isvr Institute of Sound and Vibration Research

BEASTS – A Computer Program for Boundary Element Analysis of Soil and Three-dimensional Structures

L. Andersen and C.J.C. Jones

ISVR Technical Memorandum 868

August 2001

University of Southampton

SCIENTIFIC PUBLICATIONS BY THE ISVR

Technical Reports are published to promote timely dissemination of research results by ISVR personnel. This medium permits more detailed presentation than is usually acceptable for scientific journals. Responsibility for both the content and any opinions expressed rests entirely with the author(s).

Technical Memoranda are produced to enable the early or preliminary release of information by ISVR personnel where such release is deemed to the appropriate. Information contained in these memoranda may be incomplete, or form part of a continuing programme; this should be borne in mind when using or quoting from these documents.

Contract Reports are produced to record the results of scientific work carried out for sponsors, under contract. The ISVR treats these reports as confidential to sponsors and does not make them available for general circulation. Individual sponsors may, however, authorize subsequent release of the material.

UNIVERSITY OF SOUTHAMPTON
INSTITUTE OF SOUND AND VIBRATION RESEARCH
DYNAMICS GROUP

BEASTS - A Computer Program for Boundary Element Analysis of Soil and Three-dimensional Structures

By

L. Andersen and C.J.C. Jones

ISVR Technical Memorandum No. 868

August 2001

Authorised for issue by
Dr. M.J. Brennan
Group Chairman

Summary

A boundary element (BE) computer program for analysing three-dimensional solid structures interacting with a layered sub-soil has been developed. The software is intended primarily to be used for the analysis of ground vibration due to railway traffic.

Several sub-domains may be specified within a single model. Coupling of the individual BE domains is established in the finite element sense by turning each of the domains into a finite element. This method is ideal for further implementation of any finite elements in the program, whether it may be solid elements, beam or shell elements.

The program provides a special way of treating open domains. The method is a development of a technique that has previously been developed for an open domain in two dimensions. Two types of elements are available in the program: a triangular element with quadratic interpolation and a quadrilateral element with biquadratic interpolation.

This report describes the software package and provides instructions for its use. A numerical example is presented where the developed program has been used for the analysis of a bored tunnel at a number of frequencies.

Contents

1 Introduction

When dealing with (semi) infinite structures or soil, an analytical solution is only possible for the most simple geometries. For problems with arbitrary geometries the *boundary element method* (BEM) is a powerful tool. The big advantage of the BEM, compared to the finite element method, is the inherent ability to satisfy the radiation conditions. Hence the BEM may be successfully used for such problems as ground vibration due to railway traffic, either on the surface of the ground or in a bored or cut-and-cover tunnel.

Previously a two-dimensional boundary element (BE) model has been produced for a medium consisting of a single domain [4]. For the analysis of more complicated geometries with multiple domains a computer program, TEA (Two-dimensional Elastodynamic Analysis), has been developed where several BE sub-domains can be included in a single model along with a number of different types of finite elements. The theory of the two-dimensional coupled BEM and FEM scheme is presented in [5] and documentation for the software package TEA is given in [6]. Many problems cannot be investigated using the two-dimensional model. A three-dimensional model has therefore been produced. The present report describes the computer program package BEASTS (Boundary Element Analysis of Soil and Three-dimensional Structures) which can be used for analysis of three-dimensional vibration wave propagation problems in the frequency domain. Instructions for the use of the software are given and, to illustrate the functionality of the program, a number of numerical examples are given.

The program is an extension of a BE program for a single, closed BE domain by J. Dominguez [3], whereby coupling of multiple sub-domains is performed according to the method described by O. von Estorff and E. Kausel [7]. The theory that has been implemented in the software is described in detail in a separate report [2].

2 Program Structure

In this section the structure of the BEASTS suite of programs is described. The source code is written in FORTRAN 77 and may be compiled on any system. Here, however, only an example of the program's use on an MSDOS or Windows 2000 system will be given. For UNIX the main batch file would have to be rewritten as a script. For visualization of the BE mesh and the results produced by BEASTS, graphic routines have been written in MATLAB. They are described in Section 4.

2.1 Program Flow and Files

The program package consists of two sub-programs: FLIES (Form Local Input for Each Subdomain), which is used first to process a single file of input data for a project, and ANTS (Assembly and Numerical Time-harmonic Solution), which is the main boundary element analysis program. The flow of the program is sketched in Figure 1. In the following paragraphs a short description will be given of each of the sub-programs, especially regarding the files that are used for input and are produced as output from the individual program units.

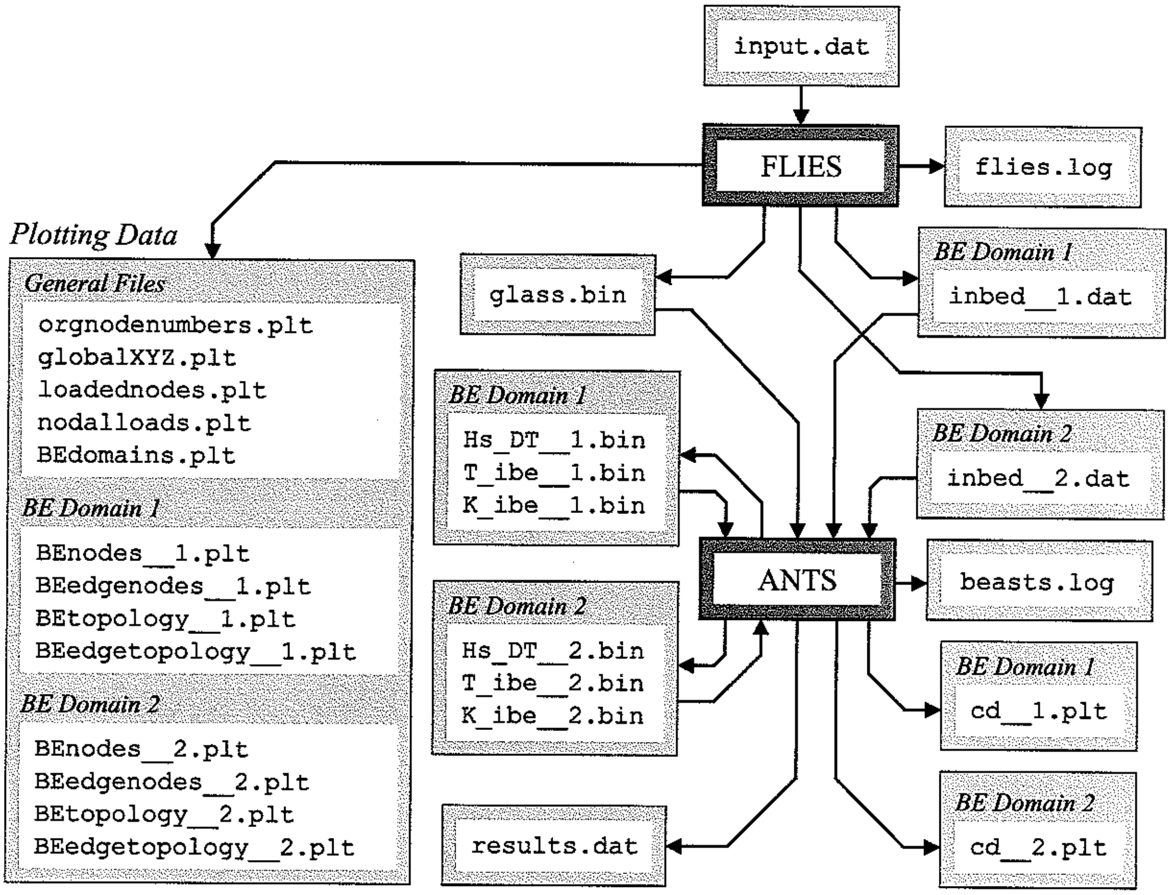

Figure 1. Flow diagram for FLIES and ANTS, here shown for a situation with two boundary element domains. File names appear in `fixed width` style.

2.1.1 FLIES - Form Local Input for Each Sub-domain

This program unit reads the global input data file and checks if there are any fatal errors in the input such as missing nodes or materials. A local node numbering scheme is established for the individual boundary element domains so that no redundant nodes are present. It then writes a separate input data file for each sub-domain in a readable text format similar to the global input file. Also, if a problem with geometrical symmetry around the plane described by $y = 0$ is analysed, extra so-called *edge elements* and *edge nodes* (see Section 3.10) may be constructed if necessary.

In addition to making the local input data files for each of the $n = 1..N$ sub-domains, FLIES also derives a conversion table between local nodes and global nodes which is used by ANTS when assembling the global system of equations. Moreover a number of files containing plotting information are saved for later use when visualising the mesh and plotting the results in another program, see Sections 4 and 5.

Files read by FLIES:

`input.dat`	This is a copy of the global input file created by the MSDOS batch program BEASTS in the working directory.

Files created by FLIES:

`flies.log`	Logging file written by FLIES when processing the input data. Error messages and warnings may appear in this file if some of the input was found to be inconsistent.
`glass.bin`	Global assembly file in binary format with conversion tables for node numbers between the respective sub-domains and the global system. The file also contains information about the loads (referring to global node numbers) to be used by ANTS.
`inbed_n.dat`	Local input data file in ASCII format for boundary element domain n. An input data file is created for each of the $n = 1..N$ boundary element domains. The specification style is analogous to that used in the global input file, but a local node numbering is used.
`orgnodenumbers.plt`	The global nodes are rearranged and new nodes may be created or redundant nodes erased by FLIES. For each global node in the final scheme this text file lists the order of appearance in the global input file. Both this and all succeeding plotting (`.plt`) files are in ASCII format and only contain numerical data.
`globalXYZ.plt`	The x, y and z coordinates for each of the global nodes in the rearranged scheme are listed in this file, one node per line.
`loadednodes.plt`	An updated list of the global nodes with loads applied to them.
`nodalloads.plt`	This file contains a list of the load amplitudes at each of the loaded nodes with one line containing the data for one node. Six numbers are given in each line being the real and imaginary part of the x components and subsequently the real and imaginary parts for the y and z components.
`BEdomains.plt`	This file only contains one number: the number of boundary element domains, N.
`BEnodes_n.plt`	The number of nodes in each element within sub-domain n is written to this file. Notice also that the element order is rearranged, so the order of appearance of elements in this file will generally differ from the order in the global input file.

`BEedgenodes_n.plt`	If sub-domain n is open, some of the elements will form a rim along the edge. The number of nodes in each of these *edge elements* is listed in this file.
`BEtopology_n.plt`	The boundary element topology for sub-domain n is written in this file with one line per element and one number in each line corresponding to one of the nodes. The topology data is given in terms of the rearranged global node numbering, not the local node numbering used for the individual sub-domains.
`BEedgetopology_n.plt`	This file contains the topology for any edge elements in sub-domain n. The style is the same as that of `BEtopology_n.plt`.

2.1.2 ANTS - Assembly and Numerical Time-harmonic Solution

This program runs the main analysis using the various input files produced by FLIES. Basically the program consists of two parts. In the first part the system matrices for each of the boundary element domains are derived separately and an equivalent finite element stiffness matrix is derived, cf. [2]. Subsequently, in the second part of ANTS, the individual domains are assembled in a global system and the complex amplitudes of displacement for each of the nodes are calculated.

Files read by ANTS:

`glass.bin`	The global assembly data file.
`inbed_n.dat`	The local input files for each of the BE domains.

Temporary files used by ANTS:

`Hs_DT_n.bin`	The first time a sub-domain is processed, the diagonal terms of the static part of the **H** matrix (see [2]) are determined and saved to this binary file. The results are later reused in the calculations for each of the frequencies.
`T_ibe_n.bin`	In order to transform the boundary element matrices into a format that is compatible with finite elements, a transformation matrix must be derived for each domain. This is carried out only once at the beginning of the program, and the matrix is stored in this binary file for later use.
`K_ibe_n.bin`	The original boundary element system matrices for sub-domain n are turned into an equivalent finite element stiffness matrix. To save memory while calculating the stiffness matrices for other BE domains, the stiffness matrix for BE domain n is temporarily stored in this file.

Files created by ANTS:

`ants.log` Information on the progress of the calculation is written to this logging file. If an error occurs it will be reported here.

`results.dat` The main results file gives information about the global node coordinates in the revised scheme and contains a list of the complex displacement amplitudes for each node for each frequency. A complete description of this, the main output file, is given in Section 5.

`cd_m.plt` This file contains the complex displacement amplitudes for one of the $m = 1..M$ frequencies. The purpose is the enable the user to load the output data into another program in an easy way when plotting the result. Notice that this file is only generated when the **multi file output* module is used, see Section 3.

2.2 Running BEASTS from MSDOS

The main MSDOS batch program BEASTS is called from the command line using the syntax:

```
beasts [input data file] [results file] [working directory] [-na]
```

The *input data file* contains all information about the geometry and material properties etc. for the problem that is going to be analysed. It must be prepared in advance according to the specifications described in the next section. Once the analysis has been carried out, the *results file* stores the results. If the file already exists it will be overwritten.

The *working directory* is used by the programs FLIES and ANTS for storing the plotting files and exchanging temporary files. It will be created automatically if it does not already exist. Notice that all file and directory names must include the full path. The paths may be given relative to the current location on a hard disk, e.g. the path '*..\data\temp*' is satisfactory. A warning should be given at this stage against using the working directory for storage of any non-temporary files with extensions `.dat`, `.plt`, `.bin` or `.log`. Files with these extensions are deleted every time BEASTS runs. Thus it is advisable, though not necessary, to store the global input data files and result files in a different directory.

The argument `-na` (no analysis) is optional. If it is included, FLIES will be executed but ANTS will not run. This may be useful when checking the input file, see Section 4. A full listing of the MSDOS batch file is contained in Appendix A.

3 Data Preparation

BEASTS has a single global input data file, the structure of which will be described in this Section. The general layout of the input is similar to that used in the two-dimensional elastodynamic analysis program TEA [6]. Example input data files are given in Appendix B.

As discussed in the previous section, the sub-program FLIES reads the global input file and splits the data into a number of local input files for each of the boundary element sub-domains in the model. The local input files are stored in ASCII format with a layout directly corresponding to that of the global input file. The local input files will not be described here in further detail. Generally the layout of the local input files is similar to the layout of the global input file. The user may examine the files, but it is not advised that any modifications are made to the local input files as this may easily cause problems in the global assembly.

3.1 General Data Specification

The global input data file is an ordinary ASCII format file. As the program is not case sensitive, text may be in upper or lower case. The input file is processed line by line with no regard to the column position of data except that the data must be contained within the first 132 columns in all cases.

Data is specified in *modules*. Some of the modules are mandatory whereas other modules are optional. The first line in a module is the *module header* which is identified by the fact that it begins with the '*' character. The following module headers may be used in the global input data file:

**title*

**material properties*

**frequencies*

**multi file output*

**plane of symmetry*

**node coordinates*

**ANSYS node order*

**standard node order*

**boundary element domain*

**loads*

**end*

After the module header the relevant data for the module should appear. In the Subsections 3.2 to 3.12 the specification of the individual modules will be described in detail. Most of the modules contain numerical data. SI units (*i.e.* in the base dimensions kg, m and s) should be used throughout all modules of input data. In general, FLIES checks whether the appropriate number of data items has been specified. If this is not the case, the program will halt and an error message will be written to the log file `flies.log`.

Within some of the modules it is possible to use a number of *options*. An option is specified by starting the line with the '+' character. The options are generally used to reduce the amount of work necessary in order to generate the nodes and element mesh. See Subsections 3.2 to 3.12 for a further explanation of the options that apply within each of the modules. Note that it is not always necessary (though still recommended) to write the module or option headers in full length. The part of the headers that must be included as a minimum is underlined in the syntax definition for the respective modules/options.

Unless otherwise stated the modules may appear in any order and empty lines may be inserted at any place within or between modules to enhance the layout and readability of the data. Furthermore, comments may be included in the input data file in the same way as in a FORTRAN 77 source code. Thus, a *full-line comment* may be inserted by starting the line with the letter '*c*'. The beginning of an *end-of-line comment* is indicated by the '*!*' character.

3.2 The **Title* Module

Syntax: The **title* module has the following syntax:

```
*title
 title
```

Description: This module is optional. After the header, only one line of data must appear. This line, containing up to 132 characters of text, is the title of the job being processed. It will appear in the local input files written by FLIES as well as in the results file written by ANTS. However it is not included in the plotting files, which only contain numerical data. The title may contain the name of a structure, a job reference number or other kinds of information which may be used to keep track of input and output files. As an exception to the general rule, comments cannot be given in this module, as this would prevent the use of title beginning with a '*c*'.

3.3 The **Material Properties* Module

Syntax: The **material properties* module has the following syntax:

```
*material properties
   E   nu   rho   eta
   E   nu   rho   eta
   :   :     :     :
   E   nu   rho   eta
```

Description: This module is mandatory. Each data line in the module defines a single material and should contain the following four properties: Young's modulus, `E`, Poisson's ratio, `nu`, mass density, `rho`, and loss factor, `eta`. At least one line of data, *i.e.* the specification of one material, is required. The materials are referenced in the **boundary element domain* module by the order in which they appear in the **material properties* module. Thus, if many materials are defined, it may be useful to add the numbers of the material property sets as end-of-line comments, that is after a '*!*' character. The maximum number of different materials allowed is set in the maximum dimensions include file `maxdim.inc`, see Appendix C.

3.4 The **Frequencies* Module

Syntax: The **frequencies* module has the following syntax:

```
*frequencies
   f
   f
   ⋮
   f
```

Description: This module is mandatory. The module specifies a list of frequencies for which the analysis should be performed. Frequencies should appear in successive lines following the module header, one frequency, `f`, per line. In the results file the real and imaginary parts of the complex displacement amplitudes will be listed for each frequency in the order in which they appear in the **frequencies* module. Notice that the frequencies are the physical frequencies, f (measured in Hz), not the corresponding circular frequencies, $\omega = 2\pi f$.

3.5 The **Multi File Output* Module

Syntax: The **multi file output* module has the following syntax:

```
*multi file output
```

Description: This module is optional. When it appears in the input data file, plotting files named `cd__1.plt`, `cd__2.plt`, `cd__3.plt` etc. will be written in the working directory. Each of these files contain the complex amplitudes of displacement for one single frequency. No data after the header are given in the this module.

3.6 The **Plane of Symmetry* Module

Syntax: The **plane of symmetry* module has the following syntax:

```
*plane of symmetry
```

Description: This module is optional. If the module is included in the input data file, a plane of symmetry will be placed at $y = 0$. This may be useful for the analysis of models that are symmetric in one direction. There are no facilities for applying a plane of symmetry in the x or z directions. This should be kept in mind when generating the nodes. Furthermore, only positive y coordinates are allowed as the reflected part of the model is assumed to be present in the half-space $y < 0$.

By default both the geometry and the load/response is assumed to be symmetrical around the $y = 0$ plane when this module is included. The case of geometrical symmetry and anti-symmetric load and response may be analysed using the option *+antisymmetric load* , which is described below. It should be noted that any load on a symmetric structure may be split into a symmetric part and an anti-symmetric part. The total response may hence be found as a linear combination of the response due to the two parts, as it is common practice in FE analysis of symmetric models. However, at this stage, no facility exists in BEASTS to do this automatically.

The *+Antisymmetric Load* Option

Syntax: The *+antisymmetric load* option has the following syntax:

```
+antisymmetric load
```

Description: When the *+antisymmetric load* option appears in the **plane of symmetry* module, the geometry is still assumed to be symmetrical around the $y = 0$ plane. However the applied load, and therefore also the response, is assumed to be anti-symmetric.

3.7 The **Node Coordinates* Module

Syntax: The **node coordinates* module has the following syntax:

```
*node coordinates
   x    y    z
   x    y    z
   :    :    :
   x    y    z
```

Description: This module is mandatory and should appear only once in the input data file. It is preferable, though not necessary, for the **node coordinates* module to appear before any **boundary element domain* modules.

Each line of data specifies the x, the y and the z co-ordinate of a single node. Thus three numerical values, x, y and z, should appear in each line of the module. Node numbers should not be given in the input. The program automatically assigns *global node numbers* to the nodes due to the order in which they appear within the **node coordinates* module. However it may be useful for cross-reference to add node numbers after end-of-line comments. The user should be aware that the node numbers assigned by the program are later changed, so that the numbering used in the results file does generally not correspond to the original numbering. However one of the plotting files contains a conversion table that will help the user in bringing the nodes back into the original order.

The loads and element topology specifications in the global input data file use the original global node numbers, not the *local node numbers* defined in the local input data files generated by the program FLIES for each of the boundary element domains. Redundant nodes are allowed in the input, but they will be removed before the analysis is performed, and a warning is given in the log file if such nodes appear within the **node coordinates* module.

The *+Offset Nodes* Option

Syntax: The *+offset nodes* option has the following syntax:

```
+offset nodes
   node1   node2   Dx   Dy   Dz   NoC
```

The `NoC` (number of copies) parameter is optional. `NoC` is set to 1 if no other value is given.

Description: As an alternative to defining each node separately the *+offset nodes* option may be used to copy an array of nodes a number of times in a certain direction with a given spacing. Before this option may appear in the **node coordinates* module, at least one node should be defined. Otherwise no nodes exist to be copied.

The *+offset nodes* option has a single line of data following the option header. This line must contain the following values: `node1`, `node2`, `Dy`, `Dy`, `Dz` and `NoC`. The `node1` and `node2` parameters define the array of existing nodes which should be copied. Notice that `node1` must be less than or equal to `node2`. If `node1` is equal to `node2` only one node will be copied. The nodes `node1` to `node2` are offset by the distances `Dx`, `Dy`, and `Dz` in the x, y and z directions, respectively. The new nodes will be further offset until a total number of copies equal to `NoC` has been made.

Nodes which have been generated with the *+offset nodes* option may be further offset by successive use of the same option. The nodes receive global node numbers according to the order in which they are generated, that is counting through the nodes in the first copy, then through the nodes in the second copy, if any, and so forth.

3.8 The **ANSYS Node Order* Module

Syntax: The **ANSYS node order* module has the following syntax:

```
*ANSYS node order
```

Description: This module is optional but may appear several times. No data should appear in the module, only the header. The **ANSYS node order* module switches the interpretation of the element topology into ANSYS format, which is different from the standard format being used. See Section 3.10 for further details. Obviously the **ANSYS node order* module must appear before a **boundary element domain* module where the ANSYS node order should be used.

3.9 The **Standard Node Order* Module

Syntax: The **standard node order* module has the following syntax:

```
*standard node order
```

Description: This module is optional but may appear several times. No data should appear in the module, only the header. The module is the counterpart to the **ANSYS node order* module and is used to switch the element topology interpretation back to standard. It must appear in the input previous to a **boundary element domain* module where the standard node order should be used.

3.10 The **Boundary Element Domain* Module

Syntax: The **boundary element domain* module has the following syntax:

```
*boundary element domain
   mat.no.
   node1   node2   ...   nodeN
   node1   node2   ...   nodeN
     ⋮       ⋮             ⋮
   node1   node2   ...   nodeN
```

Description: This module is mandatory and should appear once for each of the boundary element domains in the model. The number of boundary element domains that may be used is controlled by a parameter in the maximum dimensions file (see Appendix C) as is the number of elements in each domain.

A boundary element domain consists of a single material only. The reference number for this material (see Subsection 3.3) is given as the first line of data following the module header. In the subsequent lines the topology data for the elements in the domain are given, one line for each element. The topology for the elements should contain the global node numbers corresponding to each of the `N` nodes in the element. Hence, a single line of data will contain `N` integers.

There are two types of boundary elements available in the program: a *six-noded triangular element* with quadratic interpolation and a *nine-noded quadrilateral elements* with biquadratic interpolation . The order in which node numbers should be given depends on whether the standard node order or the ANSYS node order is used. Figure 2 shows the node order in the two cases.

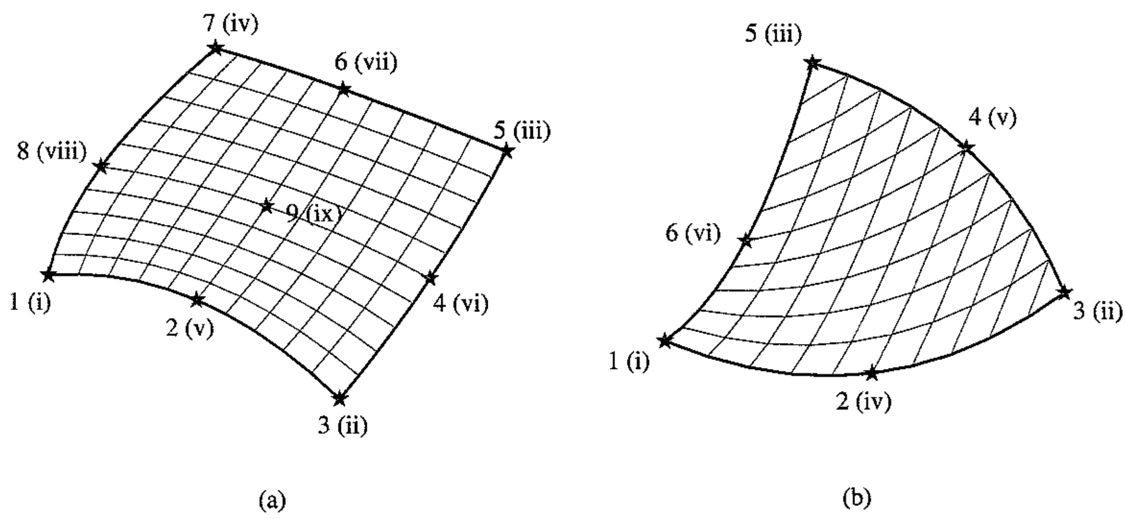

Figure 2. BEASTS element types: (a) Nine-noded quadrilateral element with biquadratic interpolation, (b) six-noded triangular element with quadratic interpolation. Arabic numerals indicate the standard node order, roman numerals indicate the ANSYS node order. The observer is placed *outside* the BE domain.

In the boundary element method the outward normal vector is used in the integration of the traction Green's function over the boundary, see [2]. To ensure that the normal vector which is derived from the element topology points in the right direction, the nodes should appear in clockwise order when looking 'through' the element from a point inside the domain.

The integration over the boundary elements is performed using Gaussian quadrature. However the singular terms of the Green's function for the traction have to be treated in a special way. Often this is done by static rigid body motion considerations (see [3]) which are however limited to the analysis of interior or exterior domains where the entire boundary is discretized, *i.e. closed* domains. For *open* domains where only part of the boundary is described with boundary element Ahmad and Banerjee [1] proposed the use of *enclosing elements* to transform the open region into a closed one. BEASTS uses a variant of this enclosing elements technique, but instead of enclosing the entire domain, a small closed region is established for a few elements at a time. The method is described in detail in [4] for the two-dimensional problem and in [2] for this, the three-dimensional problem. The authors believe that the method will work satisfactorily as long as the user constructs the element mesh in a reasonable way that is in agreement with good discretization practice, see [2] for further explanation.

The advantage of the local enclosing elements technique is that closed and open domains can be treated in the same way automatically, given that the user provides the information on how the boundary is supposed to continue beyond the edge for open domains. Thus, when modelling an open domain where only a part of the surface is described, e.g. a half-space, some of the elements will form a rim along the edge. In BEASTS these *edge elements* are used to obtain the necessary information about how the boundary continues outside the model. They are *not* regarded as true elements and will be discarded when the singular terms of the traction Green's function have been determined for the nodes close to the edge.

A variety of options are available within the **boundary element domain* module. Some of the options may be used to generate elements in a fast way. However the user should be aware that this may cause the input file to be less intelligible. The options are listed below. Unless otherwise stated, only the option header should appear.

The *+No Edge Elements* Option

Syntax: The *+no edge elements* option has the following syntax:

```
+no edge elements
```

Description: When dealing with an open domain, BEASTS will by default use the rim of elements along the edge as *edge elements*. The appearance of the *+no edge elements* option within a **boundary element domain* module tells the program to disallow this procedure. Hereby the standard procedure to determine the diagonal terms of the static part of matrix **H** (see [2]) cannot be used for the nodes on the edge. Instead these terms will all be set to 0.5, assuming that the edge is on a smooth and flat part of the boundary.

The +*Mirror On* Option

Syntax: The +*mirror on* option has the following syntax:

```
+mirror on
```

Description: As explained in the text above, the direction of the normal vector is crucial in the boundary element method. As it may sometimes be difficult to see how the nodes in an element should be ordered to ensure the right direction of the normal vector, a method to turn the normal vector 180° in an easy way is provided. By insertion of the +*mirror on* option within the **boundary element domain* module, subsequent element topology data will be read in a different order so that the normal is mirrored around the element.

The +*Mirror Off* Option

Syntax: The +*mirror off* option has the following syntax:

```
+mirror off
```

Description: When this option is used, the normal for any subsequent elements in the **boundary element domain* module will not be mirrored around the element, but will keep its original direction as given by the topology listed in the input data file.

The +*Mirror* Option

Syntax: The +*mirror* option has the following syntax:

```
+mirror
```

Description: This option shifts the mirror on/off. In other words, it may be used instead of either the +*mirror on* option or the +*mirror off* option to change the interpretation of the element topology with respect to the direction of the normal vector.

The +*Copy Elements* Option

Syntax: The +*copy elements* option has the following syntax:

```
+copy elements
   elem1   elem2   nodestep   NoC
```

The `NoC` (number of copies) parameter is optional. `NoC` is set to 1 if no other value is given.

Description: Instead of giving the topology for all elements in a BE sub-domain, a few of the elements may be defined in the standard way and the rest of them may be generated using the *+copy elements* option. This option is to some extent similar to the *+offset nodes* option, but works for elements instead of nodes.

The *+copy elements* option has four items of data. The parameters `elem1` and `elem2` define an array of elements that should be copied. `elem1` must be less than `elem2` or, alternatively, equal to `elem2` in which case only one element is copied. The parameter `nodestep` defines the increment in node numbers between the original elements' topology and the new elements' topology. For example a copy of an element with the topology 1 2 3 4 5 6 would result in a new element with the topology 11 12 13 14 15 16, given that `nodestep` was set to 10. Finally `NoC` defines the number of copies that should be taken of the original elements. The node numbers will be further increased by `nodestep` in each successive copy.

The *+Mirror Copy Elements* Option

Syntax: The *+mirror copy elements* option has the following syntax:

```
+mirror copy elements
   elem1   elem2   nodestep
```

Description: This option is a special version of the *+copy elements* option and works in a similar way. However, when the *+mirror copy elements* option is used, the normal vector of the new elements is mirrored around the element. This is ideal for generating the opposing side of a symmetric structure or the other side of a soil layer, once one side has been made in the standard way or using the *+copy elements* option. Notice that only one copy of the original elements may be taken with the *+copy elements* option.

The *+ANSYS Node Order* Option

Syntax: The *+ANSYS node order* option has the following syntax:

```
+ANSYS node order
```

Description: This option shifts the local interpretation of the element topology into ANSYS mode for the rest of the **boundary element domain* module. However, any succeeding **boundary element domain* modules will still use the global topology interpretation mode set by either the **ANSYS node order* module or the **standard node order* module.

The *+Standard Node Order* Option

Syntax: The *+standard node order* option has the following syntax:

```
+standard node order
```

Description: Similar to the *+ANSYS node order* option, this is a local version of the **standard node order* module.

3.11 The **Loads* Module

Syntax: The **loads* module has the following syntax:

```
*loads
   node   RePx   ImPx   RePy   ImPy   RePz   ImPz
   node   RePx   ImPx   RePy   ImPy   RePz   ImPz
    ⋮      ⋮      ⋮      ⋮      ⋮      ⋮      ⋮
   node   RePx   ImPx   RePy   ImPy   RePz   ImPz
```

Description: This module is mandatory and must appear only once in the global input data file. The load in BEASTS is applied as point loads in the finite element sense. These loads are defined in the lines following the header of this module. The `node` parameter (which must be given as the first item of data in a line) indicates the global node (in the original numbering) where a point load should be applied. The next six values are `RePx`, `ImPx`, `RePy`, `ImPy`, `RePz` and `ImPz`. They define the real and imaginary part of the complex point load amplitude in the x, the y and the z direction, respectively.

3.12 The **End* Module

Syntax: The **end* module has the following syntax:

```
*end
```

This module is mandatory. It must appear as the last module of the global input data file and no further data is given after the header.

4 Checking the Input Data

FLIES checks if all mandatory modules appear in the input file and if there are any consistency errors in the data such as references to non-existing global nodes in the element topology data. Also, FLIES tests whether the array sizes that have been set in the include file `maxdim.inc` (see Appendix C) are high enough for a given problem, or if, for example, the allowed number of global nodes should be higher.

However the user has to check if the geometry produced from the input data is correct. Especially in the boundary element method, it is crucial that the normal vectors point in the right direction, *i.e. out* from the surface. To visualize the mesh, which may be very helpful in the error checking phase, and also for presentation of the model, a program has been produced in MATLAB. This program, called BUGS (BEASTS User Graphics System), shows the elements, nodes and normal vectors of the model using the plotting data files provided by FLIES. Furthermore, the position of applied point loads is indicated and the element and node numbers may be shown. Here it should be noticed that the node numbers are plotted in the original order in which they appeared in the global input file as defined by the user. However, the element numbers are given in the rearranged order with true elements appearing before edge elements.

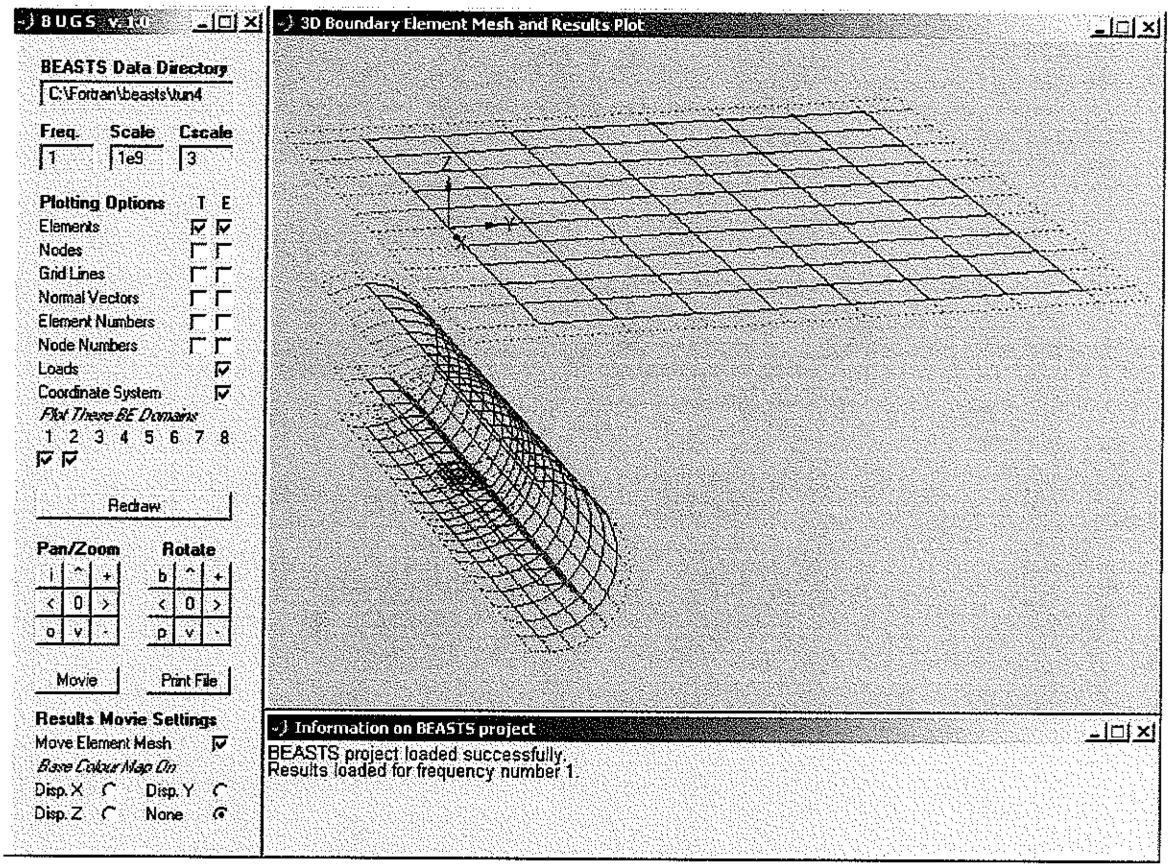

Figure 3. BUGS user interface. The BE mesh shown in the figure corresponds to the numerical example given in Section 6.

Finally, if an error occurred in the input data, so that the plotting files were never written by FLIES, this will be reported in BUGS (reading the error message in `flies.log`).

The program has a user interface as illustrated in Figure 3. To alter the view of the model, the Pan/Zoom and Rotate facilities may be used. The i and o buttons are used to zoom in and out, respectively, and the + and - buttons are used to adjust the pan/zoom or rotation step. The Print File button creates an Encapsulated Post Script file called `BEASTS_figure.eps` with the contents of the '3D Boundary Element Mesh and Results Plot' window. This file may afterwards be found in the BEASTS data directory. Any existing figure file will be overwritten.

Apart from the description given in the above text, the use of BUGS to illustrate the model geometry is straight forward and no further documentation will be provided in this report. User help is given in each of the MATLAB subroutines used to plot the elements and the normal vectors in BUGS.

5 Retrieving the Results

The results produced by ANTS are saved in the main results file, the name of which is declared when running BEASTS. At the beginning of the results file, the title of the BEASTS project appears and after this all the node coordinates are listed with the revised node numbering. A line contains four items of data being the node number, the x, the y and the z co-ordinate. Thus the co-ordinate listing appears in the following way.

```
node   x   y   z
node   x   y   z
 ⋮     ⋮   ⋮   ⋮
node   x   y   z
```

Notice that only nodes belonging to true elements are listed. Any nodes that are only used by edge elements are not part of the global system and are therefore discarded in the final output.

After the co-ordinates, the complex amplitudes calculated for each of the nodes are listed for one frequency at a time. The frequencies, for which the results are listed, appear in the headers of the respective modules. The remaining lines of the module for a single frequency contain the following data:

```
node   ReUx   ImUx   ReUy   ImUy   ReUz   ImUz
node   ReUx   ImUx   ReUy   ImUy   ReUz   ImPz
 ⋮      ⋮      ⋮      ⋮      ⋮      ⋮      ⋮
node   ReUx   ImUx   ReUy   ImUy   ReUz   ImUz
```

Here `node` is the node number (in the revised node numbering scheme) and `ReUx`, `ImUx`, `ReUy`, `ImUy`, `ReUz` and `ImUz` are the real and imaginary parts of the complex amplitudes at the nodes in the x, the y and the z direction, respectively.

If the **multi file output* module is included in the global input file, a separate list of the complex amplitudes (real and imaginary parts) will be saved in addition to the main results file (see Subsections 2.1 and 3.5). These files may be used for plotting the results in various ways with the BUGS program.

When the Movie button in BUGS is pressed, an .avi format movie with the displacement time series for a single loading period is generated. The movie is saved in the BEASTS data directory with the file name `BEASTS_movie.avi`. The mesh displacements are scaled by the Scale parameter. However, since displacements in some parts of the model may often be orders of magnitude smaller than the displacements in other parts of the model an alternative means of illustrating the results is offered.

Thus, a coloured surface may be plotted with negative displacements indicated in blue, positive displacements in red and zero displacements in green. Either the x, the y or the z component of the displacements may be shown in this way. If the Cscale (colour scale) parameter is set to a positive number, n, the colour intensity is based on the nth root of the absolute value of the displacement amplitudes (*i.e.* small displacements are 'magnified' for large values of n). When Cscale is set to zero, the colours are based on the logarithmic values of the amplitudes. A warning should be given here that this actually means that errors in the results (which are often about ten orders of magnitude smaller than the results when double precision is used) become comparable in size to the actual results. Thus, for instance, unexpected and erroneous non-zero displacements across a line of symmetry may appear; but actually these displacements are only due to a small amount of inaccuracy in the calculation.

6 Numerical Example: A Bored Tunnel in a Half-Space

Reference [2] describes the theory that has been implemented in the computer program BEASTS. In that report numerical examples were given, where the wave propagation in a homogeneous and a layered half-space due to a surface load was studied using various methods of analysis. Here an example is presented in which BEASTS is used for the analysis of a bored tunnel in a half-space.

6.1 Description of the Model

A bored tunnel in a homogeneous half-space is considered. The tunnel has a depth of 12.0 metres (*i.e.* the tunnel floor is 12 metres below the surface of the ground) and has a circular cross section with a diameter of 6.0 m. The data do not represent a specific location and the example is merely provided as an illustration of the use of the software. On Figure 4 the cross section of the tunnel is shown with stars indicating the position of nodes in the BE mesh. At the centre of each element (only quadrilateral elements are used) the normal vectors are plotted. The symmetry around the $x - z$ plane is utilized so that the tunnel and half-space are only modelled at $y \geq 0$. The dotted elements are edge elements (see Subsection 3.10) and are only used to calculate the diagonal terms of the Green's function matrix for the tractions (see Reference [2]).

In this simplified tunnel model there is no concrete lining. The reason is that no finite elements are available in BEASTS at the present stage and boundary elements are not suitable for the analysis of thin structures. The distance between the two surfaces necessary to model

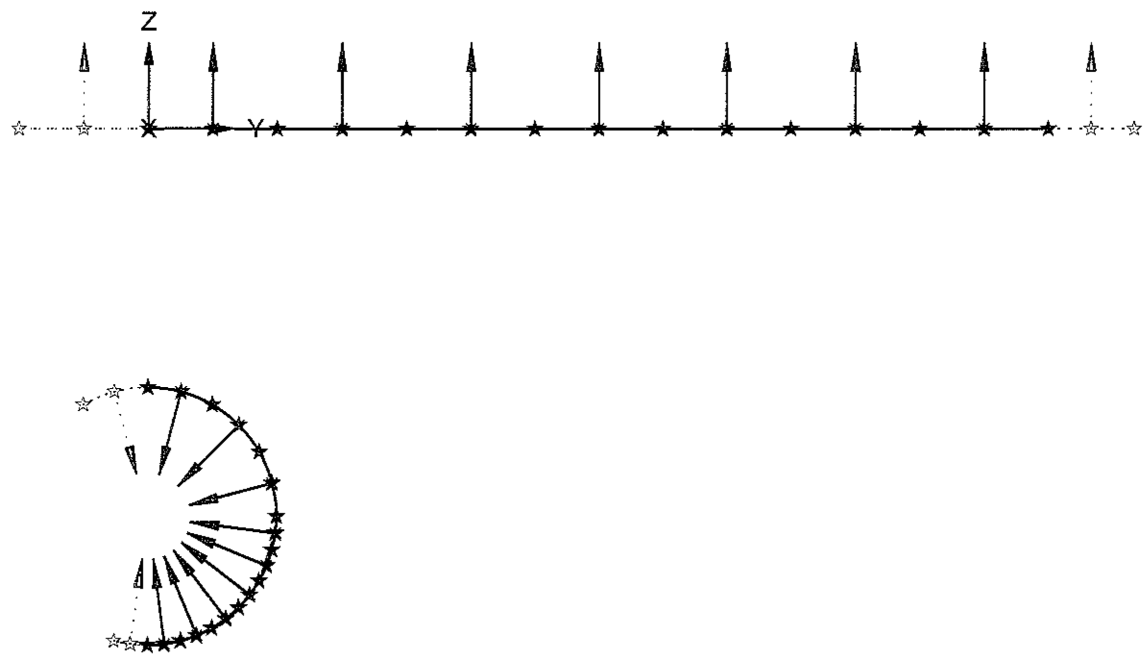

Figure 4. Cross section of the model of a bored tunnel in a homogeneous half-space with BE discretization in the $y - z$ plane. The magenta arrows are the outward normal vectors at the centre of each element and dashed lines indicate edge elements.

the concrete lining with boundary elements should be at least one element apart due to the singularities in the Green's functions. Hence, for a realistic width of the lining, an element size of no more than approximately 0.3×0.3 m^2 should be used. For this reason the number of degrees of freedom in the model would be enormous.

Inside the tunnel a concrete beam has been placed as a simple model of a railway track. This tunnel invert has been made with an unrealistic height (much higher than a real track) in order to reduce the number of degrees of freedom due to the limitations of the BEM explained above. To counteract the increased height of the track the stiffness of the material is reduced compared to the typical stiffness of concrete so that the bending stiffness of the beam is approximately the same as it would have been for a real track. The cross section of the tunnel invert is illustrated on Figure 5 and material properties for both the half-space and the tunnel invert are listed in Table 1.

Table 1. Parameters for the half-space and the tunnel invert.

Sub-domain	E [MPa]	ν	ρ [kg/m^3]	η
Half-space	350	0.35	2000	0.10
Tunnel invert	7000	0.15	2400	0.03

On Figure 6 the entire model (without normal vectors) is shown. The track is subject to a vertical load on a small area in the middle of the modelled part of the tunnel. In BEASTS the load is applied as point forces. The nodes where the load is applied are marked on Figure 6. The point loads are applied so that they correspond to a uniformly distributed vertical surface traction with a total magnitude of 1 N. Close to the loaded area, the mesh used for the discretization of the tunnel and the invert has been refined. The distances between the individual layers of nodes in the x-direction may be found in Appendix B where the input data for BEASTS are listed, including the point forces applied to each of the loaded nodes.

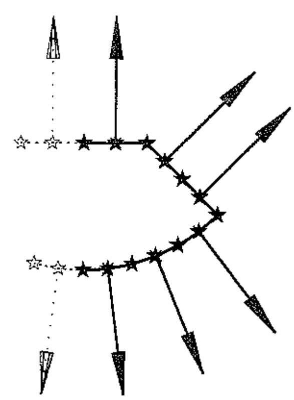

Figure 5. Cross section of the tunnel invert. The magenta arrows are the outward normal vectors at the centre of each element and dashed lines indicate edge elements.

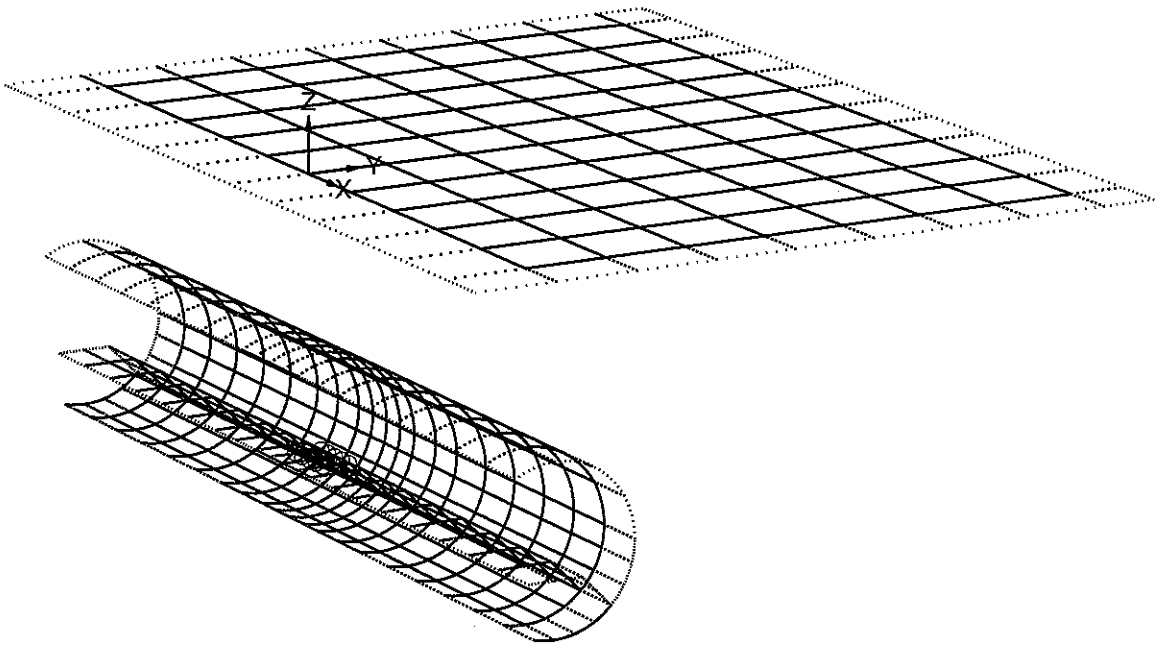

Figure 6. Mesh used for boundary element analysis of a bored tunnel in a homogeneous half-space. Dashed lines indicate edge elements and red circles show nodes where point loads are applied.

6.2 Example Results

Figure 7 shows the displacement pattern at various time steps during an excitation period for each of the frequencies 20, 40 and 60 Hz. In all cases the displacement is scaled by the factor 10^9 and the colours indicating the vertical displacements are based on the third root of the absolute values of the displacement amplitudes, thus magnifying small displacements.

For all the analysed frequencies the wave propagation pattern on the surface of the ground is seen to be rather complex just above the tunnel. The 'shadow zone' reaches approximately 12 metres away from the plane of symmetry corresponding to the depth of the tunnel. Beyond this distance the displacements appear in the shapes of regular, almost circular wave fronts, indicating that the influence of the tunnel is vanishing. Regarding the reliability of the results it is clear that the model cannot be used for frequencies much higher than 60 Hz as the number of elements per wavelength at this frequency is already very small (approximately two elements are available per Rayleigh wavelength on the half-space surface). Furthermore, at the artificial ends of the model in the direction of the tunnel the results may be erroneous. Since the surface of the half-space would, in reality, 'feel' parts of the tunnel beyond the artificial end of the model the surface displacements may be inaccurately determined. The extra contributions to the surface displacement from the part of the tunnel and track which is not modelled may not necessarily increase the response at the surface. In fact a reduction is possible because the waves arriving at a single surface point from different points along the beam are out of phase. Anyway, the opposite effect (*i.e.* the influence from the non-modelled part of the surface to the modelled part of the track) is likely to be less pronounced since the displacements are much smaller at the surface than those of the track.

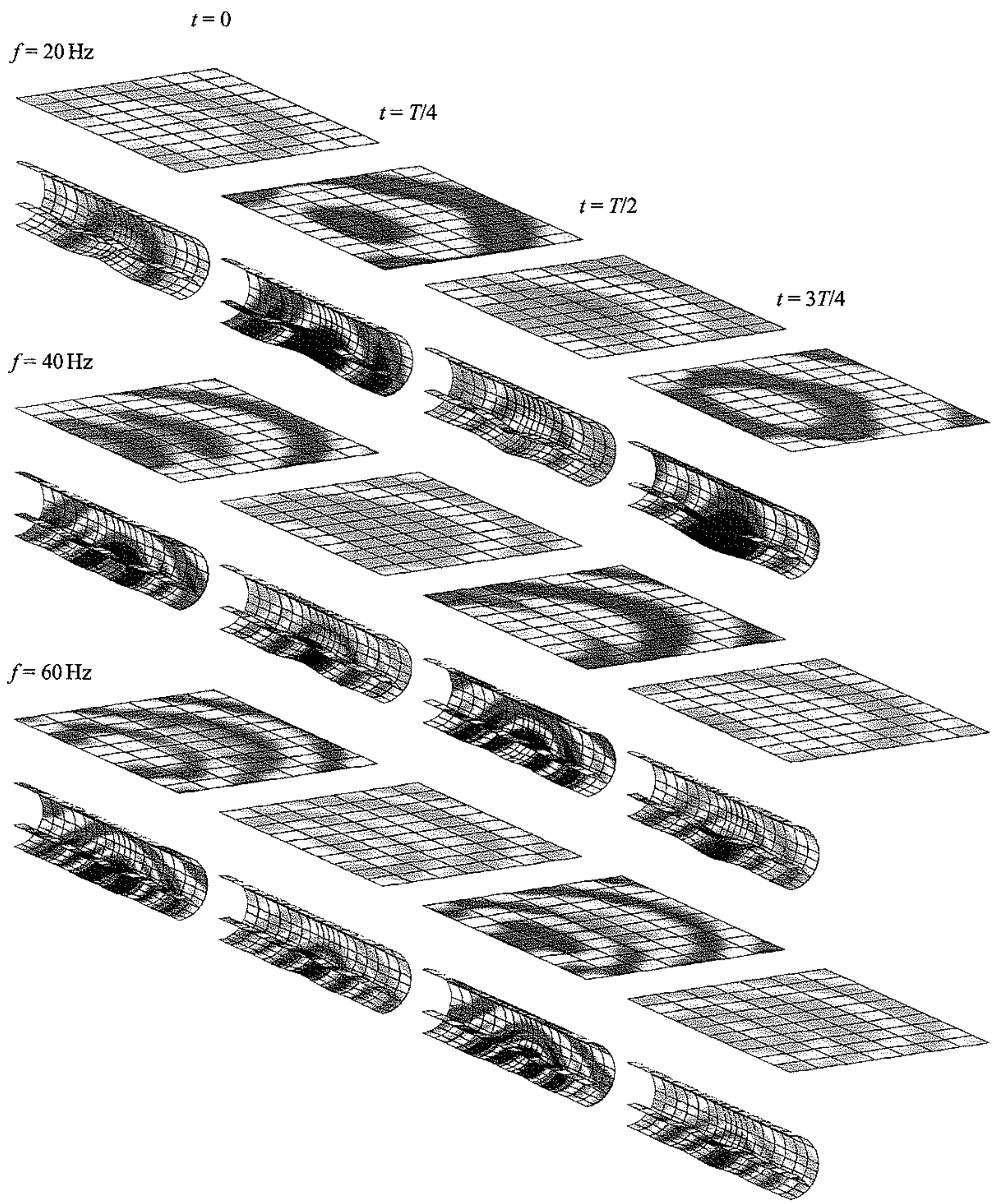

Figure 7. Results from the boundary element analysis of the bored tunnel in a homogeneous half-space at the three frequencies 20, 40 and 60 Hz. Red and blue colours indicate positive and negative vertical displacements, respectively, and parts of the surface with near-zero vertical displacement are in shades of green.

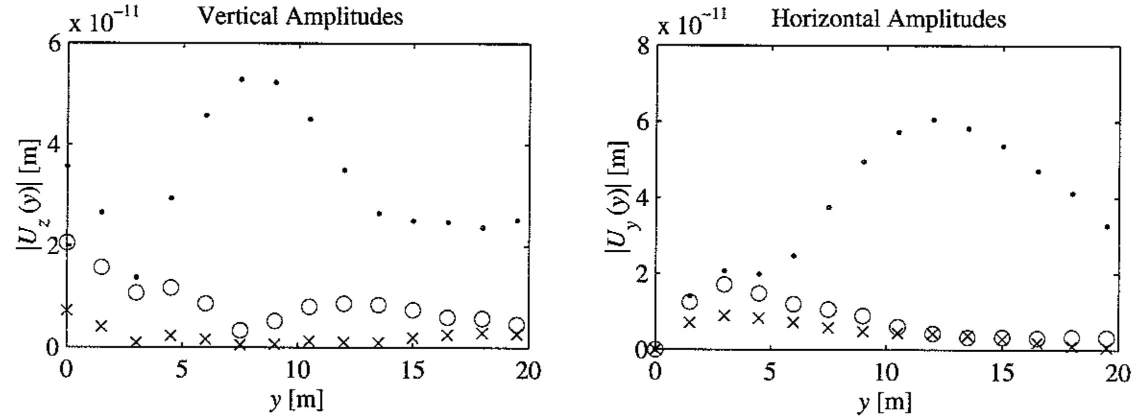

Figure 8. Displacement amplitudes at the surface of the half-space along the y-axis. The results for 20, 40 and 60 Hz are plotted with red dots $(\cdot)$, green circles $(\circ)$ and blue crosses $(\times)$, respectively.

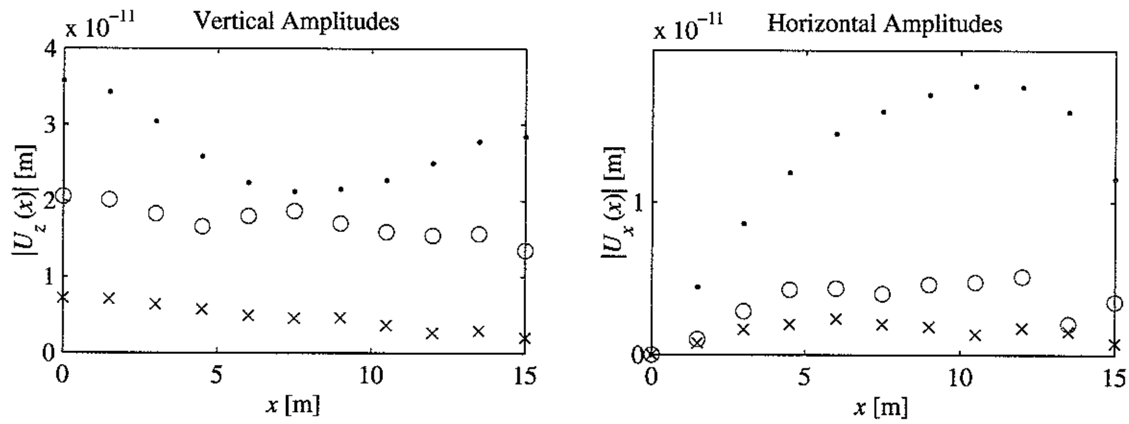

Figure 9. Displacement amplitudes at the surface of the half-space along the x-axis. The results for 20, 40 and 60 Hz are plotted with red dots $(\cdot)$, green circles $(\circ)$ and blue crosses $(\times)$, respectively.

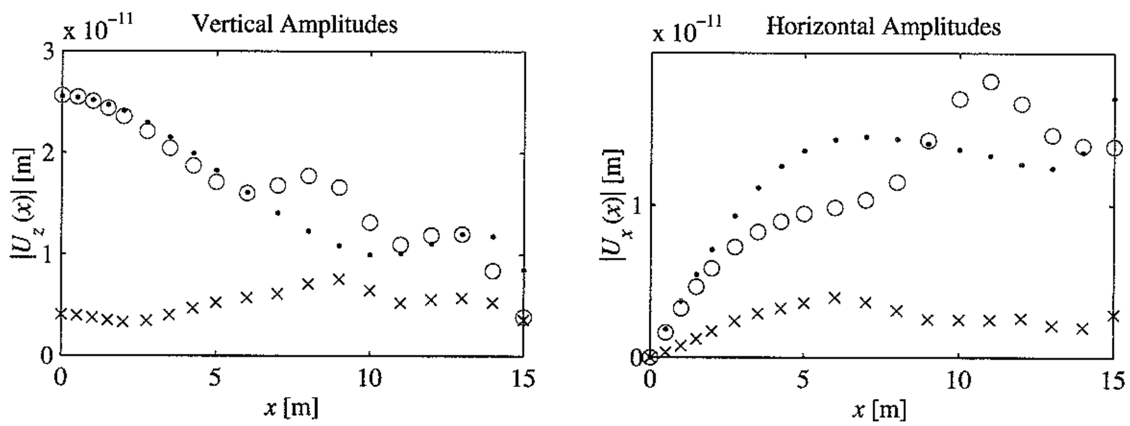

Figure 10. Displacement amplitudes at the crown of the tunnel along the x-axis. The results for 20, 40 and 60 Hz are plotted with red dots $(\cdot)$, green circles $(\circ)$ and blue crosses $(\times)$, respectively.

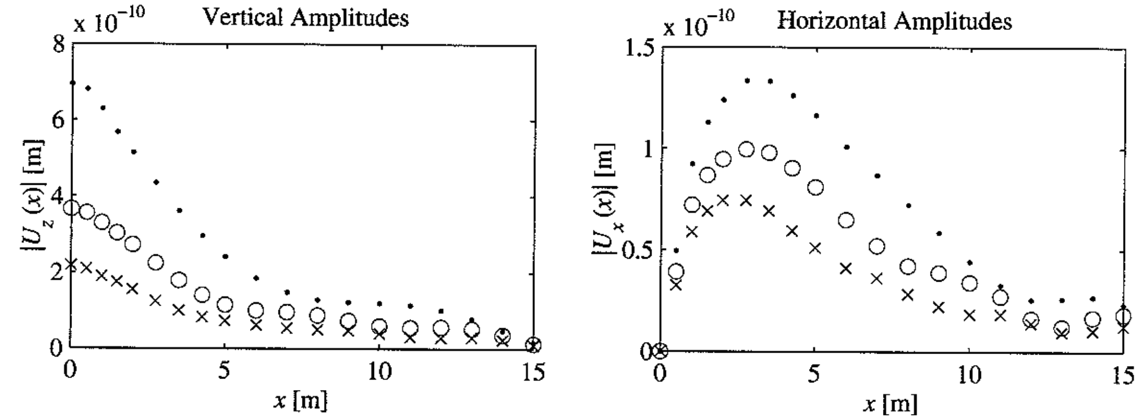

Figure 11. Displacement amplitudes at the top of the track along the x-axis. The results for 20, 40 and 60 Hz are plotted with red dots (·), green circles (∘) and blue crosses (×), respectively.

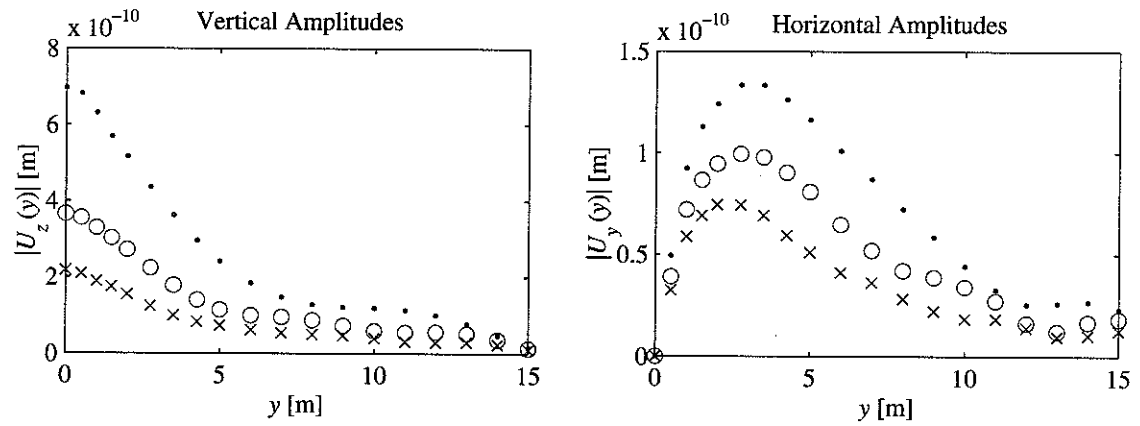

Figure 12. Displacement amplitudes at the bottom of the tunnel/track along the x-axis. The results for 20, 40 and 60 Hz are plotted with red dots (·), green circles (∘) and blue crosses (×), respectively.

A further look at the colour intensity on Figure 7 may suggest that the overall response is stronger at 20 Hz than is the case for 40 and 60 Hz, specifically at the surface of the half-space at a distance of 8 - 9 metres away from the plane of symmetry along the y-axis. This becomes clearly evident on Figure 8 where the displacement amplitudes on the surface of the half-space along the y-axis are compared for the three frequencies. Figure 9 shows that similar observations can be made for the surface displacements right above the centre line of the tunnel, particularly in the case of the horizontal displacements.

The Figures 10 to 12 illustrate the displacement amplitudes along the x-axis at various depths, namely at the crown of the tunnel (6 m below the surface of the ground), at the top of the track (10 m below the surface) and at the bottom of the tunnel. Several interesting findings can be made, some of which are described below.

A comparison of Figures 9 and 10 shows that in some cases the displacement response at the surface just above the tunnel is stronger than that at the crown of the tunnel, even though the distance from the load is much bigger. The phenomenon is most pronounced for the vertical displacements at 10 Hz. Another interesting feature is that the vertical displacements at the crown are almost identical for the frequencies 20 Hz and 40 Hz in the vicinity of the loaded area. When Figures 9 and 10 are compared with 8 it also becomes evident that the surface displacement response at $f = 20$ Hz, 8 - 9 metres away from the plane of symmetry is stronger than it is anywhere immediately above the tunnel. This means that, for the frequency 20 Hz, the most critical area on the surface is actually a distance away from the tunnel where the foundations of buildings might be if the tunnel was bored under a street.

Finally, from Figures 11 and 12 it is observed that the displacement amplitude at the top and at the bottom of the track are close to being identical. Thus it may be concluded that the track acts like a beam with no internal deformation of the cross section. At low frequencies, below 20 Hz, the vertical displacement amplitude drops more quickly with increasing distance from the load along the tunnel than for the higher frequencies. This may indicate that the major part of the displacements at 20 Hz are due to evanescent waves. A phenomenon of this kind would be expected as travelling waves are known not to exist in beams on elastic foundations below a certain cut-off frequency.

7 Conclusions

In this report the user documentation has been given for the computer program BEASTS where the boundary element theory described in Reference [2] has been implemented. Two element types are available: a 9 noded quadrilateral element and a 6 noded triangular element. The quadratic interpolation makes the program suitable for the analysis of all kinds of elastodynamic problems, including flexural wave problems, and the program may be used for open as well as closed domains. A number of options are available which simplifies the generation of the node and element mesh for a model, and a separate program has been made in MATLAB for illustration of the geometry and results for a BEASTS project.

As an example of the use of BEASTS, an analysis has been carried out for a bored tunnel in a half-space. For the unlined case, which has been considered in the present project, a model that provides satisfactory results for frequencies up to 60 Hz can be made with a reasonable number of degrees of freedom, so that the analysis may be performed on a decent computer. The analysis shows that the response at the surface of the ground due to a vertical load in the tunnel is much stronger for a relatively low frequency of 20 Hz than it is when the excitation frequency is 60 Hz. Furthermore, the largest displacement amplitudes are encountered at a distance away from the centre line of the tunnel, which is of the order of the depth of the tunnel.

Acknowledgement

L. Andersen would like to thank the Danish Technical Research Council for financial support via the research project: 'Damping Mechanisms in Dynamics of Structures and Materials'.

References

[1] S. Ahmad and P.K. Banerjee. Multi-domain BEM for two-dimansional problems of elastodynamics. *Int.J.Num.Meth.Eng.*, 26:891–911, 1988.

[2] L. Andersen and C.J.C. Jones. Three-Dimensional Elastodynamic Analysis Using Multiple Boundary Element Domains. ISVR Technical Memorandum 867, Institute of Sound and Vibration Research, University of Southampton, Southampton SO17 1BJ, England, October 2001.

[3] J. Dominguez. *Boundary elements in dynamics*. Computational Mechanics Publications, Southampton, 1993.

[4] C.J.C. Jones and D.J. Thompson. A boundary Element Model for Two-Dimensional Elastodynamics on a Single Open or Closed Domain. ISVR Technical Memorandum 838, Institute of Sound and Vibration Research, University of Southampton, Southampton SO17 1BJ, England, May 1999.

[5] C.J.C. Jones, D.J. Thompson, and M. Petyt. Ground-Borne Vibration and Noise from Trains: Elastodynamic Analysis Using the Combined Boundary Element and Finite Element Methods. ISVR Technical Memorandum 844, Institute of Sound and Vibration Research, University of Southampton, Southampton SO17 1BJ, England, October 1999.

[6] C.J.C. Jones, D.J. Thompson, and M. Petyt. TEA - A Suite of Computer Programs for Elastoydnamic Analysis Using Coupled Boundary Elements and Finite elements. ISVR Technical Memorandum 840, Institute of Sound and Vibration Research, University of Southampton, Southampton SO17 1BJ, England, August 1999.

[7] O. von Estorff and E. Kausel. Coupling of boundary and finite elements for soil- structure interaction problems. *Earthquake Eng. Struct. Dyn.*, 18:1065–1075, 1989.

A Listing of BEASTS.BAT

This MSDOS or Windows 2000 batch file is run from the command prompt. It automatically creates a directory where all files with plotting data for a BEASTS project will be stored for later use in BUGS.

```
@rem start of file beasts.bat
@echo off

rem   Run this program from the dos-promt using the following command:
rem   beasts [input path/file] [output path/file] [working directory] [-na]
rem   If the -na option is included, no analysis will be performed.

rem Store current directory and set file/temp directory variables

  set thisdir=%CD%
  set inpfile=%1
  set resfile=%2
  set tempdir=%3
  set runants=%4

  if not defined runants set runants=-da
  if not defined tempdir goto ERROR1
  if not exist %inpfile% goto ERROR2

rem Check if the working directory already exists

  if not exist %tempdir% goto MAKEWD

rem Delete all .dat, .plt and .dat files in existing working directory

  cd %tempdir%

  del *.dat
  del *.plt
  del *.log
  del *.bin

  cd %thisdir%

  goto COPYINPUT

:MAKEWD

rem Create new working directory

  md %tempdir%

:COPYINPUT

  copy %inpfile% %tempdir%\input.dat
```

```
  cls

  echo ***********************************************************************
  echo *                                                                     *
  echo * Boundary Element Analysis of Soil and Three-dimensional Structures *
  echo *                                                                     *
  echo * ------------------------------------------------------------------ *
  echo *                                                                     *
  echo * A BEM program by L. Andersen and C.J.C. Jones, ISVR, July 2001     *
  echo *                                                                     *
  echo ***********************************************************************

rem Change directory to working directory

  cd %tempdir%

rem Generate seperate input files for each BE/FE domain

  %thisdir%\bin\flies

rem Delete the copy of the input file

  del input.dat

rem Check if input was alright

  if not exist globalxyz.plt goto ERROR3

rem Skip analysis if the -na option has been used

  if %runants%==-na goto SKIPANTS

rem Global assembly and solver for coubled domains

  %thisdir%\bin\ants

rem Remove binary files (which may take up quite a lot of space)

  del *.bin

  echo Finished BEASTS analysis %date% %time%.

:SKIPANTS

rem Return to initial directory

  cd %thisdir%

rem Move results file to program directory

  if exist %resfile% del %resfile%
  if exist %tempdir%\results.dat move %tempdir%\results.dat %resfile%
```

```
  goto FINISH

:ERROR1

  echo BEASTS: Parameter format not correct. Try using this format:
  echo beasts [input data file] [results file] [working directory] [-na]
  echo If the -na option is included, no analysis will be performed.
  goto FINISH

:ERROR2

  echo *** ERROR *** Input file not found: %inpfile%
  goto FINISH

:ERROR3

  echo *** ERROR *** Bad input. See the error report in flies.log.
  cd %thisdir%

:FINISH

rem Clear variables

  set thisdir=
  set inpfile=
  set resfile=
  set tempdir=
  set runants=

@echo on
@rem End of file beasts.bat
```

B Example Input File

This is the input file for the computer program BEASTS used for the boundary element analysis of an unlined tunnel in Subsection 6.

```
*Title

An Unlined Tunnel, LA, 10/8/2001

C This is an example file for use with the BE program BEASTS

*Material properties

C      E       nu     rho      lf

     0.35D9  0.250  2000.0  0.10 ! ground
     7.00D9  0.150  2400.0  0.03 ! concrete (reduced strength)

*multi file output

*plane of symmetry

*Frequencies

     20.000
     40.000
     60.000

*Node coordinates

C The tunnel cross section at x = -17.00

 -17.00000000000000   0.00000000000000  -6.00000000000000 !  1
 -17.00000000000000   0.77645713530756  -6.10222252113280 !  2
 -17.00000000000000   1.50000000000000  -6.40192378864668 !  3
 -17.00000000000000   2.12132034355964  -6.87867965644036 !  4
 -17.00000000000000   2.59807621135332  -7.50000000000000 !  5
 -17.00000000000000   2.89777747886720  -8.22354286469244 !  6
 -17.00000000000000   3.00000000000000  -9.00000000000000 !  7
 -17.00000000000000   2.97433458412143  -9.39157857666016 !  8
 -17.00000000000000   2.89777747886720  -9.77645713530756 !  9
 -17.00000000000000   2.77163859753386 -10.14805029709527 ! 10
 -17.00000000000000   2.59807621135332 -10.50000000000000 ! 11
 -17.00000000000000   2.38006002087371 -10.82628428702616 ! 12
 -17.00000000000000   2.12132034355964 -11.12132034355964 ! 13
 -17.00000000000000   1.82628428702616 -11.38006002087371 ! 14
 -17.00000000000000   1.50000000000000 -11.59807621135332 ! 15
 -17.00000000000000   1.14805029709527 -11.77163859753386 ! 16
 -17.00000000000000   0.77645713530756 -11.89777747886721 ! 17
 -17.00000000000000   0.39157857666016 -11.97433458412143 ! 18
 -17.00000000000000   0.00000000000000 -12.00000000000000 ! 19
```

```
C The tunnel invert at x = -17.00

 +offset nodes
      19   19   0.00    0.00    2.00   1 ! 20

 +offset nodes
      20   20   0.00    0.50    0.00   2 ! 21 to 22

 +offset nodes
      22   22   0.00    0.28033008588991  -0.28033008588991   3 ! 23 to 25

C The tunnel and invert from x = -16.00 to x = -5.00

 +offset nodes
       1   25   1.00    0.00    0.00  12 ! 26 to 325

C The tunnel and invert from x = -4.25 to x = -2.00

 +offset nodes
     301  325   0.75    0.00    0.00   4 ! 326 to 425

C The tunnel and invert from x = -1.50 to x = 2.00

 +offset nodes
     401  425   0.50    0.00    0.00   8 ! 426 to 625

C The tunnel and invert from x = 2.75 to x = 5.00

 +offset nodes
     601  625   0.75    0.00    0.00   4 ! 626 to 725

C The tunnel and invert from x = 6.00 to x = 17.00

 +offset nodes
     701  725   1.00    0.00    0.00  12 ! 726 to 1025

C The first node of the half-space surface

 +offset nodes
       1    1   0.00    0.00    6.00   1 ! 1026

C The rest of the half-space surface

 +offset nodes
    1026 1026   0.00    1.50    0.00  14 ! 1027 to 1040

 +offset nodes
    1040 1040   0.00    1.00    0.00   2 ! 1041 to 1042

 +offset nodes
    1026 1042   1.00    0.00    0.00   2 ! 1043 to 1076
```

```
 +offset nodes
    1060 1076   1.50    0.00    0.00   20 ! 1077 to 1416

 +offset nodes
    1400 1416   1.00    0.00    0.00    2 ! 1417 to 1440

*Boundary Element Domain 1 ! The soil

      1 ! Material Properties Set 1 - Soil

      1    2    3   28   53   52   51   26   27 ! 1

 +copy elements
      1   1   2   8 ! 2 to 9

 +copy elements
      1   9  50  19 ! 10 to 180

 +mirror on

    1026 1027 1028 1045 1062 1061 1060 1043 1044 ! 181

 +mirror off

 +copy elements
     181  181    2    7 ! 182 to 188

 +copy elements
     181  188   34   11 ! 189 to 276

*Boundary Element Domain 2 ! The tunnel invert

      2 ! Material Properties Set 2 - Concrete

 +mirror on

     13  14  15  40  65  64  63  38  39 ! 1

 +copy elements
      1   1   2   2 ! 2 to 3

 +copy elements
      1   1   7   1 ! 4

 +copy elements
      4   4   2   1 ! 5

     24  25  13  38  63  75  74  49  50 ! 6

 +copy elements
      1   6  50  19 ! 7 to 120
```

*Loads

470	0.000	0.000	0.000	0.000	0.0138889	0.000
471	0.000	0.000	0.000	0.000	0.0555556	0.000
472	0.000	0.000	0.000	0.000	0.0138889	0.000
495	0.000	0.000	0.000	0.000	0.0555556	0.000
496	0.000	0.000	0.000	0.000	0.2222222	0.000
497	0.000	0.000	0.000	0.000	0.0555556	0.000
520	0.000	0.000	0.000	0.000	0.0277778	0.000
521	0.000	0.000	0.000	0.000	0.1111111	0.000
522	0.000	0.000	0.000	0.000	0.0277778	0.000
545	0.000	0.000	0.000	0.000	0.0555556	0.000
546	0.000	0.000	0.000	0.000	0.2222222	0.000
547	0.000	0.000	0.000	0.000	0.0555556	0.000
570	0.000	0.000	0.000	0.000	0.0138889	0.000
571	0.000	0.000	0.000	0.000	0.0555556	0.000
572	0.000	0.000	0.000	0.000	0.0138889	0.000

*End

C Setting the Maximum Dimensions

The FORTRAN 77 include file `maxdim.inc` contains information about the size of various arrays used in the BEASTS software. The meaning of the parameters is explained within the file.

Notice that some of the parameters are used to control the size of arrays that are used to store finite element and point stiffness element data. Elements of these types are not yet implemented in BEASTS, but the facilities to process input data for finite and point stiffness elements are to some extend included in FLIES.

```
c ====================================================================
c     includefile maxdim
c ====================================================================

c   Sets the maximum dimensions for arrays used in the three dimensional
c   boundary elements program BEASTS. Also sets some parameters which
c   are used for the special determination of the diagonal Hs terms.

c --------------------------------------------------------------------

c   The parameters are:

c     nnmax - Max. no. global nodes
c     plmax - Max. no. point (nodal) loads
c     bdmax - Max. no. boundary element domains
c     pemax - Max. no. point stiffness elements
c     femax - Max. no. finite elements
c     bnmax - Max. no. nodes in a single boundary element domain
c     bemax - Max. no. elements in a single boundary element domain
c     nfmax - Max. no. frequencies
c     mpmax - Max. no. material parameter sets
c     fesbw - Semi-bandwidth of finite element stiffness matrix Kfe
c     aemax - Max. no. adjacent elements to a single node in a BE domain

c     eitvd - Elements in the 'vertical' direction      (used by HsDiag)
c     lnmax - Max. no. nodes on local closed surface    (used by HsDiag)
c     lemax - Max. no. local enclosing elements         (used by HsDiag)

c --------------------------------------------------------------------

c   Original code by L. Andersen 2001

c ====================================================================
c Define Dimension Integers
c ====================================================================

      integer nnmax,plmax,bdmax,pemax,femax,bnmax,bemax
      integer nfmax,mpmax,fesbw,aemax,eitvd,lnmax,lemax
```

```
c ==================================================================
c Set Parameters
c ==================================================================

      parameter(nnmax = 1400) ! Maximum number of global nodes

      parameter(plmax =  500) ! Maximum number point (nodal) loads

      parameter(bdmax =    5) ! Maximum number of BE domains

      parameter(pemax =  500) ! Maximum number of point stiffness
                              ! elements

      parameter(femax =  500) ! Maximum number of finite elements

      parameter(bnmax = 1000) ! Maximum number of nodes in a single
                              ! boundary element domain

      parameter(bemax =  250) ! Maximum number of elements in a single
                              ! boundary element domain

      parameter(nfmax =  128) ! Maximum number of frequencies

      parameter(mpmax =   10) ! Maximum number of material param. sets

      parameter(fesbw =  100) ! Semi-bandwidth of the FE matrices

      parameter(aemax =   12) ! Maximum number of adjacent boundary
                              ! elements to a single node in each domain

c ------------------------------------------------------------------
c  The following parameters are only used in the determination of the
c  diagonal terms of Hs matrix for each of the boundary element domains
c ------------------------------------------------------------------

      parameter(eitvd =    1) ! Number of elements in the 'vertical'
                              ! direction on the false local boundaries
                              ! used for evaluation of the Hs diagonal

      parameter(lnmax =   98) ! Maximum number of nodes on each false
                              ! local closed boundary
                              ! § lnmax = 4*(eitvd+1)*aemax+2

      parameter(lemax =   36) ! Maximum number of elements on each false
                              ! local closed boundary
                              ! § lemax = (2+eitvd)*12

c ==================================================================
c End of includefile maxdim
c ==================================================================
```

Index