frequency $\omega > 0$, the frequency given by

$$f = \omega/(2\pi), \tag{1.3}$$

then the pressure and velocity potential are given by the following two equations respectively:

$$P(\mathbf{r}, t) = Re(p(\mathbf{r})e^{-i\omega t}) \tag{1.4}$$

$$\Phi(\mathbf{r}, t) = Re(\phi(\mathbf{r})e^{-i\omega t}) \tag{1.5}$$

where p and ϕ are functions of of position alone. p is a known as the acoustic pressure, and ϕ is known as the acoustic potential.

The acoustic potential, ϕ, is related to the acoustic pressure, p, by the following equation

$$p = i\omega\rho_0\phi \tag{1.6}$$

The wave equation 1.1 and the resulting relations 1.2 are satisfied if p satisfies the Helmholtz equation

$$(\nabla^2 + k^2)p = 0 \tag{1.7}$$

where k is known as the wavenumber and is given by

$$k = \omega/c. \tag{1.8}$$

If the fluid is bounded by a rigid obstacle, an appropriate boundary condition is

$$\frac{\partial\phi}{\partial n} = 0 \tag{1.9}$$

where $\partial\phi/\partial n$ denote the rate of change in the direction of the normal to the barrier, which is directed out of the fluid and into the surface of the barrier. More generally, the normal velocity is non-zero, defined by

$$Z_s = \frac{p}{\frac{\partial\phi}{\partial n}} \tag{1.10}$$

where Z_s is called the specific surface impedance. In general Z_s depends on the variation of the acoustic field throughout the medium of propagation. However the ratio $\frac{\partial \phi}{\partial n}$ is often a constant in which case the boundary is called locally reacting.

Applying equation (1.6), we can rewrite (1.10) as

$$\frac{\partial \phi}{\partial n} = ik\beta\phi \qquad or \qquad \frac{\partial p}{\partial n} = ik\beta p \tag{1.11}$$

where

$$\beta = \frac{\rho_s c}{Z_s}. \tag{1.12}$$

When the medium of propagation is bounded, the Helmholtz equation and impedance boundary condition ensure a unique solution [24]. If the fluid extends to infinity an additional boundary condition is necessary to ensure uniqueness, the two dimensional Sommerfeld radiation condition given by,

$$\frac{\partial p}{\partial r} - ikp = o(r^{-1/2}) \tag{1.13}$$

$$p = O(r^{-1/2})$$

uniformly as $r \to \infty$, [25].

One solution of the Helmholtz equation (1.7) is

$$p(\mathbf{r}) = G_f(\mathbf{r}, \mathbf{r}_0) = -\frac{e^{ik|\mathbf{r}-\mathbf{r}_0|}}{4\pi|\mathbf{r}-\mathbf{r}_0|}, \tag{1.14}$$

which satisfies 1.7 at every point $\mathbf{r} \neq \mathbf{r}_0$. This solution represents the acoustic pressure generated at a point $\mathbf{r}$ due to a source $\mathbf{r}_0$. More precisely 1.14 satisfies,

$$(\nabla^2 + k^2)G_f(\mathbf{r}, \mathbf{r}_0) = \delta(\mathbf{r} - \mathbf{r}_0) \tag{1.15}$$

where δ is the Dirac Delta function, [24].

1.1.1 Practical Measures of Sound

The mean squared sound pressure is defined as,

$$(P^2)_{av} = \frac{1}{T}\int_0^T P^2 dt, \tag{1.16}$$

where T is the period for the averaging. In the time harmonic case, i.e. when $P = \Re(pe^{-i\omega t})$,

$$(P^2)_{av} = \frac{|p|^2}{2}. \tag{1.17}$$

The Sound Pressure Level, measured in decibels, is given by,

$$SPL = 10log_{10}\left(\frac{(P^2)_{av}}{(P_{ref})^2}\right)dB \tag{1.18}$$

where P_{ref} is a reference pressure, usually $2 \times 10^{-5}Nm^{-2}$. A concept which is also later referred to is the Excess Attenuation, EA, which is defined by,

$$EA = SPL_{FF} - SPL \tag{1.19}$$

where SPL is the actual sound pressure level, and SPL_{FF} is the sound pressure level that would have been measured in the same position relative to the source, if the propagation had been taking place in free field conditions.

Another useful quantity is the Insertion Loss, given by,

$$IL = SPL_{\beta_C} - SPL \tag{1.20}$$

where SPL_{β_C} is the sound pressure level with only flat ground present.

Chapter 2

The Boundary Element Method

2.1 Formulation of the problem

In order to efficiently model outdoor sound propagation to investigate noise barriers we use a two dimensional cross-sectional model, where the noise is a line source perpendicular to the page. We assume that any noise barriers we model sit on a infinitely long homogeneous plane. We also assume that the atmosphere is homogeneous, i.e no wind or temperature gradient are include in this model (for disscusion of how to get around this see literature review).

We assume that the region of propagation D lies in the upper half-plane $U = \{(x, y) : x \in \mathbb{R}, y > 0\}$, and that D is a local perturbation of U, i.e. for some $R > 0$, $\{\mathbf{r} \in D : |\mathbf{r}| > R\} = \{\mathbf{r} \in U : |\mathbf{r}| > R$ (see Figure 2.1). D is the region of propagation, ∂D denotes the boundary of D, and γ is the barrier surface, i.e. the part of ∂D which lies in the upper half-plane U. $\mathbf{r}$ position of the receiver point, and $\mathbf{r}_0$ is the source position.

For this two dimensional model the pressure satisfies equation (1.15), which states

$$(\nabla^2 + k^2)p(\mathbf{r}, \mathbf{r}_0) = \delta(\mathbf{r} - \mathbf{r}_0) \qquad \mathbf{r} \in D \tag{2.1}$$

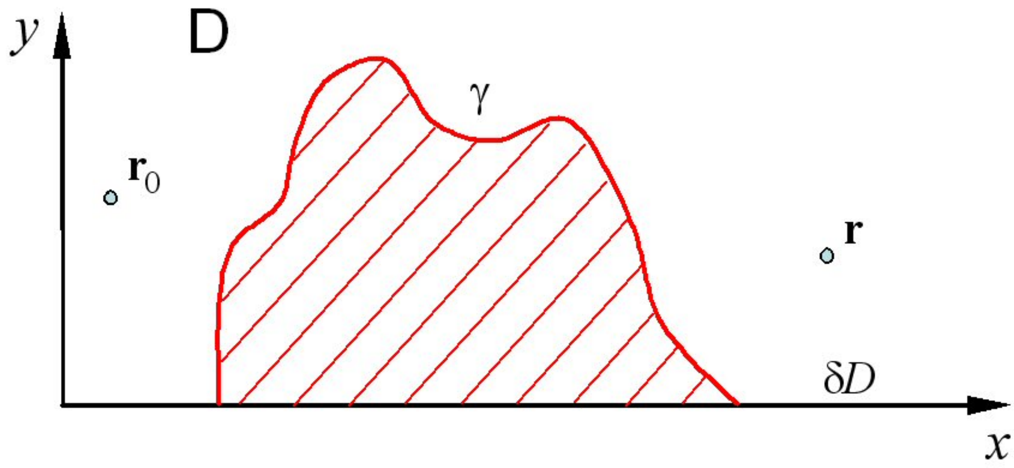


Figure 2.1: The Two-Dimensional Model.

where δ is the Dirac Delta function. It also satisfies the impedance boundary condition from (1.11), i.e.

$$\frac{\partial p}{\partial n} = ik\beta p, \qquad \mathbf{r} \in \partial D \tag{2.2}$$

and the Sommerfeld radiation conditions from (1.13),

$$\frac{\partial p}{\partial r} - ikp = o(r^{-1/2}) \tag{2.3}$$

$$p = O(r^{-1/2})$$

uniformly as $r \to \infty$. In this BVP the normal, $\frac{\partial}{\partial n(\mathbf{r})}$, is directed out of D and into ∂D.

2.2 Literature Review

Given a boundary value problem(BVP), i.e. a differential equation (e.g. Helmholtz) and a set of boundary conditions, the most common solution methods are the finite difference method (FDM) or finite element method (FEM). The Boundary element method (BEM) is a more recent numerical method, which is based on transforming a problem from a Boundary Value problem to an integral equation applied only to the boundary of the region and hence incorporating the boundary conditions directly. The advantage of the BEM is that it can easily accommodate complex boundary geometries, and so is well suited to the problem of traffic noise barriers.

The BEM can be traced back to 1903, when it was used as an integral equation method by Fredholm [1]. Then in the sixties with the development of computers, which enabled numerical calculations, the BEM found its way into engineering applications. In potential flow problems such methods where referred to as panel methods, see Hess [2] and Hess & Smith [3]. However it wasn't until 1978 that it was used to model outdoor sound propagation by Daumas [4]. The method was used for predicting acoustic field around vertical screens on a flat rigid ground surface. The BEM was first applied to barriers of arbitrary cross section and more complicated absorptive surfaces by Seznec [5]. These ideas where then developed further by Chandler-Wilde *et al* [6] and Hothersall *et al* [7]. In these papers they discuss the mathematics of the method and how it is applied to noise barrier problems using a computer program. The results are compared to outdoor model experiments carried out by Rasmussen [11]

One of the most widely used methods of noise control in Britain are earth bunds. The BEM is applied to the problem of optimising the shape of earth bunds and combining earth bunds with noise barrier configurations by Watts [14]. Increasing the angle of the fount of the slope of the bund (figure (2.2)) improve the screening. The design shown in figure (2.2d) proving to be most

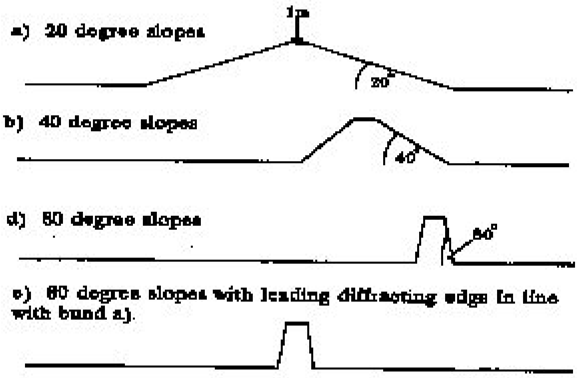


Figure 2.2: Examples of earth bunds with modified slope angles [8].

effective. In terms of applying barriers to the tops of the bund, a combination of 3 0.5 m high barriers placed at 0.5 m centres improved the screening by a significant amount compared to an earth bund of the same overall height.

Much work has been done using the BEM to investigate modifications which can be made to existing noise barriers, Watts *et al* [13]. The obvious way of improving a noise barrier is to increase the height, however this often requires the foundations to improved and can be visually intrusive. The transport research laboratory(TRL) have performed extensive studies of the performance of noise barrier caps which can be fitted to on the top of existing noise barriers. Figure (2.2) shows some of the examples of different types of barrier and their relative insertion losses for traffic noise. It was found that the BEM gives good agreement with full-scale measurements however it tends to slightly over predict the performance of T-shaped barriers and slightly under-predict the performance of multiple-edge configurations, this is possibly due to assumptions made in the BEM model regarding surface absorption.

The BEM has also been applied to investigate the effects of porous as-

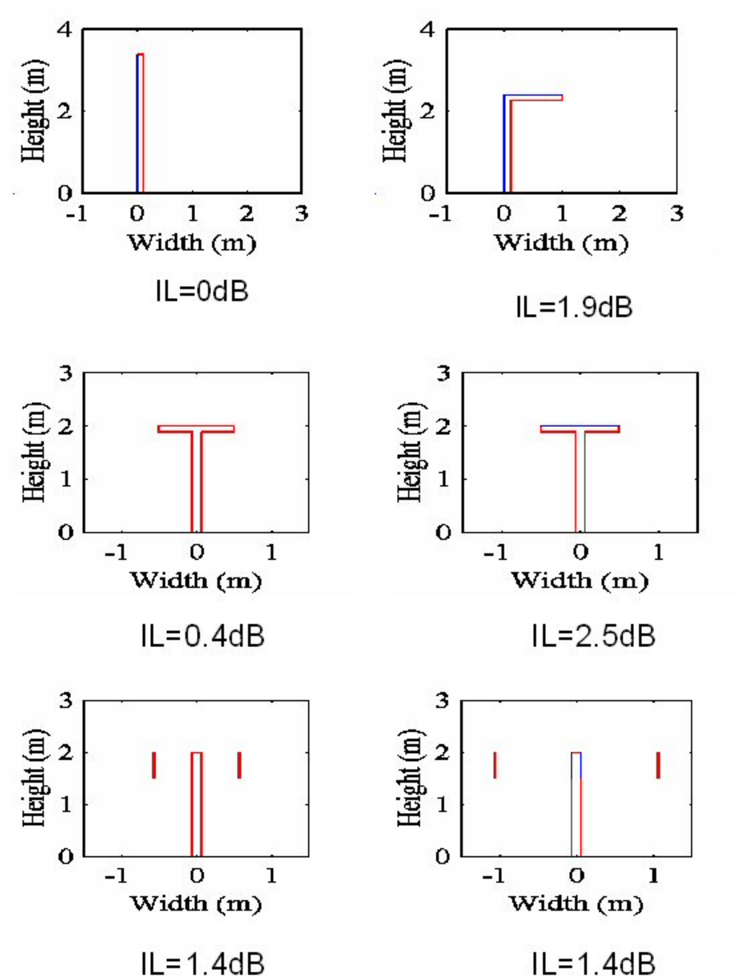


Figure 2.3: Examples of Barrier Profiles and Insertion Loss [8].

phalt(PA) road surfaces on reducing traffic noise by Watts *et al* [9]. It was discovered that the effects of PA were reduced when used reduced when used in conjunction with noise barriers. Multiple reflections occur when there are buildings on both sides of the road, and obviously noise barriers cannot be used. Using the BEM calculations were done for a single facade on one side

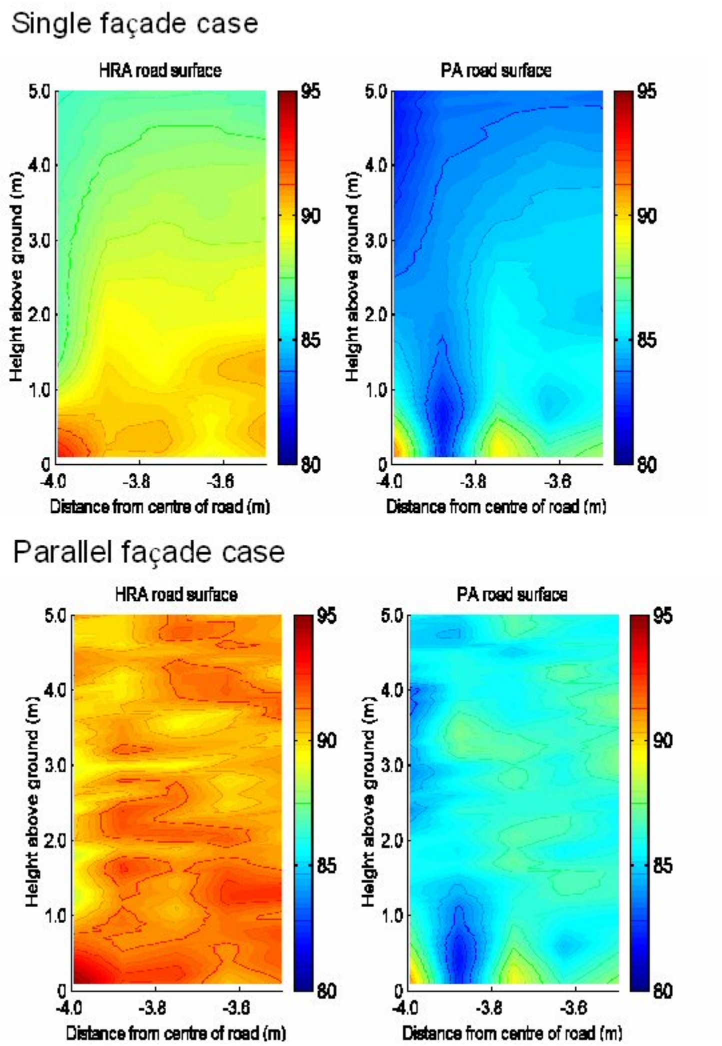


Figure 2.4: Noise levels at facades in narrow streets [8].

of the road and for facades on both side of the road with and without PA. figure (2.2) shows contour plots of sound field close to building facade [8]. There was a much greater reduction in the parallel facade case, which would suggest that PA would be an effective solution to reduce traffic noise levels in built up areas.

Morgan *et al* [10] details a modification of the BEM which allows for

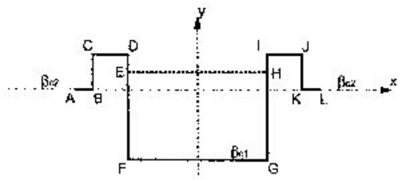


Figure 2.5: Cross section with cutting [8].

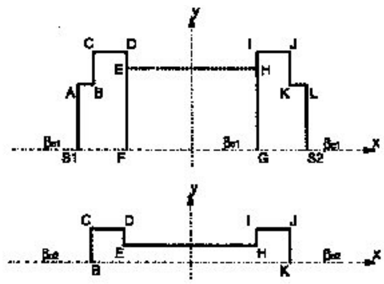


Figure 2.6: Stages of the Modified BEM [8].

cutting (which means part of the surface is below ground level), something which the standard BEM does not allow, see figure 2.2. A two stage process is used were a shortened section is raised above the ground and the pressures are calculated with the usual method. Then in the second stage the a revised cross section based on the first calculation is used to calculate the final pressure at the receiver point, see figure (2.2). This is quite an important feature as many roads or railways run in a cutting or on an embankment.

The BEM has also been applied to planes other than the cross section of the road, in order to model barriers with varying profile. In order to do this a 2-D model where the view is from above is used, the barrier is assumed to have infinite height, and the line source is replace by a series of point sources. Watts [15] used this technique to investigate the performance of Louvred barriers, and found the predictions of the method to give good agreement with scale model experiments.

More recently a new BEM method called the Meteo-BEM has been developed, Premat *et al* [16]. This is a method which allows outdoor sound to be modeled in a inhomogeneous medium. It does this by including meteorological effects in the Green's function. A height dependent Green's function is expressed , the sound pressure $p_s(r, z)$, is a function of both range r, and height z.

The results using the new method compared give good agreement with experimental results. The method is however more computationally expensive.

Another extension of the BEM, which accounts for a refracting atmosphere as well as a non-uniform boundary is detailed in Taherzade [17]. This time the Green's function is evaluated using the fast field program method(FFP), [18]. This method is known as BIE-FFP and represents considerable improvements on previous applications of BEM to barriers with a refracting atmosphere.

The BEM has also been applied to transport noise problems in three dimensions, however this increases the computational times significantly. Studies into truck-tyre shields and enclosures were investigated using a three dimensional BEM by TRL and details are reported in Philips *et al* [19].

2.3 The Boundary Integral Equation

We then employ the BEM to solve this BVP. The first step in the BEM is to convert the BVP to a boundary integral equation (BIE). This requires a fundamental solution of the Helmholtz equation which also satisfies the Sommerfeld radiation conditions. One such known function is th two dimensional free field Green's function defined by the equation

$$G_f(\mathbf{r}, \mathbf{r}_0) := -\frac{i}{4} H_0^{(1)}(k|\mathbf{r} - \mathbf{r}_0|) \tag{2.4}$$

where $H_0^{(1)}$ is the Hankel function of the first kind of order zero.

Let the Green's function for the upper half-plane with homogeneous impedance boundary condition with constant admittance, β_c, be denoted by G_{β_c}. So G_{β_c} is the solution to the BVP, in the case when no barrier is present, when D is the upper half-plane, and β in 2.2 has the constant value β_c. Explicitly

$$G_{\beta_c}(\mathbf{r}, \mathbf{r}_0) = G_f(\mathbf{r}, \mathbf{r}_0) + G_f(\mathbf{r}, \mathbf{r}_0^{'}) + P_{\beta_c}(\mathbf{r}, \mathbf{r}_0) \tag{2.5}$$

where $G_f(\mathbf{r}, \mathbf{r}_0)$ is the direct wave contribution, $G_f(\mathbf{r}, \mathbf{r}_0^{'})$ is the reflected wave contribution, with $\mathbf{r}_0^{'} = (x_0, -y_0)$ the image of the source in the boundary. $P_{\beta_c}(\mathbf{r}, \mathbf{r}_0)$ is a correction factor to account for non-zero boundary admittance, explicitly

$$P_{\beta_c}(\mathbf{r}, \mathbf{r}_0) = \widehat{P}_{\beta_c}(k(x - x_0), k(y + y_0)) \qquad \mathbf{r}, \mathbf{r}_0 \in \overline{D} \tag{2.6}$$

where, for $\xi \in \mathbb{R}$, $y \geq 0$,

$$\widehat{P}_{\beta_c}(\xi, \eta) = \frac{i\beta_c}{2\pi} \int_{-\infty}^{+\infty} \frac{exp(i(\eta(1 - s^2)^{\frac{1}{2}} - \xi s))}{(1 - s^2)^{\frac{1}{2}} + \beta_c} ds, \tag{2.7}$$

and

$$\Re\{(1 - s^2)^{\frac{1}{2}}\} \qquad \& \qquad \Im\{(1 - s^2)^{\frac{1}{2}}\} \qquad \geq 0.$$

The efficiency of the BEM depends on the efficient calculation of $G_{\beta_c}(\mathbf{r}, \mathbf{r}_0)$. In Matlab we have efficient built in functions to calculate the Hankel function $H_0^{(1)}$, and efficient and accurate approximations for $P_{\beta_c}(\mathbf{r}, \mathbf{r}_0)$ are given in [20].

Apply Green's second theorem to the functions $v = G_{\beta_c}(., \mathbf{r})$ and $u = p(., \mathbf{r}_0)$ in a region E consisting of part of D contained in a large circle of radius R centered on the origin, excluding small circles of radius ϵ around $\mathbf{r}$ and $\mathbf{r}_0$. Since $\nabla^2 u + k^2 u = \nabla^2 v + k^2 v = 0$ in E, we obtain

$$\int_{\partial E} \left(u \frac{\partial v}{\partial n} - v \frac{\partial u}{\partial n} \right) ds = 0. \tag{2.8}$$

Letting $\epsilon \to 0$ and $R \to \infty$ we obtain,

$$\epsilon(\mathbf{r}) p(\mathbf{r}, \mathbf{r}_0) = G_{\beta_c}(\mathbf{r}, \mathbf{r}_0) + \int_\gamma \left(\frac{\partial G_{\beta_c}(\mathbf{r_s}, \mathbf{r})}{\partial n(\mathbf{r}_s)} - ik\beta(\mathbf{r}_s) G_{\beta_c}(\mathbf{r}_s, \mathbf{r}) \right) p(\mathbf{r}_s, \mathbf{r}_0) ds(\mathbf{r}_s) \tag{2.9}$$

where $ds(\mathbf{r}_s)$ is the arc length of an element on γ at $\mathbf{r} = (x_s, y_s)$, and $\epsilon(\mathbf{r}) = 1$ when $\mathbf{r}$ lies anywhere in D except on the barrier surface γ, $\epsilon(\mathbf{r}) = \frac{1}{2}$ if $\mathbf{r}$ is at some point on γ which is not a corner point, $\epsilon(\mathbf{r}) = \frac{\Omega}{2\pi}$ if $\mathbf{r} = (x, y)$ is a corner point on γ where $y > 0$, where Ω is the angle in the medium subtended by two tangents to the boundary at $\mathbf{r}$, and $\epsilon(\mathbf{r}) = \frac{\Omega}{\pi}$ if $\mathbf{r} = (x, 0)$ is a corner on the ground surface.

Equation 2.9, the BIE, expresses the pressure at a receiver point in the region D solely in terms of the pressure on γ. It can be shown that the BIE is formally equivalent to the BVP, [6].

2.4 Numerical Solution of the BIE

In order to numerically solve the BIE we use a boundary element collocation method. We assume that γ is a polygonal consisting of N straight line segments, $\gamma_1, \gamma_2, \gamma_3, ..., \gamma_N$, where the mid-point of each element is $\mathbf{r}_n = (x_n, y_n)$, and the length is denoted by h_n, if γ is not a polygon then it first is to be

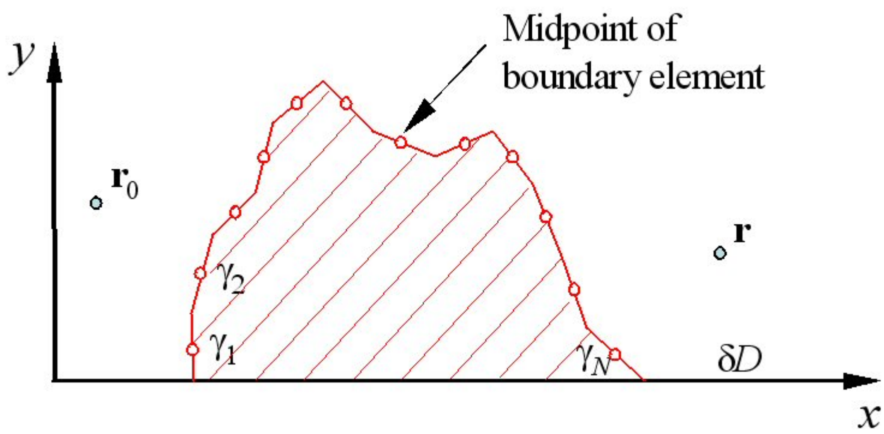


Figure 2.7: The Two-Dimensional Model.

approximated in this way (see Figure 2.7). Then 2.9 can be written as

$$\epsilon(\mathbf{r})p(\mathbf{r},\mathbf{r}_0) = G_{\beta_c}(\mathbf{r}_0,\mathbf{r}) + \sum_{n=1}^{N}\int_{\gamma_n}\left\{\frac{\partial G_{\beta_c}(\mathbf{r}_s,\mathbf{r})}{\partial n(\mathbf{r}_s)} - ik\beta(\mathbf{r}_s)G_{\beta_c}(\mathbf{r}_s,\mathbf{r})\right\}p(\mathbf{r}_s,\mathbf{r_0})ds(\mathbf{r_s}) \tag{2.10}$$

which can be approximated by [23],

$$\epsilon(\mathbf{r})p(\mathbf{r},\mathbf{r}_0 \approx G_{\beta_c}(\mathbf{r}_0,\mathbf{r}) + \sum_{n=1}^{N}\{B(\mathbf{r},\gamma_n) - ik\beta(\mathbf{r_n})C(\mathbf{r},\gamma_n)\}p(\mathbf{r}_n,\mathbf{r}_0). \tag{2.11}$$

For $D := \mathbf{r} \in D \cup \partial D$, $\mathbf{r}_0 \in D$,

$$C(\mathbf{r},\gamma_n) := \int_{\gamma_n} G_{\beta_c}(\mathbf{r}_s,\mathbf{r})ds(\mathbf{r}_s) \tag{2.12}$$

$$B(\mathbf{r},\gamma_n) := \int_{\gamma_n} \frac{\partial G_{\beta_c}(\mathbf{r}_s,\mathbf{r})}{\partial n(\mathbf{r}_s)}ds(\mathbf{r}_s) \tag{2.13}$$

are single-layer and double-layer potentials respectively. From 2.5 we can write

$$C(\mathbf{r},\gamma) = E(\mathbf{r},\gamma_n) + E(\mathbf{r},\gamma_n^{'}) + C_p(\mathbf{r},\gamma), \tag{2.14}$$

$$B(\mathbf{r},\gamma) = D(\mathbf{r},\gamma_n) + D(\mathbf{r},\gamma_n^{'}) + B_p(\mathbf{r},\gamma), \tag{2.15}$$

where, for a given straight line arc Γ and $\mathbf{r} \in \mathbf{R}^2$,

$$D(\mathbf{r}, \Gamma) := \int_\Gamma \frac{\partial G_f(\mathbf{r}, \mathbf{r}_s)}{\partial n(\mathbf{r}_s)} ds(\mathbf{r}_s), \tag{2.16}$$

$$E(\mathbf{r}, \Gamma) := \int_\Gamma G_f(\mathbf{r}, \mathbf{r}_s) ds(\mathbf{r}_s), \tag{2.17}$$

while

$$B_p(\mathbf{r}, \gamma_n) := \int_{\gamma_n} \frac{\partial P_{\beta_c}(\mathbf{r}, \mathbf{r}_s)}{\partial n(\mathbf{r}_s)} ds(\mathbf{r}_s), \tag{2.18}$$

$$C_p(\mathbf{r}, \gamma_n) := \int_{\gamma_n} P_{\beta_c}(\mathbf{r}, \mathbf{r}_s) ds(\mathbf{r}_s). \tag{2.19}$$

In equation 2.11 we can replace the single and double layer potential, C and B, by approximations c and b respectively, where

$$c(\mathbf{r}, \gamma_n) := e(\mathbf{r}, \gamma_n) + e(\mathbf{r}, \gamma_n^{'}) + c_p(\mathbf{r}, \gamma_n), \tag{2.20}$$

$$b(\mathbf{r}, \gamma_n) := d(\mathbf{r}, \gamma_n) + d(\mathbf{r}, \gamma_n^{'}) + b_p(\mathbf{r}, \gamma_n). \tag{2.21}$$

Here d and e denote accurate product midpoint rule approximations to D and E, given in Chandler-Wilde *et al* [6]. b_p and c_p are approximations to B_p and C_p respectively.
We approximate $C_p(\mathbf{r}, \gamma_n)$ by the midpoint rule,

$$c_p(\mathbf{r}, \gamma_n) := h_n P(\mathbf{r}, \mathbf{r}_n), \tag{2.22}$$

and from Morgan [23] we can approximate, $B_p(\mathbf{r}, \gamma_n)$, to

$$b_p(\mathbf{r}, \gamma_n) := h_n n_x(\mathbf{r}_n) \frac{\partial P_{beta_c}(\mathbf{r}, \mathbf{r}_s)}{\partial x_s}\bigg|_{\mathbf{r}_s = \mathbf{r}_n} - ik\beta_c n_y(\mathbf{r}_n)[2e(\mathbf{r}, \gamma_n^{'}) + c_p(\mathbf{r}, \gamma_n)]. \tag{2.23}$$

To obtain the numerical solution p_N, we need to solve:

$$\epsilon(\mathbf{r}) p_N(\mathbf{r}, \mathbf{r}_0) = G_{\beta_c}(\mathbf{r}_0, \mathbf{r}) + \sum_{n=1}^{N} \{b(\mathbf{r}, \gamma_n) - ik\beta(\mathbf{r}_n) c(\mathbf{r}, \gamma_n)\} p_N(\mathbf{r}_n, \mathbf{r}_0), \qquad \mathbf{r} \in \overline{D} \tag{2.24}$$

This expresses p_N at the receiver point in terms of values of p_N at the mid-points of the N elements. So to find the pressure at the receiver point we first need to determine the pressure at these N positions. We can do this by setting $\mathbf{r} = \mathbf{r}_n$, for $n = 1, 2, ...N$, in equation (2.24). This results in a set of N linear equations with unknowns $p_N(\mathbf{r}_n, \mathbf{r}_0)$,

$$\sum_{n=1}^{N} a_{mn} p_N(\mathbf{r}_n, \mathbf{r}_0) = G_{beta_c}(\mathbf{r}_0, \mathbf{r}_m), \qquad m = 1, 2, ...N, \tag{2.25}$$

where

$$a_{mn} = \frac{1}{2}\delta_{mn} - b(\mathbf{r}_m, \gamma_n) + ik\beta(\mathbf{r}_n)c(\mathbf{r}, \gamma_n), \qquad m, n = 1, 2, ..., N. \tag{2.26}$$

2.5 Admittance Models

In order to calculate values for the normalised surface admittance of the impedance boundary, β_c, and the barrier surface, $\beta(\mathbf{r}_s)$, we use one of two models. Each surface is assumed to have the admittance of a rigidly backed porous layer. Where the porous layers is of depth, D, with impedance, Z_b, and complex wavenumber, k_b. The Attenborough model expresses Z_b and k_b as functions of frequency in terms of the materials porosity, flow resistivity, tortousity, and pore shape, detailed in Chandler-Wilde *et al* [21]. Delany and Bazeley present formulae for Z_b and k_b as functions of frequency and the flow resistivity of the porous medium, detailed in Delany *et al* [22].

For the results in this report we use the Delany and Bazeley model , where the admittance of ground, β_G, is given by

$$1/\beta_G = 1 + 9.08(1000f/\sigma)^{-0.75} + i11.9(1000f/\sigma)^{-0.73}, \tag{2.27}$$

where f, σ are the frequency and flow resistance in SI units. The normal surface admittance is

$$\beta = -tan(Tk_G)\beta_G, \tag{2.28}$$

where k_G is the wave number in the soft ground which is given by:

$$k_G = k\{1 + 10.8(1000f/\sigma)^{-0.70} + i10.3(1000f/\sigma)^{-0.59}\}. \tag{2.29}$$

Chapter 3

How the Program Works

The code used in this project is based on an existing BEM code by Chandler-Wilde and D.C. Hothersall in 1995 using Fortran 77.

The main structure of the Fortran 77 code was maintained when converted to Matlab. Most of the sub-routines were converted directly to functions in Matlab.

Although running code in Matlab is not as fast as using Fortran, it does have several advantages. Matlab has many in built functions, making vector and matrix manipulation much simpler, thus making numerical solutions much easier to program.

Another advantage of Matlab is variables don't need to be declared before they are are used and arrays/vectors/matrices do not need to be of a pre-allocated size. This means code is much more compact and easier to read. Not all of the sub-routines need converted as many are already built in to Matlab, such as the Hankel functions.

3.1 Overview

The program uses the Boundary Element Method, described in the previous chapter, to numerically calculate the sound pressure at a given receiver point, or a number of receiver points due to the propagation of a line source over one or more noise barriers sitting on homogeneous flat ground. The program has the following restrictions:

(i) The line source is a coherent line source.

(ii) Only two dimensional situations can be modelled, that means each noise barrier must be of infinite length and parallel to the line source. This also means there can be no variance in the barrier shape or in the ground or barrier impedance, in the direction parallel to the line source.

(iii) One or multiple noise barriers can be modelled.

(iv) The cross-section of each noise barrier must be polygonal, but otherwise arbitrary.

(v) Each face of the barrier may have a different surface impedance. The ground may also have finite or infinite surface impedance.

(vi) The spectrum of the source can be completely arbitrary, and several different spectra can simultaneously be calculated at one time.

(vii) The element length, $HCON$, specifies the accuracy of the solution. For reasonable results $HCON < 0.2$.

(viii) The following condition

$$NMAX > (L * FREQ(I)/(C * HCON(I)))$$

must be satisfied for I = 1,2,3...,NFREQ, where C = 343m/s, L is the sum of length of sections of the barriers.

(ix) No part of the barrier may be below the level of flat ground, as the program is not designed to deal with cuttings.

(x) The ground on which the barriers sit is assumed to be homogeneous in surface impedance. A region of impedance can be represented by placing a flat barrier of the required impedance on the ground.

3.2 Main Program

The main program reads in the data from two separate input files, checking that the input data is valid.

The first file, 'TINPUTSP', contains information regarding the number of frequencies and spectra for the program to be run. Then for each different frequency it has the element length in wavelengths, and the each SPL at 1M in free space for each of the spectra to be used for that frequency.

The second file, 'TINPUT', contains information on , the number of corners, the coordinates of the corners and which model to select for the absorbing surfaces. For each surface between corners it contains information on whether or not a barrier is placed on the homogeneous surface, the effective flow resistivity, the layer depth, the porosity, the tortuosity, and the dominant angle of incidence. Then it reads the surface properties of homogeneous ground, which are the flow resistivity, the layer depth, the porosity, the tortuosity, and the angle of incidence. It also reads the coordinates of the source, and the coordinates of each of the receiver points.

Once all the information has been read in, the main program then calls subroutine 'bari8c', which uses the boundary element method to perform the numerical calculations. For each of the receiver points for each of the spectra 'bari8c' returns the total SPL, the excess attenuation and the insertion loss.

The main program calculates the mean insertion loss over the receiver points. It then writes the original data to file 'barrier_output', along with the results for each source spectrum in turn.

(For a flow diagram of Matlab functions see appendix A, for sample files of 'TINPUT', 'TINPUTSP', and 'barrier_output' see appendix F)

3.3 Creating workspace and Calling the functions

Given the coordinates of the corners of the barrier and the spectral data, 'bari8c' returns the SPL at the given receiver points, and the EA and IL for the range of source spectra.

For each frequency the function calculates:-

the wave number,

the dimensionless coordinates of each of the corners,

the admittance of each face,

the geometrical boundary data by calling 'barie7' and 'bari6a',

the normalised surface admittance for each element is calculated by 'bari6b',

the admittance of the ground is calculated

the BIE is solved by calling 'barie1'

Then a second loop calculates the following for each receiver point:-

the acoustic pressure by quadrature over the boundary using 'barie2',

the acoustic pressure in field free conditions,

the acoustic pressure in absence of the barrier,

the attenuation in free field conditions,

the attenuation in presence of flat ground

the attenuation when the barrier is present

We then have a third loop which calculates for each given spectrum at each receiver point:-

the SPL in free field conditions,

the SPL in presence of flat ground ,

the SPL in presence of the barrier.

'bari8c' then uses equations (1.19) and (1.20) to calculate the excess attenuation and insertion loss at each receiver for each spectrum.

3.4 The Geometrical Data

'barie7' is called before a call to 'bari6a'. 'barie7' calculates the number of elements for each face of the barrier, given the required element length as a fraction of the wavelength. It also calculates the total number of elements in the barrier. 'bari6a' then calls 'barie5' for each face of the barrier. 'barie5' calculates the midpoint of each segment, the unit normal to each segment, and length of each segment. Once the geometrical data has been calculated 'bari6b' calculates the angle of incidence between each element and the source if the angle has not been given. It then calls complex functions 'sadme' to calculate the complex admittance of each element using the appropriate model.

3.5 Solving the BIE

Now that the barrier has been divided into straight line segments, 'barie1' approximately solves BIE on the barrier surface. It assumes that there is a constant pressure value on each segment. Then it calculates the coefficient matrix described by 2.26, using the function 'layers' which return value double and single given by midpoint rule approximations to equations 2.17 and 2.16 respectively. The RHS is the calculated using 'hnkl10' and 'pbeta' functions. Then the matrix system is solved using LDU composition, which is a built in function of Matlab, returning the complex pressure at each segment on the barrier. (See appendix B for 'barie1')

3.6 Calculation of acoustic Pressure

'barie2' calculates the pressure at a given field point, and can only be called after 'barie1'. The routine first calculates the pressure at the receiver point if the barrier were absent using 'hnkl10' and 'pbeta'. Then the correction for the barrier is added to this using 'layers' and 'pbetad'. 'pbeta' and 'pbetad' calculate functions used to calculate the acoustic potential on reflection of a cylindrical wave at a plane of homogeneous admittance. Depending on the complex admittance of the ground either a Laplace type integral or a representation as the sum of a Laplace type integral with an error function of complex argument is used. (See Appendix B for 'barie2', and Appendix C for 'layers', 'pbeta' and 'pbetad'.)

3.7 Single and Double Layer potentials

'layers' calculates approximations to the potential at a point due to single and double layer potential using a combination of analytic integration and the midpoint rule, detailed in the BEM section. Firstly it calculates the coordinates of the points relative to the rotated and displaced coordinate axes, which has origin at the centre of the given segment, and y axis in the opposite direction to that of the normal. Then the approximations to the single and double layer potentials are calculated using 'hnkin3' and 'hnksb1'. 'hnkin3' evaluates the BIE over a given element it calls 'hnkana', 'hnkin2'. (See appendix D for 'hnkin3','hnksb1','hnkana', 'hnkin2', and 'hnkl10'.)

3.8 Grading the Mesh

One way of improving the accuracy of the BEM is to grade the mesh. The mesh should be graded so that there are more points near the corners of the barrier. As we work in dimensionless coordinates when dealing with the barrier geometry, the first and last $k\lambda = 2\pi$ length of each barrier is graded. The mesh grading is given by

$$x_i = \left(\frac{i}{N}\right)^q 2\pi, \qquad i = 0, ..., N \tag{3.1}$$

where x_i is the distance of the end of the element from the corner, $q > k\beta$ is the severity of the grading and N is chosen so that

$$x_N - x_{N-1} = kh \tag{3.2}$$

where kh is the dimensionless size of the mesh spacing between the two graded sections, [26].

The function 'bari67' produces a graded mesh given a set of corners and a step size for the non-graded mid-sections of the barrier. Unfortunately due to time constraints it wasn't possible to fully integrate 'bari67' into the main code in order to produce results using a graded mesh with the BEM. (See appendix E for function 'bari67'.)

Chapter 4

Results

In order to test the Matlab BEM program results are compared with analytical results, experimental results, and other BEM calculations from Hothersall *et al* [7]. For all the results in this project the one third octave, A-weighted sound power level for a single light vehicle given in Morgan [23] is used.

4.1 Insertion Loss

To compare results of the BEM with an analytical solution, we need a two dimensional problem which can be solved exactly. So if we consider a semi-circular barrier of homogeneous admittance, on a flat boundary of zero admittance, the analytic solution can be determined (see figure (4.1)). The admittance of the barrier surface is defined by flow resistivity $\sigma = 300000 Nsm^{-4}$ with infinite layer thickness.In order to apply the BEM we first approximate the semi-circular section by an 18-sided polygon.

Figure (4.2) displays the results using Matlab BEM program for the same data, it can be seen that they give good agreement with the BEM and analytical results in figure (4.1). The differences observed at higher frequencies are attributed to the the high density of eigenfrequencies, which depend on the shape and area of the cross-section.

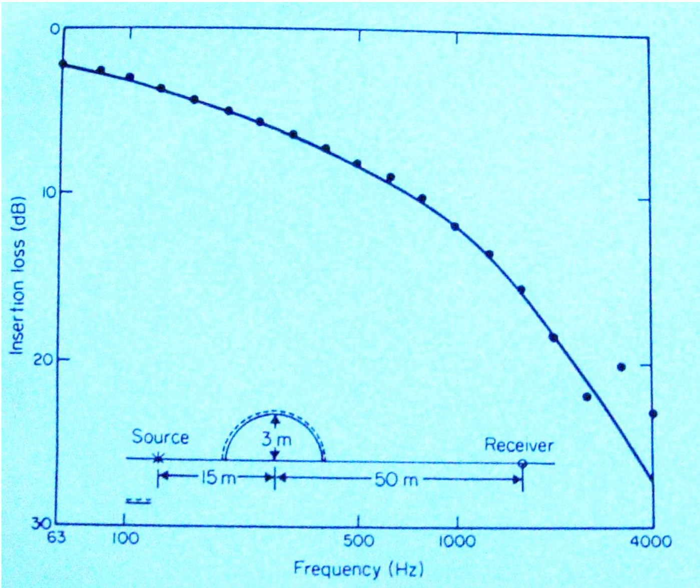


Figure 4.1: Comparison of analytical results (-) & numerical model (•) [7]

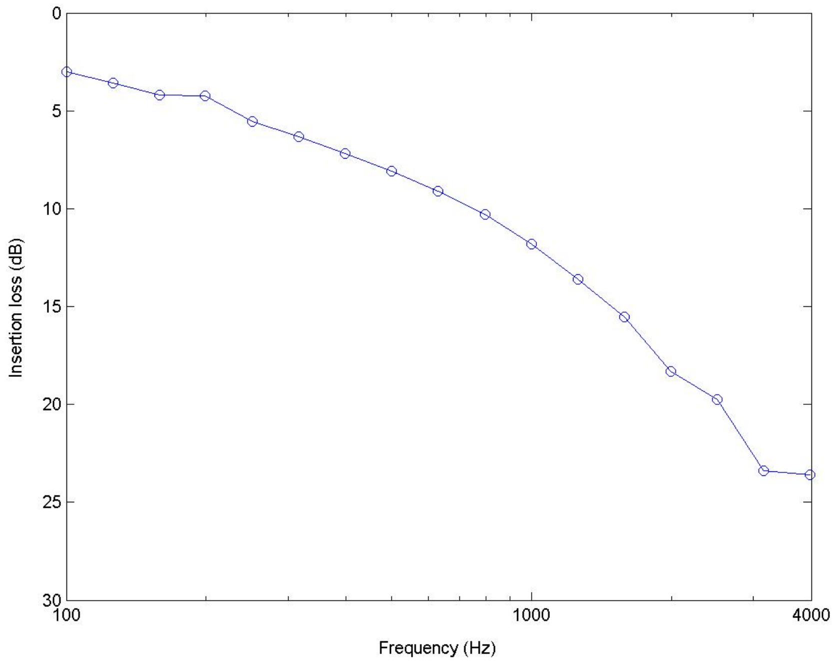


Figure 4.2: Matlab Results for semi-circular barrier configuration

4.2 Excess Attenuation

In figure (4.3) the results of an outdoor experiment carried out by Rasmussen [11] for the geometry indicated in the figure are compared with the results of Hothersall [7] BEM results. The admittance of the barrier surface is zero and the admittance of the ground surface is defined by $\sigma = 250000Nsm^{-4}$ with infinite depth.

Figure (4.4) displays the results using Matlab BEM program for the same data, it can be seen that they give good agreement with the BEM and experimental results in figure (4.3).

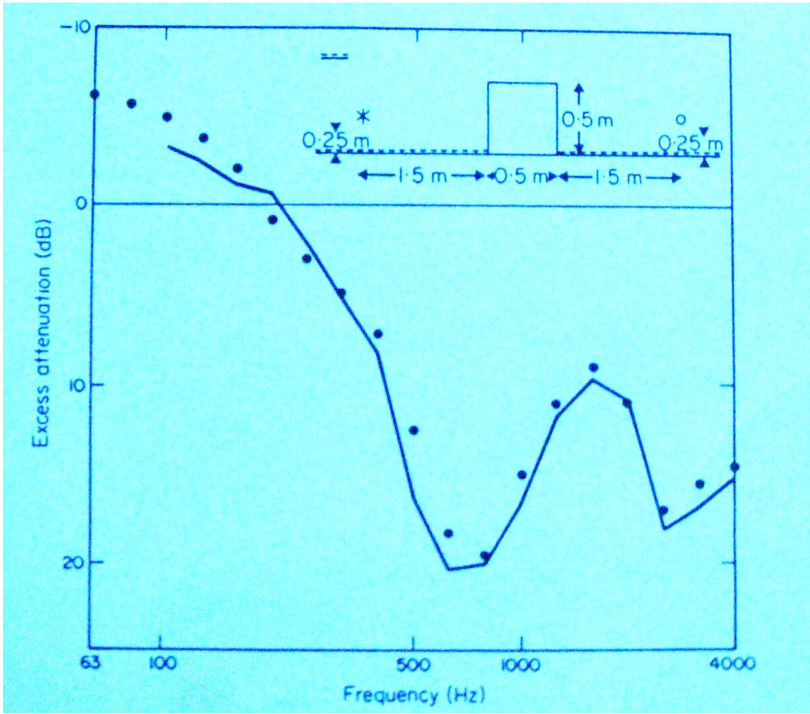


Figure 4.3: Comparison of experimental results (-) & numerical model (•) [7]

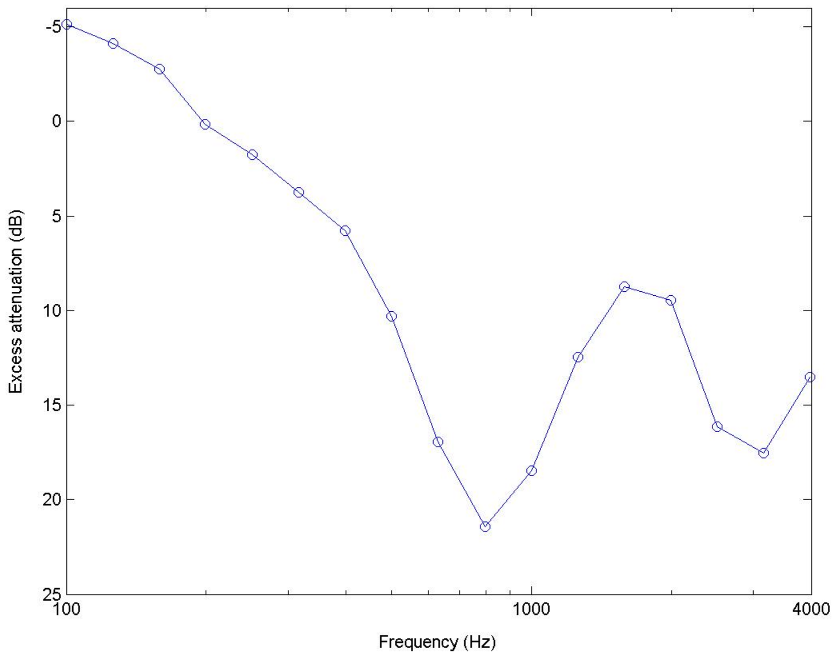


Figure 4.4: Matlab Results

4.3 Barrier Height

The insertion loss for varying barrier heights is shown in figure (4.5) as calculated by Hothersall [7] using the BEM. The barrier geometry, source position and receiver position are also given in figure (4.5). All surfaces have zero admittance.

Figure (4.6) displays the results using Matlab BEM program for the same data, it can be seen that they give good agreement with the BEM results in figure (4.5). From the results is clear that insertion loss increases with frequency, at the rate of $3 \approx dB/octave$. The insertion loss produced by increasing the height from 2m to 3m is almost double that obtained by increasing the height 4m to 5m.

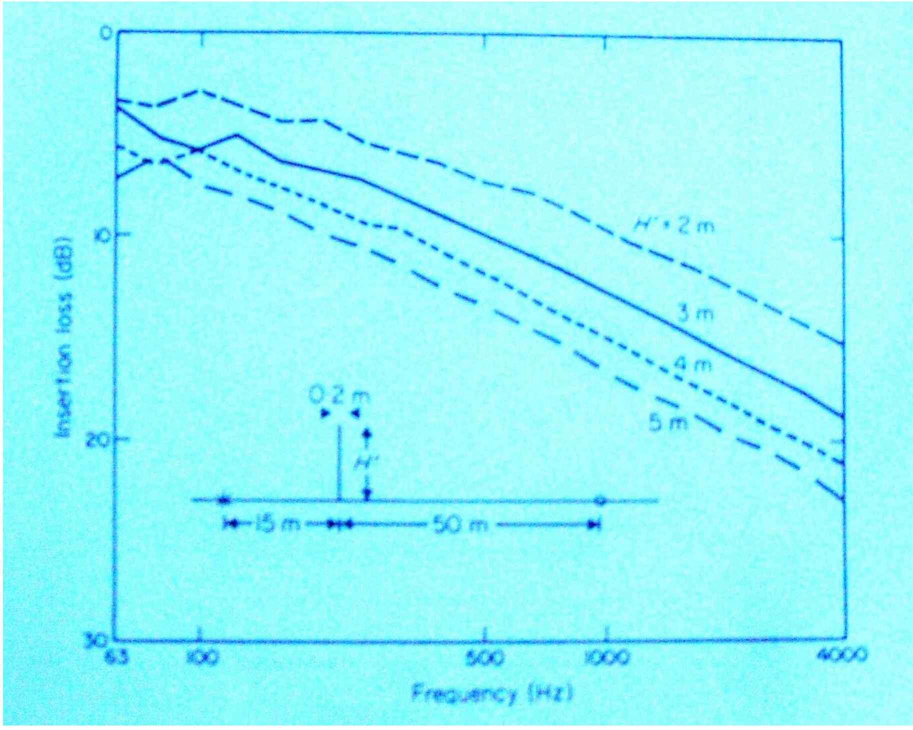


Figure 4.5: Insertion loss for range of barrier height [7]

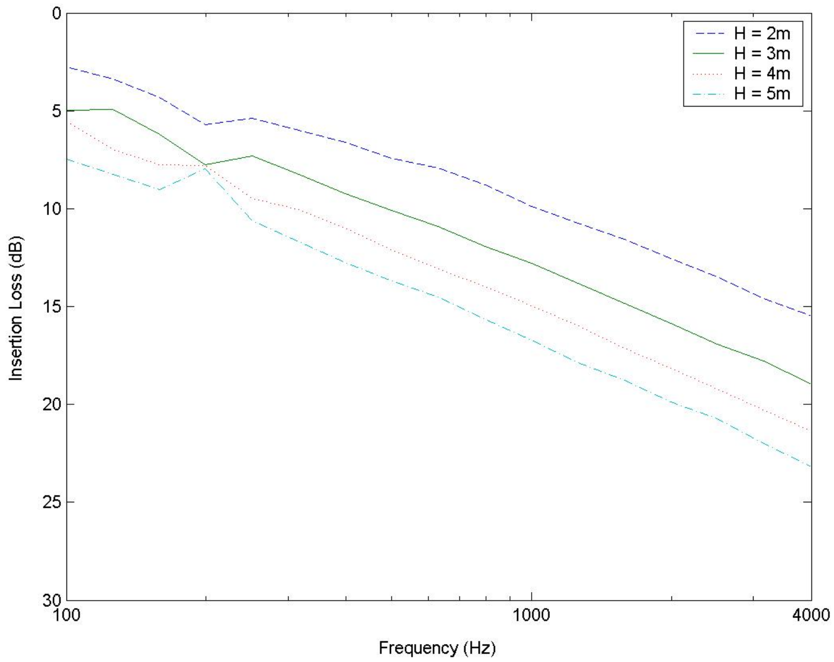


Figure 4.6: Matlab Results for same range of barrier heights

4.4 Ground Surface Cover

For a vertical wall and a broad wedge barrier the effects of changing the ground surface are investigated in figures (4.7) & (4.8). The barrier geometry is given in figure (4.7). The admittance of the surfaces with (−−) are defined by $\sigma = 250000 Nsm^{-4}$ with infinite surface depth.

Figure (4.8) displays the results using Matlab BEM program for the same data, again it can be seen that they give good agreement with the BEM results in figure (4.7). Although the barriers have different shapes it can be seen that the changing of the ground surface has a similar effect on both.

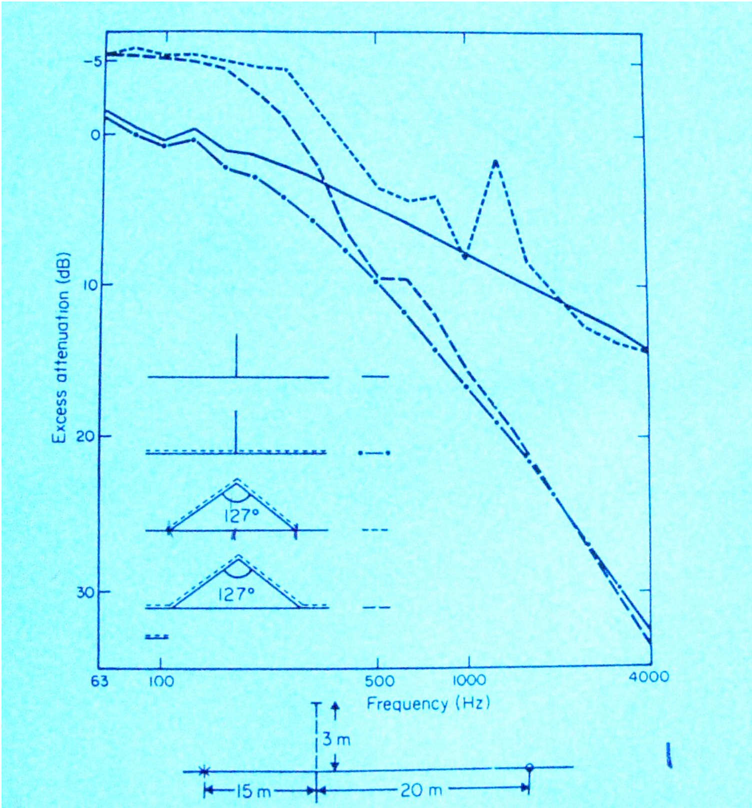


Figure 4.7: EA two different shapes of barrier for two different ground surfaces [7]

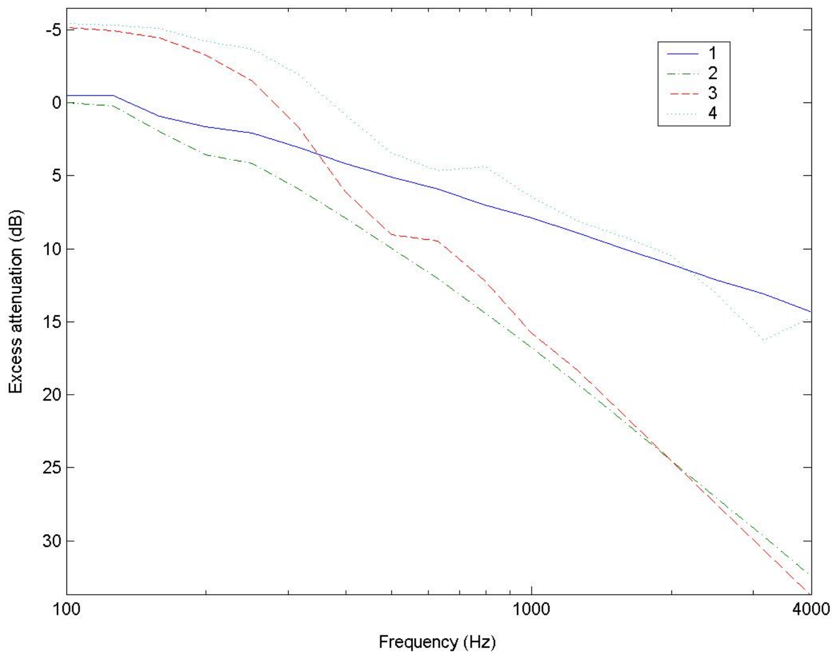


Figure 4.8: Matlab Results for same range of barriers and ground surfaces respectively

4.5 Barrier Surface Coverings

Barriers may be treated with acoustically absorbing materials. Figure (4.9) show the insertion loss of some such treated barriers. Surface admittance is zero except for those surface with the treatment indicated by ( ), where the admittance is defined by $\sigma = 20000Nsm^{-4}$ and $T = 0.1m$. The geometry of the barrier is also given in figure (4.9).

Figure (4.10) displays the results using Matlab BEM program for the same data, again it can be seen that they give good agreement with the BEM results in figure (4.9). It is clear that the treatment applied to the source side of the barrier produce no significant effect. The treated T-profile however provides significant improvement.

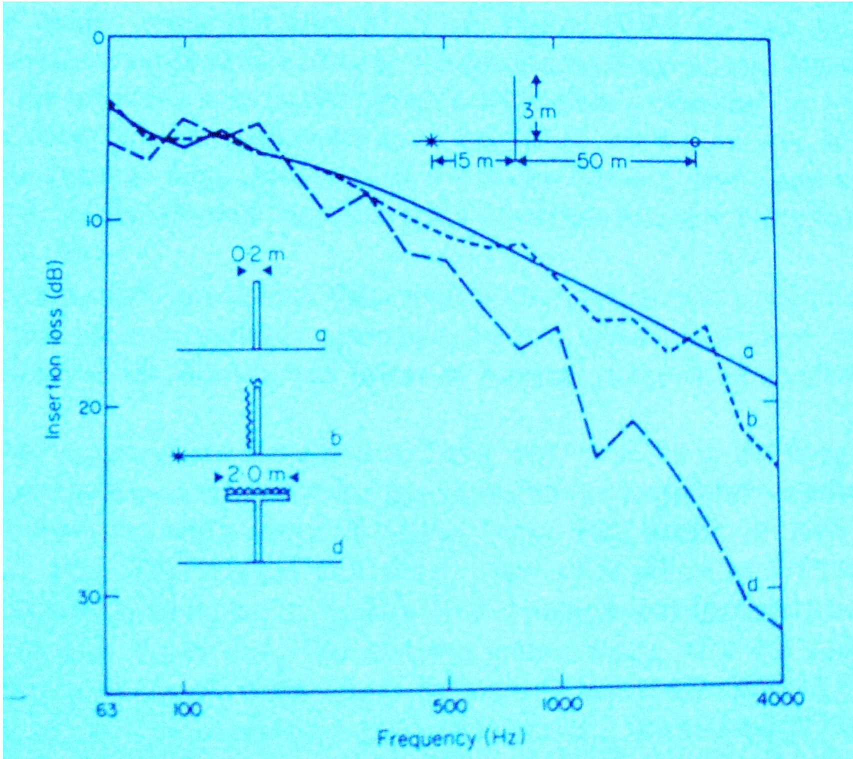


Figure 4.9: IL for surface treated barriers [7]

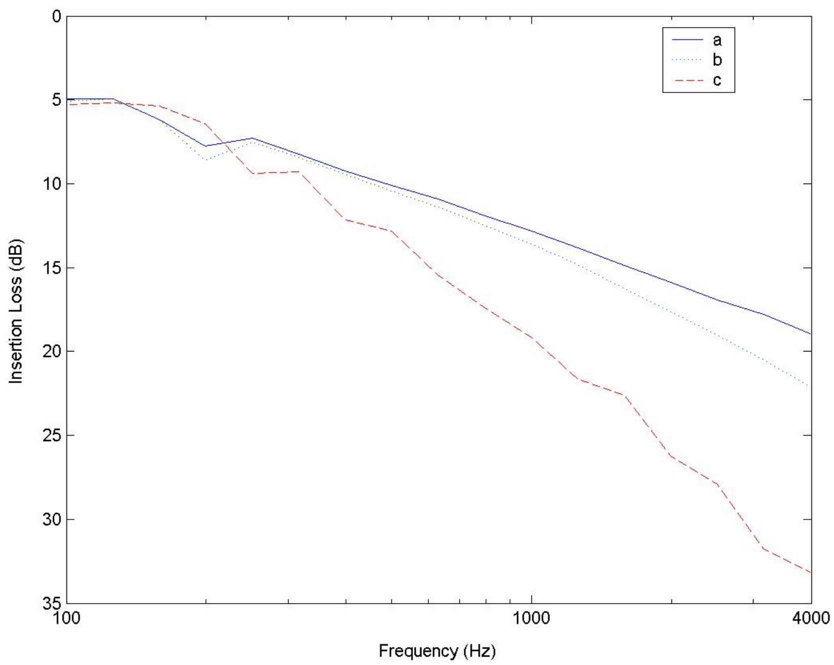


Figure 4.10: Matlab Results for same range of surface treated barriers

Chapter 5

Conclusions

The two dimensional boundary element method has been successfully applied to the problem of outdoor sound propagation, using the Matlab program described in this project. The flexibility of the BEM for modelling cross-sections with random shape and acoustical properties means that it is particularly applicable to transport a wide range of transport noise problems. It has been shown that the BEM is a powerful and accurate tool which with the use of arbitrary source spectra provides a means for devising traffic noise mitigation measures, without the need for full-scale measurements.

Although the Matlab BEM program described in this report is slower to run than previous BEM programs written in lower level languages, such the Fortran 77 program this is based on, it does offer several distinct advantages - readability, understandability, and graphic user interface (GUI). The GUI in Matlab makes the manipulation of results and plotting of graphs much easier.

The grading of the mesh discussed in this project would certainly be one way of improving the BEM. Other possible ideas for extending the work in this project are allowing for temperature and wind gradients. With the development of increasingly faster PCs three dimensional modelling will be able to offer even greater capabilities in the future.

Bibliography

[1] I.Fredholm. Sur une classe d'equations fontionelles. *Acta. Math 27:365-390, 1903'*

[2] J.L.Hess. Calculation of potential flow about bodies of revolution having axes perpindicular to the free stream direction. *J.Aero.Sci 29:726-742 1962*

[3] J.L.Hess & A.M.O.Smith. Calulation of nonlifting potential flow about arbitrary three-dimensional bodies. *J.Ship.Res 8:22-44 1964*

[4] A.Daumas. Study of diffraction by a thin screen set-up on ground. *Acustica 40,213-222 1978*

[5] R Seznec. Diffraction of sound around barriers: use of the boundary elements technique. *Journal of sound and vibration 73, 195-209 1980*

[6] S.N.Chandler-Wilde, D.C.Hothersall, D.H.Crombie & A.T.Peplow. Rencontres Scientifiques du Cinquintenaire: Ondes Acoustiques et Vibratoires, Interaction Fluide-Structures Vibrantes. *Publication du Laboratiores, Interaction Fluide-*

Structures Vibrantes, Publication du Laboratiore de Mécanique et d'Acoustique(CNRS Laboratoires, Marseille)126, 73-90 1991

[7] D.C.Hothersall, S.N.Chandler-Wilde & M.N.Hajmirzae. Efficiency of single noise barriers. *Journal of Sound and Vibration 146(2), 303-322, 1991*

[8] P.A.Morgan & G.R.Watts. Application of the Boundary Element Method to the design of transport noise mitigation measures. *Annual review paper V5*

[9] G.R.Watts, S.N.Chandler-Wilde & P.A.Morgan. The combined effects of porous asphalt surfacing and barriers on traffic noise. *Applied Acoustic 58, 351-377, 1999*

[10] P.A.Morgan, C.R.Ross & S.N.Chandler-Wilde. An efficient boundary element method for noise propagation from cuttings. *Proceedings of Inter-Noise, Liverpool, UK, 3011-3016, 1996*

[11] K.B.Ramussen Sound propagation over non-flat terrain. *Danish Acoustical Labratory Report No.35, The Danish Academy of Technical Sciences, 1982*

[12] G.R.Watts. Acoustic performance of parallel noise barriers. *Applied Acoustics 47(2), 95-119. 1996*

[13] G.R.Watts, D.H.Crombie & D.C.Hothersall. Acoustic performance of new designs of traffic noise barriers: Full-scale tests. *Journal of Sound and Vibration 177(3), 289-305, 1994*

[14] G.R.Watts. Effectiveness of novel shaped bunds in traffic noise. *Proceedings of the Institute of Acoustics 21(2),41-50, 1999*

[15] G.R.Watts, D.C.Hothersall, K.V.Horoshenkov. Measured and predicted acoustic performance of vertically lourved noise barriers. *Applied Acoustics 62(11), 1287-1311*

[16] E.Premat, Y.Gabillet. A new boundary-element method for predicting outdoor sound propagation and application to the case of a sound barrier in presence of downward refraction. Acoustical Society of America 108(6), 2775-2783, 2000

[17] S.Taherzadeh, K Ming Li, Keith Attenborough. A hybrid BIE/FFP scheme for predicting barrier efficiency outdoors. *Acoustical Society of America, 110(2), 918-924, 2001*

[18] S.J.Franke & G.W.Swenson,Jr. A Brief tutorial on the fast field program applied to sound propagation in air. *Applied Acoustics 27, 227-238, 1989*

[19] S.M.Philips, P.A.Morgan & G.R.Watts. Development of methods for understanding tyre/road noise generation and propagation. *Proceedings of Inter-Noise 99, Fort Lauderdale, Florida, USA, 219-222, 1996*

[20] S.N.Chandler-Wilde & D.C.Hothersall. The Green function for two-dimensional acoustic propagation over a homogeneous impedance plane. *Research Report, Department of Civil Engineering, University of Bradford, UK. 1991*

[21] S.N.Chandler-Wilde, & K.V.Horoshenkov. Padé approximants for the acoustical of characteristics of rigid frame porous media. *Journal of the Acoustic Society of America 98(2), 1119-1129, 1995*

[22] M.E.Delany, $ E.N.Bazley. Acoustical properties of fibrous absorbent materials. *Applied Acoustics 3, 105-116, 1970*

[23] P.A.Morgan. Boundary Element Modelling and Full Scale Measurement of the Acoustic Performance of Outdoor Noise Barriers *Phd Thesis, Dept. of Mathematical Sciences, Brunel University, 1999*

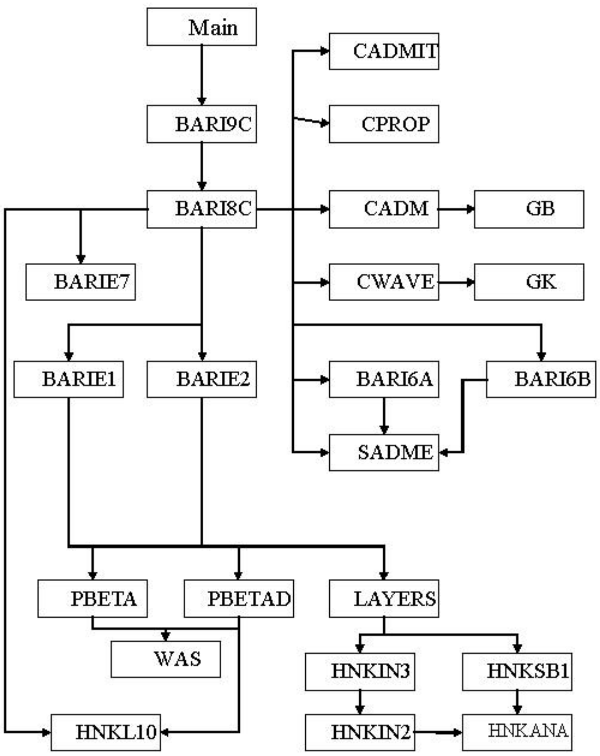
[24] S.N.Chandler-Wilde. Ground effects in Environmental Sound Propagation. *PhD Thesis, Dept of Civil and Structural Engineering University of Bradford, 1988*

[25] S.N.Chandler-Wilde, & S.Langdon. Boundary element methods for acoustics. *Notes, Dept of Mathematics University of Reading, 2005*

[26] Boundary Integral Equations of the Second Kind. *section 8.3*

Appendix A

Map of Functions



Appendix B

```
function [P,A,ANORM,WGT1,WGT2,WGT3,WGT4] = barie1(XM,YM,...
UNORMX,UNORMY,H,BETA,N,X0,Y0,BETAC,NA)
%
%
% function [P,A,ANORM,RINT,WGT1,WGT2,WGT3,WGT4] = barie1(XM,YM,...
% UNORMX,UNORMY,H,BETA,N,X0,Y0,BETAC,NA)
%
%
% Converted from FORTRAN Proplib routine created by S. Chandler Wilde 1987
%
%
% Given a subdivision of the barrier into N staright line segments, barie1
% solves a BIE on the barrier surface approximately, assuming, a constant
% pressure value in each segment.
%
% Note that dimensionless coordinates and lengths are used throughout,
% each length or coordinate made dimensionless by multiplying by the
% wavenumber = (2 * pi / wavelength) of medium propagation.
%
% Note that the cartesian cordinate system ysed has the straight
% boundary on which the barrier sits acts as the x-axis. The y-axis
% is vertically upwards (directed into the medium of propagation).
%
% Note also that an incident wave
%   -(i/4) * hnkl10(R)
% is asumed, where i = sqrt(-1), and R is the distance from the source
% point.
%
% Input:
%
% XM,YM         Real(N)
%               where XM(i) and YM(i) are the coordinates of the
%               midpoint of the ith segments
%
% UNORMX,       Real (N)
```

```
%  UNORMY        where UNORMX(i) and UNORMY(i) are the components of the
%                unit normal to the ith segment
%
% H              Real (N)
%                H(i) is the lenght of the ith segment
%
% BETA           Complex(N)
%                BETA(i) is the normalised admittance of the ith segment
%                Time dependance exp(-i*W*T) assumed.
%
% N              Integer
%                The number of line segments
%
% X0,Y0          Real
%                (X0,Y0) coordinates of the source point
%
% BETAC          Complex
%                The admitance of the flat ground on which the barrier sits
%
% NA             Integer
%                The first dimension of A in the calling program NA must be
%                greater than or equal to N
%
% Output:
%
%    WGT1,     COMPLEX(N)
%    WGT2,     WORKSPACE. ON EXIT THESE ARRAYS CONTAIN WEIGHTS USED
%    WGT3,     IN CONSTRUCTING THE MATRIX A.
%    WGT4
%
%
%
%    P         COMPLEX(N)
%              P(I) IS THE PRESSURE IN THE ITH SEGMENT
%
%    A         COMPLEX(NA,N)
%              A CONTAINS THE TRIANGULAR DECOMPOSITION OF THE COEFFICIENT
%              MATRIX OF THE SET OF LINEAR EQUATIONS THAT ARE SOLVED, AS
%              CALCULATED BY NAG ROUTINE F03AHF.
%
%    ANORM     REAL
%              THE NORM (MAXIMUM ROW SUM) OF THE COEFFICIENT MATRIX
%
%    RINT      REAL(N)
%              CONTAINS DETAILS OF THE ROW INTERCHANGES DURING THE
%              TRIANGULAR DECOMPOSITION OF THE COEFFICIENT MATRIX, AS
%             CALCULATED BY NAG ROUTINE F03AHF.
%
%  NOTE THAT, IF SEGMENT I LIES IN THE GROUND PLANE (WHICH
```

```
%  IS PERMITTED), IT IS IMPORTANT THAT YM(I) = 0.0, UNORMX(I) = 0.0,
%  AND UNORMY(I) = -1.0 EXACTLY ON ENTRY.
%
%  FUNCTIONS CALLED HNKL10,PBETA,LAYERS,PBETAD ARE CALLED
%
%
%  CALCULATE WEIGHTS FOR SINGLE LAYER POTENTIAL WITH SUPPORT SEGMENT I,
%  SINGLE LAYER POTENTIAL WITH SUPPORT THE IMAGE OF SEGMENT I, PBETAC,
%  AND D(PBETAC)/DX.
%
for j = 1:N
    WGT1(j) = -i*BETA(j);
    TEMP = -UNORMY(j)*i*BETAC;
    WGT2(j) = WGT1(j) + TEMP + TEMP;
    WGT3(j) = (WGT1(j)+TEMP)*H(j);
    WGT4(j) = H(j)*UNORMX(j);
end
%
%  NOW CALCULATE COEFFICIENT MATRIX
%
for j = 1:N
    for k = 1:N
        A(k,j) = complex(0.0,0.0);
    end
end


for j = 1:N
    if YM(j) > 0.0
        A(j,j) = complex(0.5,0.0);
    else
        A(j,j) = complex(1.0,0.0);
    end
end



for J = 1:N
    for I = 1:N

        [SINGLE,DOUBLE] = layers(XM(I),YM(I),XM(J),YM(J),UNORMX(J),UNORMY(J),H(J));
        A(I,J) = A(I,J) - DOUBLE - WGT1(J)*SINGLE;
        [SINGLE,DOUBLE] = layers(XM(I),YM(I),XM(J),-YM(J),UNORMX(J),-UNORMY(J), H(J));
        A(I,J) = A(I,J) - DOUBLE - WGT2(J)*SINGLE;
    end
 end
```

```
 if BETAC ~= 0.0
    for I = 1:N-1
        for J = I+1:N
            [PBETA1,PBETAX,PBETAY] = pbetad(XM(J)-XM(I),YM(J)+YM(I),BETAC);
            A(I,J) = A(I,J) - WGT3(J)*PBETA1 - WGT4(J)*PBETAX;
            A(J,I) = A(J,I) - WGT3(I)*PBETA1 + WGT4(I)*PBETAX;
        end
    end


    for I = 1:N
        if YM(I) == 0.0
            A(I,I) = A(I,I) - WGT3(I)*0.5*(pbeta(0.0,0.0,BETAC) + pbeta(0.5*H(I),0.0,BETAC));
        else
            A(I,I) = A(I,I) - WGT3(I)*pbeta(0.0,YM(I)+YM(I),BETAC);
        end
    end
 end

%  NOW CALCULATE THE NORM OF THE COEFFICINT MATRIX
%
      ANORM = norm(A,inf);

%  NOW CALCULATE THE RHS VECTOR
%
for I = 1:N
         XDIF = XM(I) - X0;
         YDIF = YM(I) - Y0;
         YSUM = YM(I) + Y0;
         XDIFS = XDIF*XDIF;
         RDIR = sqrt(XDIFS+YDIF*YDIF);
         RREF = sqrt(XDIFS+YSUM*YSUM);
         P(I) = complex(0.0,-0.25)*( hnkl10(RDIR) + hnkl10(RREF) ) + pbeta(XDIF,YSUM,BETAC);
end
%P
%
%  NOW SOLVE THE LINEAR SYSTEM WITH COEFFICIENT MATRIX A, AND RHS P.
%  THE SOLUTION IS STORED IN P.
%

P=conj(P);
%A
[L,U] = lu(A);
P = U\(L\P');


function BARIE2 = barie2(XM,YM,UNORMX,UNORMY,H,P,...
WGT1,WGT2,WGT3,WGT4,N,X0,Y0,X,Y,BETAC)
```

```
%
%
% function BARIE2 = barie2(XM,YM,UNORMX,UNORMY,H,P,...
% WGT1,WGT2,WGT3,WGT4,N,X0,Y0,X,Y,BETAC)
%
% Converted from FORTRAN Proplib routine created by S. Chandler Wilde 1987.
%
%        BARIE SUITE OF FUNCTIONS FOR BOUNDARY ELEMENT SOLUTION
%        ------------------------------------------------------
%        OF 2-D BARRIER PROBLEMS
%        -----------------------
%
% barie 2 can be called, after the BIE has been solved by a call to
% barie1/4. barie2 calculates the pressure at a given field point in the
% medium, by a quadrature over the surface of the barrier. In the
% quadrature the barrier surface is divided into the same number of
% straight line segments as in barie1/4, and the pressure, for the purpose
% of the quadrature, is assumed constant in each segment.
%
% Inputs:
%
% XM,YM          Real vectors of dimension N, where XM(i) and YM(i) are the
%                coordinates of the midpoint of the ith segments
%
% UNORMX,        Real vectors of dimension N, where UNORMX(i) and UNORMY(i)
%  UNORMY        are the components of the unit normal to the ith segment
%
% H              Real vector of dimension N, H(i) is the lenght of the
%                ith segment
%
% P              Complex vector of dimension N, P(i) is the pressure in
%                the ith segment
%
% WGT1,WGT2,     Complex vectors of dimension N, the weights calculated by
%  WGT3,WGT4     'barie1'
%
% N              Integer, number of line segments
%
% X0,Y0          Real, coordinates of the source point
%
% X,Y            Real, cordinates of some point in the medium
%
% BETAC          Complex, the admitance of the flat ground on which the
%                barrier sits
%
% Outputs:
%
% BARIE2         Complex, the pressure at the field point
%
```

```
% This function calls functions:
%       hnkl10,pbeta,layers, and pbetad
%

%
% First calculate the pressure if the barrier were absent
%
XDIF = X - X0;
YDIF = Y - Y0;
YSUM = Y + Y0;
XDIFS = XDIF*XDIF;
RDIR = sqrt(XDIFS+YDIF*YDIF);
RREF = sqrt(XDIFS+YSUM*YSUM);
PR = -0.25i*( hnkl10(RDIR) + hnkl10(RREF) ) + pbeta(XDIF,YSUM,BETAC);
%
% Now add the correction for the barrier(a quadrature over the barrier
% surface)
%
for j = 1:N
    PJ=0.0;
    [SINGLE DOUBLE] = layers(X,Y,XM(j),YM(j),UNORMX(j),UNORMY(j),H(j));
    PJ = PJ + DOUBLE + WGT1(j)*SINGLE;

    [SINGLE DOUBLE] = layers(X,Y,XM(j),-YM(j),UNORMX(j),-UNORMY(j),H(j));
    PJ = PJ + DOUBLE + WGT2(j)*SINGLE;
    [PBETA1 PBETAX PBETAY] = pbetad(XM(j)-X,YM(j)+Y,BETAC);
    PJ = PJ + WGT3(j)*PBETA1 + WGT4(j)*PBETAX;
    PR = PR + PJ*P(j);
end
BARIE2 = PR;
%
```

Appendix C

```
function [single,double] = layers(x,y,xm,ym,unormx,unormy,h)
%
% function [single,double] = layers(x,y,xm,ym,unormx,unormy,h)
%
% Converted from FORTRAN Proplib routine.
%
% Layers calculates approximations, using a combination of analytic
% integration and the product midpoint rule, to the potential at a
% point (X,Y) due to single and double layer potentials, with constant
% unit density, and support, a given line segment.
%
% Note that dimensionless coordinates and lengths are used throughout,
% each length or coordinate made dimensionless by multiplying by the
% wavenumber = (2 * pi / wavelength) of medium propagation.
%
% The kernel function of the single layer potential is
% -(i/4) * hnkl10( dist(R,RS) )
% where i = sqrt(-1), and dist(R,RS) is the disatnce between the
% field point R and the variable point of integration RS.
%
% Input:
%
% x,y       Real
%           (x,y) are the coordinates at which the values of the
%           potentials are to be calculated
%
% xm,ym     Real
%           (xm,ym) are the coordinates of the midpoint of the line
%           segment
%
% unormx,   Real
%  unormy   (unormx,unormy) are the components of the unit normal to
%           the line segment.
%
% h         Real
%           The length of the line segment
```

```
%
%
% Output:
%
% single    Complex
%           The approximate value of the single layer potential at (x,y)
%
% double    Complex
%           The approximate value of the single layer potential at (x,y)
%
%
% This function calls functions:
%            hnkin3, hnksb1
%

%
con=0.5/pi;
%
% Firts calculate the coords (xx,yy) of the point (x,y) realtive to the
% rotated and displaced coordinate axes, which have origin at (xm,ym),and
% yy-axis in the opposite direction to that of the unit normal.
xdif = x - xm;
ydif = y - ym;
xx = -xdif*unormy + ydif*unormx;
yy = -ydif*unormy - xdif*unormx;
%
% Now calculate approximationa tot eh single and double layer potentials
%
single = complex(0.0,-0.25*h)*hnkin3(abs(xx),abs(yy),h);
double = complex(0.0,0.0);
if(abs(yy) > 1.0E-08)
    dist = sqrt(xx*xx+yy*yy);
    hh = 0.5*h;
    double = complex(0.0,0.25*h*yy/dist)*hnksb1(dist) + ...
    con*( atan((xx+hh)/yy) - atan((xx-hh)/yy) );
end

function PBETA = pbeta(XX,YY,B)
%
% function PBETA = pbeta(XX,YY,B)
%
% Converted from FORTRAN Proplib routine.
%
% Calculates the function pbeta, used to calculate the acoustic potential
% on reflection of a cylindrical wave at a plane of homogeneous admittance.
% An expression for pbeta as a Laplace-type integral is used if the
% condition
%           abs(1-B) < 0.1
% is satisfied. Otherwise a representation as the sum of a complex argument
```

```
% is used.
%
%      Y  1
%         1
%         1
%         1 (X0,Y0) * HARMONIC LINE SOURCE
%         1
%         1
%         1                        (X1,Y1)  *  RECEIVER
%         1
%         1     GROUND OF ADMITTANCE B
%  -------0-------------------------------------------->
%                                                  X
%
%
% Inputs:
%
%       (Where K = Wavenumber = 2*pi/Wavelength, and (X0,Y0), (X1,Y1),
%       are the coordinates of the source and receiver respectively)
%

% XX     real scalar or vector
%           where XX = ( X1 - X0 ) * K.
%
% YY     non-negative scalar
%           where YY = ( Y1 + Y0 ) * K.
%
% B      complex scalar
%           B is the admitance of the ground surface. Time dependance
%           exp(-i*W*T) assumed.
%
% Output:
%
% PBETA  complex vector of same size as XX
%           PBETA  = (Acoustic potential) - (Acoustic potential when ground rigid)
%           ie Acoustic potential = -(i/4)*hnkl10(R1) - hnkl10(R2) + PBETA
%           Where
%           R1 = K * sqrt( (X1-X0)^2 + (Y1-Y0)^2 ),
%           R2 = K * sqrt( (X1-X0)^2 + (Y1+Y0)^2 ),
%           hnkl10 = Hankel Function of the first kind order zero.
%
% This function calls functions:
%       was
%
%
% The integral of H(T)*exp(-T)/sqrt(T) where H is F or G, is calculated
% below by the generalised Gauss-Laguerre Quadrature
%
% First assign the abscissae and weights of the Gauss-Laguerre rule.
```

```
% A 40-point rule for weight function t^(-1/2) exp(-t) is used, but the
% last 18 weights are discarded since they have sum < 1.9E-15.
%
X = [.0153256633315, .137966001741, .3834338413928, .7521050835315, ...
    1.244547551113, 1.861525845317, 2.604007976597, 3.473173911353, ...
    4.470426220642, 5.597403070341, 6.85599385527, 8.24835785637, ...
    9.776946394185, 11.44452906932, 13.2542248286, 15.20953878472, ...
    17.31440596067, 19.57324344853, 21.99101289126, 24.57329575658, ...
    27.32638462934, 30.25739478733 ];
%
WEIGHT =[.4876717076145, .4315498925489, .3378759385518, .2339614676086, ...
    .1432014953837, .07741719982969, .03693150230886, .0155280788108, ...
    .005746395066181, .001868635373923, .0005329557733869, .0001330348248363, ...
    .00002899288838227, .000005501449254803, 9.061071131725E-7, 1.290892590025E-7, ...
    1.584584657831E-8, 1.668611185219E-9, 1.499943982933E-10, 1.144655943702E-11,...
    7.369699640332E-13, 3.975041669405E-14 ];
%
PBETA = zeros(size(XX));
o = ones(size(XX));
if B ~= 0
   RREF = sqrt(XX.^2+YY^2);
   %
   % First evaluate the PBETA values for which RREF is very small,
   % approximating them by the value for RREF = 0.
   %
   RREF_small = RREF < 1e-8;
   if any(RREF_small)
      Ones = o(RREF_small);
      if B == 1
         Q = 1/pi;
         PBETA(RREF_small) = Q(Ones);
      else
         Q = i*sqrt(1-B^2);
         Q = -(0.5/pi)*B*log( (B-Q)/(B+Q) )/Q;
         PBETA(RREF_small) = Q(Ones);
      end
   end
   %
   % Next evaluate the PBETA values for which RREF is larger.
   %
   RREF_larger = ~RREF_small;
   if any(RREF_larger)
      RREF = RREF(RREF_larger);
      CTH = YY./RREF;
      STH = sqrt(1-CTH.^2);
      RREFI = i*RREF;
      C1 = RREF.*(CTH+B);
      C2 = i*CTH;
      C3 = RREFI + RREFI;
```

```
      if  abs(1-B) < 0.1
         C4 = C3.*(1+CTH*B);
         C5 = C1.*C1;
         %
         % Now calculate an approximation, by Gauss-Laguerre Quadrature, to
         % the integral of  F(X)*exp(-X)/sqrt(X)
         %
         clear CTH STH
         INTGRL = 0;
         for j = 1:22
            INTGRL = INTGRL + WEIGHT(j)*f(X(j),C1,C2,C3,C4,C5);
         end
         PBETA(RREF_larger) = -(RREF/pi)*B.*exp(RREFI).*INTGRL;
      else
         Q = sqrt(1-B^2);
         APLUS = 1 + B*CTH - Q*STH;
         AMINUS = 1 + B*CTH + Q*STH;
         WSQ = RREFI.*APLUS;
         W = sqrt(RREFI) .* sqrt(APLUS);
         %
         % Note that  -pi/4  <=  ARG(W)  <=  3*pi/4
         %
         C4 = RREFI.*AMINUS;
         C5 = W./(C3*Q);
         %
         % Now calculate an approximation, by Gauss-Laguerre Quadrature, to
         % the integral of  G(X)*exp(-X)/sqrt(X)
         %
         clear APLUS AMINUS CTH STH
         INTGRL = 0;
         for j = 1:22
            INTGRL = INTGRL + WEIGHT(j)*g(X(j),C1,C2,C3,C4,C5,WSQ);
         end
         PBETA(RREF_larger) = B*exp(RREFI).*( (0.5/Q)*wvect(W) - (RREF/pi).*INTGRL );
      end
   end
end
%
% sub-function g
%
function G = g(T,C1,C2,C3,C4,C5,WSQ)
%
G = ((C1+C2*T)./(sqrt(T-C3).*(T-C4))+C5)./(T-WSQ);
clear C1 C2 C3 C4 C5 WSQ
%
% sub-function f
%
function F = f(T,C1,C2,C3,C4,C5)
%
```

```
F = (C1+C2*T)./(sqrt(T-C3).*(T*T-C4*T-C5));
clear C1 C2 C3 C4 C5

function[PBETA,PBETAX,PBETAY] = pbetad(XX,YY,B)
%
% function[PBETA,PBETAX,PBETAY] = pbetad(XX,YY,B)
%
% Converted from FORTRAN Proplib routine.
%
% Calculates the function pbeta, used to calculate the acoustic potential
% on reflection of a cylindrical wave at a plane of homogeneous admittance.
% An expression for pbeta as a Laplace-type integral is used if the
% condition
%           abs(1-B) < 0.1
% is satisfied. Otherwise a representation as the sum of a complex argument
% is used.
%
%      Y  1
%         1
%         1
%         1 (X0,Y0) * HARMONIC LINE SOURCE
%         1
%         1
%         1                       (X1,Y1)  *  RECEIVER
%         1
%         1     GROUND OF ADMITTANCE B
%  -------0------------------------------------------->
%                                                  X
%
%
%
% Inputs:
%
%       (Where K = Wavenumber = 2*pi/Wavelength, and (X0,Y0), (X1,Y1),
%       are the coordinates of the source and receiver respectively)
%

% XX     real scalar or vector
%           where XX = ( X1 - X0 ) * K.
%
% YY     non-negative scalar
%           where YY = ( Y1 + Y0 ) * K.
%
% B      complex scalar
%           B is the admitance of the ground surface. Time dependance
%           exp(-i*W*T) assumed.
%
% Output:
%
```

```
% PBETA  complex vector of same size as XX
%           PBETA  = (Acoustic potential) - (Acoustic potential when ground rigid)
%           ie Acoustic potential = -(i/4)*hnkl10(R1) - hnkl10(R2) + PBETA
%           Where
%           R1 = K * sqrt( (X1-X0)^2 + (Y1-Y0)^2 ),
%           R2 = K * sqrt( (X1-X0)^2 + (Y1+Y0)^2 ),
%           hnkl10 = Hankel Function of the first kind order zero.
%
% PBETAX  partial derivative with respect to XX of PBETA(XX,YY,B)
%
% PBETAY  partial derivative with respect to YY of PBETA(XX,YY,B)
%
%  Note
%      D(PBETA(K*(X1-X0))) / D(X1)  =  K * PBETAX
%      D(PBETA(K*(Y1+Y0))) / D(Y1)  =  K * PBETAY
%
% This function calls functions:
%       was,hnkl10
%
%      PARAMETER(NABSC=22)
%      REAL X(NABSC),WEIGHT(NABSC)
%
% NABSC = Number of Abscissae used in the Gauss-Laguerre Quadrature
%
% Note that the 40 point Gauss rule is used, but the last 18 Abscissae are
% neglected, having combined weights less than 1.9E-15
%
%
% Below are Absissae and the weights for the integration of
% G(X)*exp(-X)/sqrt(X) by the generalised Gauss-Laguerre Quadrature
%
NABSC = 22;
X =     [.0153256633315 .137966001741 .3834338413928 .7521050835315 ...
        1.244547551113 1.861525845317 2.604007976597 3.473173911353 ...
        4.470426220642 5.597403070341 6.85599385527 8.24835785637 ...
        9.776946394185 11.44452906932 13.2542248286 15.20953878472 ...
        17.31440596067 19.57324344853 21.99101289126 24.57329575658 ...
        27.32638462934 30.25739478733];
%
WEIGHT = [.4876717076145 .4315498925489 .3378759385518 .2339614676086 ...
        .1432014953837 .07741719982969 .03693150230886 .0155280788108 ...
        .005746395066181 .001868635373923 .0005329557733869 .0001330348248363 ...
        .00002899288838227 .000005501449254803 9.061071131725E-7, 1.290892590025E-7 ...
        1.584584657831E-8 1.668611185219E-9 1.499943982933E-10 1.144655943702E-11 ...
        7.369699640332E-13 3.975041669405E-14];
%;
PBETA = 0.0;
PBETAX = 0.0;
PBETAY = 0.0;
```

```
if (B ~= 0.0)
    RREF = sqrt(XX*XX+YY*YY);
    if ( RREF < 1.0E-08 )
        if ( B == 1.0 )
            PBETA = 1.0/pi;
        else
            Q = 1.0*sqrt(1.0-B*B);
            PBETA = -(0.5/pi)*B*log( (B-Q)/(B+Q) )/Q;
        end
    else
        CTH = YY/RREF;
        STH = sqrt(1.0-CTH*CTH);
        RREFI = RREF*i;
        C1 = RREF*(CTH+B);
        C2 = CTH*i;
        C3 = RREFI + RREFI;
        if ( abs(1.0-B) < 0.1 )
            C4 = C3*(1.0 + CTH*B);
            C5 = C1*C1;
            C6 = i*B;
%
% Now calculate the approximations to two integrals by Gauss-Laguerre
% Quadrature
%
               INTGR1 = 0;
               INTGR2 = 0;
               for j = 1:NABSC
                  T1 = sqrt(X(j)-C3)*(X(j)*X(j) - X(j)*C4 - C5);
                  G1 = (C1 + X(j)*C2)/T1;
                  G2 = (C1 + X(j)*C6)/T1;
                  INTGR1 = INTGR1 + WEIGHT(j)*G1;
                  INTGR2 = INTGR2 + WEIGHT(j)*G2;
              end
%
               T1 = -(RREF/PI)*B*EXP(RREFI);
               PBETA = T1*INTGR1;
               if ( XX < 0 )
               PBETAX = (-i*STH)*T1*INTGR2;
               else
               PBETAX = (i*STH)*T1*INTGR2;
               end
               PBETAY = B*( -i*PBETA - 0.5*hnkl10(RREF) );
         else
               Q = sqrt(1.0-B*B);
               APLUS = 1.0 + B*CTH - Q*STH;
               AMINUS = 1.0 + B*CTH + Q*STH;
               WSQ = RREFI*APLUS;
               W = sqrt(RREFI) * sqrt(APLUS);
% Note that -pi/4  <  arg(W)  <  3*pi/4
```

```
              C4 = RREFI*AMINUS;
              C5 = W/(C3*Q);
              C6 = STH*C1;
              C7 = i*STH*B;
              C8 = Q*C5;
%
% Now calculate the approximations to two integrals by Gauss-Laguerre
% Quadrature
%
              INTGR1 = 0;
              INTGR2 = 0;
              for j = 1:NABSC
                 T1 = sqrt(X(j)-C3)*(X(j)-C4);
                 T2 = X(j) - WSQ;
                 G1 = ((C1 + X(j)*C2)/T1 + C5)/T2;
                 G2 = ((C6 + X(j)*C7)/T1 + C8)/T2;
                 INTGR1 = INTGR1 + WEIGHT(j)*G1;
                 INTGR2 = INTGR2 + WEIGHT(j)*G2;
               end
%
               T1 = B*exp(RREFI);
               T2 = 0.5*was(W);
               S1 = RREF/pi;
               PBETA = T1*(T2/Q - S1*INTGR1);
               if ( XX < 0 )
               PBETAX = -i*T1*(T2 - S1*INTGR2);
               else
               PBETAX = i*T1*(T2 - S1*INTGR2);
               end
               PBETAY = B * ( -i*PBETA - 0.5*hnkl10(RREF) );
          end
      end
  end

%
```

Appendix D

```
function HNKIN3 = hnkin3(X,Z,H)
%
% function HNKIN3 = hnkin3(X,Z,H)
%
% THIS FUNCTION EVALUATES THE INTEGRAL
%
%                             X+0.5*H
%    HNKIN3  = ( L1(R) * 0.5 * INT   LOG(S*S+Z*Z)*DS  + H*L2(R) ) / H
%                            X-0.5*H
%
% WHERE             R = SQRT(X*X+Z*Z)
%
% AND L1,L2 ARE THE UNIQUE ENTIRE FUNCTIONS DEFINED BY
%
%    HNKL10(X) = L1(X)*LOG(X) + L2(X),  FOR X > 0,
%
% WITH HNKL10 THE HANKEL FUNCTION OF THE FIRST KIND OF ORDER ZERO.
%
% HNKIN3 CAN BE REGARDED AS AN APPROXIMATION TO THE MEAN VALUE OF
% HNKL10(SQRT(Z*Z+T*T)) IN -0.5*H+X <= T <= X+0.5*H, I.E.,
%
%                X+0.5*H
%    HNKINT  = ( INT    HNKL10(SQRT(Z*Z+S*S))*DS ) / H .
%               X-0.5*H
%
% THE RELATIVE ERROR ABS((HNKINT-HNKIN3)/HNKINT) IS OF ORDER O(H*H), AND
% THE ABSOLUTE ERROR ABS(HNKINT-HNKIN3) IS OF ORDER O(H*H*LOG(H)),
% AS H TENDS TO ZERO, UNIFORMLY FOR X >= 0 AND Z >= 0.
%
% X MUST BE > 0, AND H > 0 ON ENTRY. IF Z < 0 ON
% ENTRY THEN Z = 0 IS ASSUMED.
%
% FUNCTION HNKIN2 IS CALLED.
%
%
  C1 = 3;
```

```
  C2 = 300;
%
% C1 AND C2 ARE CONSTANTS WHICH DETERMINE THE REGION FOR EACH
% APPROXIMATION. THE VALUE OF C2 DEPENDS ON THE NUMBER OF
% SIGNIFICANT FIGURES USED IN SINGLE PRECISION FLOATING POINT
% ARITHMETIC.
%
      if Z <= 0
         HNKIN3 = hnkin2(X,H);
      else
         ZSQ = Z^2;
         XSQ = X.^2;
         RSQ = ZSQ + XSQ;
         R4 = RSQ.*RSQ;
         R = sqrt(RSQ);
         HNK = besselj(0,R)+i*bessely(0,R);
         L1 = i*(2/pi)*besselj(0,R);
         L2 = HNK - log(R).*L1;
         if R >= C2*H
%
%           USE THE FIRST TWO TERMS, UP TO O( H*(R/H)**(-2) ), OF
%           THE TAYLOR SERIES FOR THE INTEGRAL, IF R/H IS LARGE.
%
            HNKIN3 = HNK + L1.*(H*H*(ZSQ-XSQ)./(24*R4));
         else
%
%           OTHERWISE USE THE EXACT EXPRESSION FOR THE INTEGRAL, BUT
%           CARE MUST BE TAKEN, AND C2 CHOSEN APPROPRIATELY, TO AVOID
%           ROUNDING ERROR.
%
            HH = 0.5*H;
            T = X + HH;
            B = X - HH;
            RT = T.*T + ZSQ;
            RB = B.*B + ZSQ;
            if B <= 0
               TERM = -H + Z*(atan(T/Z)-atan(B/Z));
            else
%
%              IF B .GT. 0, A MORE ACCURATE AND QUICKER EXPRESSION
%              FOR THE DIFFERENCE BETWEEN THE ATAN FUNCTIONS IS USED
%
               TERM = -H + Z*atan(H*Z./(ZSQ+T.*B));
            end
            if R <= C1*H
               TERM = 0.5*( T.*log(RT) - B.*log(RB) )  +  TERM;
               HNKIN3 = (TERM/H).*L1 + L2;
            else
               TERM = 0.5*X.*log(RT./RB) + 0.25*H*log(RT.*RB./R4) + TERM;
```

```
                HNKIN3 = HNK + (TERM/H).*L1;
             end
          end
       end
%
%
% sub-function  hnkin2

function HNKIN2 = hnkin2(X,H)
%
% function HNKIN2 = hnkin2(X,H)
%
% THIS FUNCTION EVALUATES THE INTEGRAL
%
%                         X+0.5*H
%    HNKIN2  = ( L1(X) *  INT    LOG(ABS(S))*DS   +  H * L2(X) ) / H
%                        X-0.5*H
%
% WHERE L1,L2 ARE THE UNIQUE ENTIRE FUNCTIONS DEFINED BY
%
%    HNKL10(X) = L1(X)*LOG(X) + L2(X),  FOR X .GT. 0,
%
% WITH HNKL10 THE HANKEL FUNCTION OF THE FIRST KIND OF ORDER ZERO.
%
% HNKIN2 CAN BE REGARDED AS AN APPROXIMATION TO THE MEAN VALUE OF
% HNKL10(ABS(T)) IN -0.5*H + X .LE. T .LE. X + 0.5*H, I.E.,
%
%                 X+0.5*H
%    HNKINT  = ( INT    HNKL10(ABS(S))*DS ) / H .
%                X-0.5*H
%
% THE RELATIVE ERROR ABS((HNKINT-HNKIN2)/HNKINT) IS OF ORDER O(H*H), AND
% THE ABSOLUTE ERROR ABS(HNKINT-HNKIN2) IS OF ORDER O(H*H*LOG(H)),
% AS H TENDS TO ZERO, UNIFORMLY FOR X.GE.0.0.
%
% X,H ARE REAL. X MUST BE .GE. 0.0, AND H .GT. 0.0 ON ENTRY.
% X,H ARE UNCHANGED ON EXIT.
%
% PROPLIB ROUTINE HNKANA IS CALLED.
%
  CON1 = 2/pi;
  GAMMA = .5772156649015328606;
  CON2 = 2*log(2) + 1 - GAMMA;
%
  for j = 1:5
   C(j) = 1/(2*j*(2*j+1));
  end
      HH = 0.5*H;
      HINV = 1/H;
```

```
      if X > 3*H
%        IN THIS CASE USE
%           HNKIN2 = BESSELJ(0,X) + i*BESSELY(0,X) -
%                    I*CON1*BESSELJ(0,X)*( C(1)*Z + C(2)*Z**2 ... + C(5)*Z**5)
%        WHERE I = SQRT(-1), Z = (0.5*H/X)^2.
         [L1,L2] = hnkana(X);
         Z = 0.5*H./X;
         Z = Z.*Z;
         SUM = ((((C(5)*Z+C(4))*Z+C(3))*Z+C(2))*Z+C(1))*Z;
         HNKIN2 = besselj(0,X)+i*bessely(0,X) - L1*SUM;
      else
%        OTHERWISE USE THE FORMULA
%        HNKIN2 = L1(X) * ( T*(LOG(T)-1) - B*(LOG(ABS(B))-1) )/H + L2(X)
%        WITH MODIFICATION IF B .EQ. 0.0. (B,T DEFINED BELOW)
         B = X - HH;
         T = X + HH;
         if X>H
            LGB = log(B);
            LGT = log(T);
            [L1,L2] = hnkana(X);
            HNKIN2 = L1.*(HINV.*(X.*(LGT-LGB) + HH*(LGT+LGB) - H)) + L2;
         elseif X == 0
            HNKIN2 = 1+i*(CON1.*(log(H)-CON2));
         elseif B == 0
            [L1,L2] = hnkana(X);
            HNKIN2 = L1.*(log(H)-1) + L2;
         else
            [L1,L2] = hnkana(X);
            LGB = log(abs(B));
            LGT = log(T);
            HNKIN2 = L1*(HINV*(T*(LGT-1) - B*(LGB-1))) + L2;
         end
      end



function [L1,L2] = hnkana(X)
%
% function [L1,L2] = hnkana(X)
%
% WHERE HNKL10(X) IS THE HANKEL FUNCTION OF THE FIRST KIND OF ORDER ZERO,
% THIS FUNCTION CALCULATES THE COMPLEX-VALUED ENTIRE FUNCTIONS
%   L1(X) = I*(2/PI)*J0(X),
%   L2(X) = HNKL10(X) - LOG(X)*L1(X),
% WHERE I = SQRT(-1) AND J0 IS THE BESSEL FUNCTION OF ORDER ZERO.
%
% ON ENTRY
% X = POSITIVE REAL NUMBER
%
```

```
% NOTE THAT   HNKL10(X) = L1(X)*LOG(X) + L2(X),  AND THAT L1,L2 ARE
% ENTIRE FUNCTIONS BOTH OF THEIR ARGUMENTS AND OF THE SQUARE OF
% THEIR ARGUMENTS.
%
L1 = i*(2/pi)*besselj(0,X);
HNK = besselj(0,X)+i*bessely(0,X);
L2 = HNK - log(X).*L1; % This will have precision problems for x very small

function h1 = hnksb1(x)
%
% hnksb1(x) is an approximation to
%
% h1(x) = hnkl11(x) + 2*i/(pi*x)
%
% where hnkl11 is the principal of the first kind of order one, and i is
% sqrt(-1)
%
% THE APPROXIMATION USED for x < 0.1 is given in Abramowitz and Stegun
% 9.1.11, New Yorkdover 1973.
if x == 0
    h1 = 0;
elseif x < 0.1
    a = zeros(1,5);
    for k = 0:4
        a(k+1) = (psi(k+1)+psi(k+2))/(factorial(k)*factorial(k+1));
    end
    sum = 0.0;
    z = -x^2/4;
%    for k = 0:4
%        sum = sum + a(k+1)*z^k;
%    end
    sum = (((a(5)*z + a(4))*z + a(3))*z + a(2))*z + a(1);
    j = besselj(1,x);
    y = 2/pi*log(x/2)*j-x/(2*pi)*sum;
    h1 = complex(j,y);
else
    h1 = besselh(1,x) + 2*i/(pi*x);
end



function y = hnkl10(x)
%
% function y = hnkl10(x)
%
% This function calculates the Hankel function of the first
% kind of order zero.
%
% Input
```

```
%
% x   real scalar or vector
%
% y   complex scalar or vector containing the values of the
%     Hankel function at x
%
y = besselj(0,x)+i*bessely(0,x);
```

Appendix E

```
function  [XM,YM,UX,UY,H,NTOTAL,N]  = bari67(HCON,X,Y,NSEC1,EXIST)
%
% function  [XM,YM,UX,UY,H,NTOTAL,N]  = bari67(HCON,X,Y,NSEC1,EXIST)
%
% bari67 replace barie7, bari6a and the intervening code in bari8c
% to produce a graded mesh, where q is the severity of the mesh grading.
%
q=2.5;
HNORM=2*pi*HCON;
count=0;

n1=fix(1/(1-(1-(HNORM/(2*pi)))^(1/q)));
for i = 1:(NSEC1-1)
    N(i)=0
    if (EXIST(i))

       theta = atan((Y(i+1)-Y(i))/(X(i+1)-X(i)));
       xtotlen=(X(i+1)-X(i));
       ytotlen=(Y(i+1)-Y(i));
       UNX = -ytotlen/HCON;
       UNY = xtotlen/HCON;

       if(X(i+1)-X(i) > 4*pi)
        xgradlen=2*pi*cos(theta);
        xinternal=xtotlen-2*xgradlen;

        for j=1:n1+1
            sx(j)=xgradlen*((j-1)/n1)^q;
        end

        tempsx=sx;
        sx(1)=sx(2)/2;
        for j=2:n1
            sx(j)=sx(j)+(sx(j+1)-sx(j))/2;
        end
```

```
    %grading the region to the right of x(i)
    for j=1:n1
        XM(count+j) = X(i)+sx(j);
        YM(count+j) = XM(count+j)*tan(theta);
        elelen(count+j)=(tempsx(j+1)-tempsx(j))/cos(theta);
        normelex(count+j)=UNY;
        normeley(count+j)=UNX;
    end
    count=count+n1;
    N(i)=N(i)+n1;

    %grading the internal points equally
    no_intx=fix(xinternal/(HNORM*cos(theta)));
    no_intx
    sizeintx=xinternal/no_intx;
    sizeintx*no_intx

    XM(count+1)=X(i)+xgradlen +sizeintx/2;
    YM(count+1) = XM(count+1)*tan(theta);
    elelen(count+1)=sizeintx/cos(theta);
    for j=2:no_intx

        XM(count+j)=XM(count+j-1)+sizeintx;
        YM(count+j) = XM(count+j)*tan(theta);
        elelen(count+j)=sizeintx/cos(theta);
        normelex(count+j)=UNY;
        normeley(count+j)=UNX;
    end
    count=count+no_intx;
    N(i)=N(i)+no_intx;

    %grading the region to the left of x(i+1)
    for j=1:n1

        XM(count+j) = X(i+1)-sx(n1-j+1);
        YM(count+j) = XM(count+j)*tan(theta);
        elelen(count+j)=(tempsx(n1-j+2)-tempsx(n1-j+1))/cos(theta);
        normelex(count+j)=UNY;
        normeley(count+j)=UNX;
    end
    count=count+n1;
    N(i)=N(i)+n1;

else

    xgradlen=(X(i+1)-X(i))/2;

    for j=1:n1+1
        sx(j)=xgradlen*((j-1)/n1)^q;
```

```
        end
        tempsx=sx;
        sx(1)=sx(2)/2;
        for j=2:n1
            sx(j)=sx(j)+(sx(j+1)-sx(j))/2;
        end

        %grading the region to the right of x(i)
        for j=1:n1
            XM(count+j) = X(i)+sx(j);
            YM(count+j) = XM(count+j)*tan(theta);
            elelen(count+j)=(tempsx(j+1)-tempsx(j))/cos(theta);
            normelex(count+j)=UNY;
            normeley(count+j)=UNX;
        end
        count=count+n1;
        N(i)=N(i)+n1;


        %grading the region to the left of x(i+1)
        for j=1:n1

            XM(count+j) = X(i+1)-sx(n1-j+1);
            YM(count+j) = XM(count+j)*tan(theta);
            elelen(count+j)=(tempsx(n1-j+2)-tempsx(n1-j+1))/cos(theta);
            normelex(count+j)=UNY;
            normeley(count+j)=UNX;
        end
        count=count+n1;
        N(i)=N(i)+n1;

    end%end of if less than 4 pi


    end %end of if exist


end %end of for loop 1


UX = normelex;
UY = normeley;
H = elelen;
NTOTAL = count;
```

Appendix F

Sample Input and Output files

The TINPUT file contains the necessary geometrical data to define the cross sectional model used in the BEM.

```
4                 % Number of corners
15.1   0          % Coordinates of
15.1   3          % corners (in m).
14.9   3
14.9   0
0                 % model
1  300000. 0.0 0.5 1.8 0.   % surface properties
1  300000. 0.0 0.5 1.8 0.   % between corners
1  300000. 0.0 0.5 1.8 0.
300000.  0.0  0.5 0.5 0.    % ground properties
35 0              % source coordinates
1                 % number of reciever positions
0 0               % reciever coordinates
```

The TINPUTSP file contains the spectral data used with the program.

```
17 1                    % Number of frequncies & spectra
100.0 0.1 56.9
125.893 0.035 56.3      % Frequency (in Hz), element
158.489 0.04 57.5       % length as a fraction of a
199.25 0.45 62.1        % wavelength, and SPL at
251.189 0.05 63.3       % 1.0m from the source
316.228 0.06 66.0
398.107 0.07 68.0
501.187 0.08 71.0
630.957 0.09 75.8
794.328 0.1 80.1
1000. 0.125 83.9
1258.925 0.125 80.8
```

```
1584.893 0.125 78.3
1995.262 0.125 75.2
2511.886 0.125 73.1
3162.278 0.16 69.7
3981.072 0.2 65.8
```

The 'barrier_output' file contains of the data and the results for each frequency in turn.

```
 OUTPUT FILE FOR OUTDOOR SOUND PROPAGATION OVER BARRIER

 THE FOLLOWING ELEMENT LENGTHS ARE USED IN THE BOUNDARY
 ELEMENT METHOD.

   FREQUENCY (HZ)   ELEMENT LENGTH (WAVELENGTHS)
   --------------   ----------------------------

       100.000                 0.1000
       125.893                 0.0350
       158.489                 0.0400
       199.250                 0.4500
       251.189                 0.0500
       316.228                 0.0600
       398.107                 0.0700
       501.187                 0.0800
       630.957                 0.0900
       794.328                 0.1000
      1000.000                 0.1250
      1258.925                 0.1250
      1584.893                 0.1250
      1995.262                 0.1250
      2511.886                 0.1250
      3162.278                 0.1600
      3981.072                 0.2000


SOURCE COORDINATES:-
     X-COORDINATE IS          35.00
     Y-COORDINATE IS           0.00

GROUND AND BARRIER ELEMENTS ARE MODELLED USING THE
ATTENBOROUGH MODEL

 BARRIER CORNER COORDINATES:

   CORNER NO.    X/METRES   Y/METRES
   ----------    --------   --------
```

```
       1          15.10       0.00
       2          15.10       3.00
       3          14.90       3.00
       4          14.90       0.00


POSITIONS AND SURFACE PROPERTIES OF BARRIER SIDES:
               SIDE PRESENT       EFFECTIVE    POROUS LAYER                     DOMINANT
               BETWEEN THIS   FLOW RESISTIVITY    DEPTH                         ANGLE OF
   SIDE NOS. PAIR OF CORNERS?    (SI UNITS)      (METRES)   POROSITY TORTUOSITY INCIDENCE
 ----------- ---------------- ---------------- ------------ -------- ---------- ---------
       1             1            300000            0          0.5       1.8        0
       2             1            300000            0          0.5       1.8        0
       3             1            300000            0          0.5       1.8        0


         SOURCE SPECTRUM NUMBER  1
         -------------------------

 THIS SPECTRUM IS THE SINGLE FREQUENCY   100 HZ.

 RECEIVER   X-COORD. OF  Y-COORD. OF
 NUMBER     RECEIVER/M   RECEIVER/M    SPL/DB    EA/DB   IL/DB
 --------   -----------  -----------   ------    -----   ------

     1           0.00         0.00   41.90998 -0.45052  5.57008


         SOURCE SPECTRUM NUMBER  2
         -------------------------

 THIS SPECTRUM IS THE SINGLE FREQUENCY   126 HZ.

 RECEIVER   X-COORD. OF  Y-COORD. OF
 NUMBER     RECEIVER/M   RECEIVER/M    SPL/DB    EA/DB   IL/DB
 --------   -----------  -----------   ------    -----   ------

     1           0.00         0.00   41.36731 -0.50782  5.51278

.
.
.
.
.
.


         SOURCE SPECTRUM NUMBER 17
```

```
        -------------------------

THIS SPECTRUM IS THE SINGLE FREQUENCY  3981 HZ.

RECEIVER   X-COORD. OF  Y-COORD. OF
NUMBER     RECEIVER/M   RECEIVER/M    SPL/DB    EA/DB   IL/DB
--------   -----------  -----------   ------    -----   ------

    1          0.00         0.00   36.06699 14.29234 20.31294


        SOURCE SPECTRUM NUMBER OF COMBINED FREQUENCIES
        ----------------------------------------------

THIS SPECTRUM IS SPECTRUM AS SUPPLIED IN FILE
RECEIVER   X-COORD. OF  Y-COORD. OF
NUMBER     RECEIVER/M   RECEIVER/M    SPL/DB    EA/DB   IL/DB
--------   -----------  -----------   ------    -----   ------

    1          0.00         0.00   64.93366  7.87626 13.89686
```