isvr Institute of Sound and Vibration Research

Manual for WANDS 2.1
Wavenumber Domain Software for Solids and Fluids

C-M. Nilsson and C.J.C. Jones

ISVR Technical Memorandum No 976

August 2007

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

Manual for WANDS 2.1
WaveNumber domain software
for solids and Fluids

by

C-M. Nilsson and C.J.C Jones

ISVR Technical Memorandum No: 976

August 2007

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

MANUAL FOR
WANDS 2.1
WAVENUMBER DOMAIN SOFTWARE FOR SOLIDS AND FLUIDS

Carl-Magnus Nilsson and Chris Jones

Contents

Chapter 1

Introduction

Warnings! It is highly recommended that the reader takes a look at the warning statements that are given throughout this manual, (i.e. the 'Index' at the end).

One general warning concerning all content of this manual is that coupling between different models must be checked very carefully. The sign of coupling between fluids and plates and solids depends both on node numbering and specifications of internal coordinates. Although a node reversal option may me used to fix problems concerning wrong signs.

1.1 Theory

The theory and structuring of WANDS is detailed in the theory manual [1].

1.2 Submodels

The current version of the WANDS software produce one or several matrices for different submodels. These matrices are either used in WANDS to calculate the solutions of a problem or they are written to output files.

Each basic submodel is defined by the element types that it can use. In addition there are coupling submodels that couples the basic submodels to each other.

1.2.1 Basic submodel elements

Element types currently included in WANDS are:

- Solid finite elements for iso- or ortho- tropic materials.
- Plate finite elements for iso- or ortho- tropic materials.
- Beam 'finite elements'
- Fluid boundary elements

The solid elements are of different types, in that either quadrilateral or triangular elements may be used and that different orders of interpolations are used (linear, quadratic and cubic).

However, there is only one plate element type.

Beam elements can be directly added to the matrices of either a solid or a plate submodel without the need of any special coupling matrices. Therefore beam elements are included as integral part of either a solid submodel or a plate submodel and not as submodels in their own right.

The fluid boundary elements are also either linear, quadratic or cubic. They can also be used both for internal or exterior problems.

In addition damping properties and restraints for some degrees of freedom are included.

The future version of WANDS will include fluid finite elements and solid boundary elements.

1.2.2 Coupling submodels

To couple the basic submodels, a large number of coupling submodels are included in WANDS. The coupling models included are:

- Coupling between plate finite elements and solid finite elements.

- Coupling between plate finite elements and fluid boundary elements.
- Coupling between solid finite elements and fluid boundary elements.
- Coupling between fluid boundary elements of different submodels .

Future versions will also enable couplings to fluid finite elements and to solid boundary elements.

1.2.3 Fluid and solid boundary elements

There are a few differences between the solid boundary elements and the fluid boundary elements.

The only implemented solid boundary element is a quadratic element that is largely copied from programs made by X Sheng. For the fluid boundary element model there is a choice between linear, quadratic and cubic interpolation.

Both types of boundary elements are to be numbered counterclockwise to enclose a solid or fluid domain respectively.

The coordinate system for a solid boundary element is defined as in Figure 1.1.

The coordinate system for a fluid boundary element should be such that the normal velocity is defined positive out of the fluid domain.

Note! The directions for all possible cases have not been properly tested, so there might be bugs concerning the directions.

If we require that the solid always is damped there is no need for CHIEF points for the solid elements. Thus CHIEF points are not included.

The singular points where the boundary integral is evaluated at the collocation points, i.e. the diagonal entries in the BE matrices, are evaluated numerically for the solid boundary elements and analytically for the fluid BE matrices. This means that the type of problem must be specified for the fluid BE, i.e. if the problem is external, internal or half-infinite. This is not needed for the

There are no field point calculations or power calculation implemented in the solid boundary element.

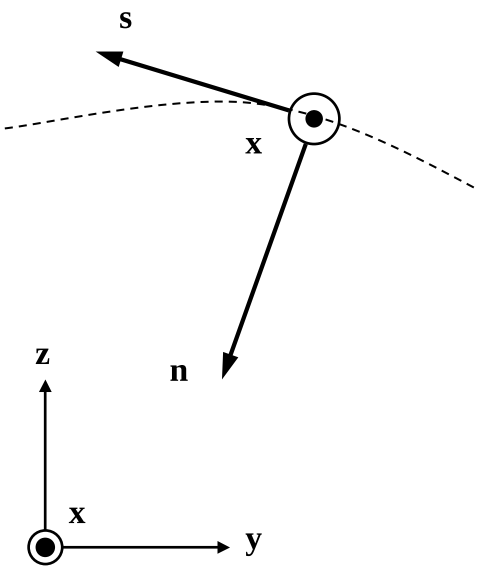

Figure 1.1: Coordinate system of solid boundary element compared to global coordinate system; **x**, **s** and **n** are the first, second and third direction of the node respectively

Chapter 2

Input file

WANDS is written so that the user has to specify the system in detail in the input data file. The `*INSYS:` section specifies the solution type that should be used, the range of the solution and the submodels used in the model. The following sections specifies the different submodels. To separate the different submodels from each other they have a *tag* . As an example, the statement,

```
*INBEM: FLUID: TOP: F1
```

is a header specifying that the following lines specifies the topology for a fluid boundary element submodel with the tag `F1`.

Tags are used to distinguish between different submodels of the same type. A tag cannot be longer than 5 characters.

2.1 INSYS:

The `INSYS:` header specifies the solution type, the range of the solution and the submodels that are used in the model. This block is necessary to include at the top of the input file as it defines what submodels should be read from the rest of the input file.

INSYS: TITLE:

Is the header for the *title block* of the model. The title block, written as:

```
*INSYS: TITLE:


Rail 60-UIC
60UIC
```

could be suitable for telling that the model relates to a rail of the UIC-60 type. The second row is a "tag" for the system, this is a short name with maximum 5 characters. This tag will be used in parts of the output file names.

2.1.1 INSYS: SOLTYPE:

This block defines the solution type that should be used in the calculations. To date only one solution type is fully implemented in WANDS and a second is partly implemented. There are however several other solution types that can be used outside WANDS.

The first solution type is referred to as *double loop solution* and is further discussed in the theory manual [1]. This solution type is the only one that can solve strongly coupled FE-BE systems.

The second type is a part of something that may be referred to as a *pseudo 2D BE* solution. To be precise, this is only a description of the radiation problem from a vibrating structure into a fluid. The output from WANDS is a number of matrices that describes this radiation. The vibrations of the system must be calculated first. This can be made by using Waveguide FE matrices and coupling matrices also calculated from the same input file. To date only one BE-submodel can be used with this solution method.

The block is typically written:

Example:

```
*INSYS: SOLTYPE:
** double loop, be sol, (more to come)


 1, 1, 0
```

Here the first `1` means that a ”double loop” solution is requested. The second `1` shows that also the ”pseudo 2D BE” solution is to be calculated. This is done after the ”double loop” solution is calculated and then only for the highest frequency. The `0` is reserved for a future solution method and has no meaning, it must however be included.

2.1.2 INSYS: WAVEDOMAIN:

This block specifies **both** the wavedomain and the frequency domain of the problem. The input is written as,

Example:

```
*INSYS: WAVEDOMAIN:
** frequency, real wavenumb, imag wavenumb
500.000,   0.01,   0.00
1000.000,   0.01,   0.00
1000.000,   0.02,   0.00
```

The first column describes the frequencies used in the calculations, the second column describes the real part of the *imposed* wavenumber and the third column describes the imaginary part of this wavenumber.

Note! The imposed wavenumbers are rarely complex, so the third column should be set to `0.0`.

2.1.3 INSYS: MODELS: FE-SOLID:

This block describes which solid FE submodels are included in the system. The first row is the name of the submodel. The second row is the submodel tag. The third row indicates which outputs are requested from this model.

Example:

```
*INSYS: MODELS: FE-SOLID:
** First row = name
** Second row = tag

RAIL UIC-60
 S1
 1, 1
```

The name of this model is, `RAIL UIC-60`. The tag, which must be included in all the following data relating to this model is `S1`. The first `1` indicates that the FE-matrices will be written. The second `1` indicates that some displacement solutions will be written (these are specified later).

2.1.4 INSYS: MODELS: FE-PLATE:

This input is totally analogous with

the `*INSYS: MODELS: FE-SOLID:` block but for plate submodels instead of the solid submodels.

2.1.5 INSYS: MODELS: FE-FLUID:

This input is also analogous with

the `*INSYS: MODELS: FE-SOLID:` block but for fluid submodels instead of the solid submodels.

2.1.6 INSYS: MODELS: BE-FLUID:

The fluid-be models are initialized with the,

```
*INSYS: MODELS: BE-FLUID:
```

block. The first row of data after this header is the name of the submodel. The second row is the tag and the third row contains information about which output data is to be written. These data applies to the first solution type (double loop).

The first digit signals that the matrices for the BE model should be written. However since there are possibly several hundred matrices associated with different wavenumbers and frequencies only those associated with the first frequency wavenumber pair are written. These matrices are the **H** and **G** matrices for the fluid velocity potential, as discussed in the theory manual, [1].

The second digit specifies that some solution data is to be written, this can be the pressure and normal velocity on the boundary or the pressure in a field point not on the boundary.

The third digit indicates that the pressure and radiation efficiency should be written. It may be worth noting that for a BE submodel with a pressure field impinging on the boundary both these quantities may be negative.

Example:

```
*INSYS: MODELS: BE-FLUID:
** First row = name
** Second row = tag

AIR
F1
 0, 0, 1

WATER
 F2
 0,1,0
```

means that there is a submodel "AIR" with the tag "F1". The only data that is requested from this model is the power output for each frequency and wavenumber.

In addition there is a submodel `WATER` with the tag `F2`, for which results for the field variables have to be written. (Exactly which ones are specified later)

2.1.7 INSYS: MODELS: BE-SOLID:

The solid-be models are initialized with the,

```
*INSYS: MODELS: BE-SOLID:
```

block. The first row of data after this header is the name of the submodel. The second row is the tag and the third row contains information about which output data is to be written. These data applies to the first solution type (double loop).

The first digit signals that the matrices for the BE model should be written. However since there are possibly several hundred matrices associated with different wavenumbers and frequencies only those associated with the first frequency wavenumber pair are written. These matrices are the **H** and **G** matrices for the solid as discussed in the theory manual.

The second digit specifies that some boundary solution data is to be written. Unlike the fluid boundary elements there are no options for field points or power for the solid boundary element models. However three inputs are still needed.

Example:

```
*INSYS: MODELS: BE-SOLID:
** First row = name
** Second row = tag

SOIL
S1
 0, 1, 1

CLAY
S2
 1,1,0
```

means that there is a submodel "`SOIL`" with the tag "`S1`". Data written from this model is solutions on the boundary. The last "`1`" does not mean anything, but should still be included. In addition there is a submodel `CLAY` with the tag `S2`, for which results both on the boundary and the BE matrices will be written.

2.1.8 INSYS: MODELS: COUP:

There are several different coupling models. Currently the following types are implemented in WANDS.

```
INSYS: MODELS: COUP: FE-PLATE: BE-FLUID:

INSYS: MODELS: COUP: BE-FLUID: BE-FLUID:

INSYS: MODELS: COUP: BE-SOLID: BE-SOLID:

INSYS: MODELS: COUP: BE-SOLID: BE-FLUID:

INSYS: MODELS: COUP: FE-SOLID: BE-FLUID:

INSYS: MODELS: COUP: FE-PLATE: FE-SOLID:

INSYS: MODELS: COUP: FE-PLATE: FE-FLUID:

INSYS: MODELS: COUP: FE-SOLID: FE-FLUID:

INSYS: MODELS: COUP: FE-SOLID: BE-SOLID:

INSYS: MODELS: COUP: FE-PLATE: BE-SOLID:
```

As for other initialization blocks, the first data row after the heading is the name of the submodel.

The second line contains three ”tags”. The first is the tag of the coupling model. The second and third are the the tags of the submodels that should be coupled together.

The coupling matrices, as defined in the theory manual, are written if the number in the third line is not equal to zero.

Example:

The statements,

```
*INSYS: MODELS: COUP: FE-SOLID: BE-FLUID:
** First row = name
** Second row = coupled systems (plate system #)&(fluid system #)

COUPLING TO FLUID 1

S1F1, S1 , F1

 1
```

initializes a submodel with the name `COUPLING TO FLUID 1` and tag ”`S1F1`”, that couples the solid-fe submodel with tag ”`S1`” to the fluid-be submodel with the tag ”`F1`”.

Note!

Here the tag `S1F1` is chosen for this coupling submodel, because it is self explanatory. However it could just as well have been tagged `ABC`, (or something else with between 1 and 5 characters).

2.1.9 INSYS: END:

The system input data must be ended with the end statement,

```
*INSYS: END:
```

2.2 INFEM:

The input data for different finite element submodels are written very much as shown in the WAFER manual. There are however some differences. For completeness the entire input data options are given in the following.

2.2.1 INFEM: SOLID:

For a solid submodel to be made this model has to include the following blocks:

`*INFEM: SOLID: COORD:`

,specifying the coordinates needed for the solid model.

`*INFEM: SOLID: TOP:`

,specifying the topology needed for the solid model and either or both of the material specifying blocks:

`*INFEM: SOLID: MAT: ISO:`

or

`*INFEM: SOLID: MAT: ORTHO:`

To include restraints of degrees of freedom, the block:

`*INFEM: SOLID: RESTR:`

is used.

INFEM: SOLID: COORD:

The `'*INFEM: SOLID: COORD:'` block defines the coordinates for the solid element model. The y- and z- coordinates for the first node are written on the first row of the block, the coordinates for the second node on the second row, etc.

Example:

```
*INFEM: SOLID: COORD: S1
 **nodes on solid model, y- and z- coordinates
```

```
0.250000,       0.010000
0.250000,       0.000000
0.250000,      -0.010000 ...
```

indicates that, for the submodel tagged `S1`, the first node has the y–coordinate = 0.25 and the z–coordinate = 0.01.

INFEM: SOLID: TOP:

The `'*INFEM: SOLID: TOP:'` block defines the topology for the solid model. Thus it specifies the data needed to form the individual elements. The data for the first element is written on the first row, the data for the second element on the second and so on.

The first entry on a line is a number specifying the element type. The available element types are given in Table 2.1.

element type number	element shape	number of nodes	interpolation
1	quadrilateral	4	linear
2	triangle	3	linear
3	quadrilateral	12	cubic
4	triangle	10	cubic
5	quadrilateral	8	quadratic
6	triangle	6	quadratic

Table 2.1: Table of available solid element types

The second entry on a line is the material number that specifies the set of material data to be used. The materials specification block is given in Section 2.2.1.

The following entries on a line specifies the nodes onto which the element is connected. The node numbers for linear and quadratic elements are always given counter clockwise about the element, see Figures 2.1 and 2.2.

Warning! Only quadrilateral elements with linear and quadratic interpolation has been tested by the author. The triangular elements with linear and quadratic interpolation is reported in [2] and should also be safe to use. The elements with

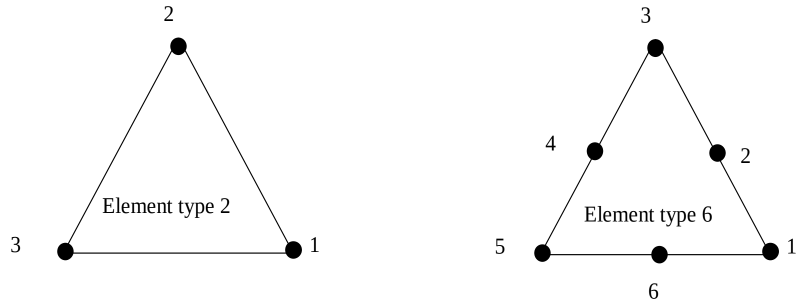

Figure 2.1: Triangular elements and their node numbering

cubic interpolation seem to be untested, and their node numbering is also uncertain. Probably the node numbering corresponds to the standard given in [3].

Example:

```
*INFEM: SOLID: TOP: S1
** eletype, material, nodes
5,    1,    8,    7,    6,    5,    1,    2,    3,    4
6,    1,    9,   10,    6,    7,    8,   11,
...
```

Note! Each line must have the correct number of entries.

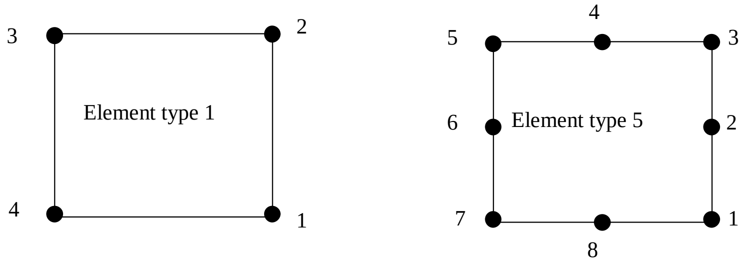

Figure 2.2: Quadrilateral elements and their node numbering

INFEM: SOLID: MAT:

INFEM: SOLID: MAT: ISO:

To specify an isotropic material for a solid model the block header:

```
*INFEM: SOLID: MAT: ISO: TAG
```

is used. Each row specifies one material type, as follows,

,

Thus a typical steel material (in SI-units) is specified by the following example.

Example:

Entry number	Entry name
1	Material number
2	E = Young's modulus
3	ν = Poisson's ratio
4	ρ = Density
5	η = Loss factor

Table 2.2: Isotropic material data order

```
*INFEM: SOLID: MAT: ISO: S2
 1,  2.100e+011,   0.300,  7800.000,   0.01
```

INFEM: SOLID: MAT: ORTHO:

An orthotropic material is specified by the block header:

```
*INFEM: SOLID: MAT: ORTHO: TAG
```

The entries on each row are as follows:

Entry number	Entry name
1	Material number
2	E_{xx}
3	E_{yy}
4	E_{zz}
5	E_{xy}
6	E_{xz}
7	E_{yz}
8	G_{xy}
9	G_{xz}
10	G_{yz}
11	ρ
12	η

Table 2.3: Orthotropic material data order

The meaning of entries number 2 to 10 are given in the theory report, [1], ρ is the material density and η is the loss factor common for all material strains.

If loss factors that are independent in different directions are to be used, entry number 12 is replaced by the loss factor corresponding to E_{xx} and other entries (number 13 to 20) corresponding to E_{yy}, E_{zz}, E_{xy} , etc. , are added. The software will recognize the number of entries written to a row and chose the right damping model.

Note! Each line must have either 12 or 20 entries.

INFEM: SOLID: RESTR:

Blocking some degrees of freedom is made by using the:

```
*INFEM: SOLID: RESTR: TAG
```

data block. The information is given in the order of:

- Element number of the element to be restrained
- Local node number of the element to be restrained
- Local degrees of freedom numbers of the node to be restrained

The restrained degrees of freedom are given with the numbers:

1	= restrained displacement in the x-direction
2	= restrained displacement in the y-direction
3	= restrained displacement in the z-direction

Table 2.4: Restrained displacements for solid

Any displacement of the node that is unrestrained is replaced with a zero (0).

Example:

```
*INFEM: SOLID: RESTR: S2
 7,     6,     0,     2,     3
 7,     5,    1,    0,    0
```

,restrains the local nodes 6 and 5 of the 7:th solid element in the submodel tagged S2. The degrees of freedom restrained are, the y- and z- displacements of node 6 and the x- displacement of node 5.

INFEM: SOLID: BEAM: MAT:

The beam data input starts with the block header:

```
*INFEM: SOLID: BEAM: MAT:
```

The entries on the following lines are given in Table 2.5

Entry number	Entry name
1	Material number
2	D_{fy}
3	D_{fz}
4	D_{fyz}
5	Dt
6	Dl
7	Dtl
8	M
9	Mt
10	η_b

Table 2.5: Beam material input data

D_{fy} and D_{fz} are the bending rigidities for the flexural motion in the y– and z– directions, respectively. D_{fyz} is the coupling term between these to motions. D_l and D_t are the longitudinal and torsional rigidities respectively and D_{tl} is their coupling term. M is the mass per unit length and Mt is the rotational moment of inertia . η_b is the loss factor for all types of motions.

Note! The y– and z– directions here refer to the global coordinate system. This is contrary to the cases for the material specifications of plate and solid elements where they refer to the local coordinates. This is because there is no rotation made (or required) before assembling the single node of a beam element.

INFEM: SOLID: BEAM: TOP:

The coupling to a solid is given in the block,

```
*INFEM: BEAM: TOP: SOLID:
```

The first entry on a line in this block gives the 'material' number, with beam data as described above.

The second entry gives the element number of the solid element.

The third and fourth entries gives the local node numbers of the element.

The fifth entry gives the local (non–dimensional) coordinate, ξ, between these nodes. This coordinate is defined such that, $\xi = -1$ corresponds to the first node and $\xi = 1$ corresponds to the second. However, $\xi = \pm 1$ may not be used, since this the rotation at a solid element node is undefined.

Entry number	Entry name
1	Material number
2	Solid element number
3	First local node of solid element
4	Second local node of solid element
5	Non dimensional coordinate between nodes

Table 2.6: Beam–solid topology input data

Example:

```
*INFEM: SOLID: BEAM: TOP: S2
 1, 2, 3, 5,0.0000
```

, corresponds to a beam with beam 'material' data number 1, which is placed in the middle ($\xi = 0.00$) between the local nodes node 3 and 5 of element 2.

INFEM: SOLID: FORCE:

To define a force acting on a plate finite element submodel, the heading,

```
*INFEM: SOLID: FORCE: TAG
```

is used. The order of the data input for each force is given in Table 2.13

Entry number	Entry name
1	element number
2	Node number
3	Direction of force
4	Real part of force
5	Imaginary part of force

Table 2.7: Plate force input data

Example:

```
*INFEM: SOLID: FORCE: S2
```

```
18,   5,  1,  2.0,   2.0
```

Defines a force on the solid submodel tagged `S2`. The force acts on element number 18 and the local element node 5 in the x-direction and with the magnitude 4 and the phase +45 degrees, (i.e. $f = 2 + 2i$).

INFEM: SOLID: WRITE: WAVEDOMAIN:

Outputs for all wavenumbers and frequencies will give very large output files. Therefore only those frequencies and wavenumbers that are specified will result in written outputs.

Example:

The,

```
*INFEM: SOLID: WRITE: WAVEDOMAIN: S1
```

```
202,  1.1,  0.0
```

```
202,  1.3,  0.0
```

means that the outputs are written either if the frequency is 202 and the wavenumber is 1.1 or 1.3. The last figure, 0.0, is the imaginary part of the wavenumber. The solution is only written if the corresponding frequency-wavenumber pair is given in the `INSYS: WAVEDOMAIN:` block.

INFEM: SOLID: WRITE: TOP:

This block specifies the degrees of freedom for which solutions are written. The order in which the input is written is the same as for specifying forces. Hence, *element, local node, direction.*

Number	Displacement
1	x-displacement
2	y-displacement
3	z-displacement

Table 2.8: Numbers associated with displacements

Example:

```
*INFEM: SOLID: WRITE: TOP: S1
17,   6,  2
```

means that the y-displacement for local node 6 of the 17:th element is written.

INFEM: SOLID: END:

A plate submodel tagged `S3` is ended with the,

```
*INFEM: PLATE: END: S3
```

statement.

2.2.2 INFEM: PLATE:

The indata blocks for the plate model are very similar to the indata for the solid model. The indata blocks for the plate model are,

```
*INFEM: PLATE: COORD:

*INFEM: PLATE: TOP:

*INFEM: PLATE: MAT: ISO:

*INFEM: PLATE: MAT: ORTHO:

*INFEM: PLATE: RESTR:
```

and, not yet implemented,

```
*INFEM: PLATE: SYM:
```

INFEM: PLATE: COORD:

The '*INFEM: PLATE: COORD:' block defines the coordinates for the plate element model. The y- and z- coordinates for the first node are written on the first row of the block, the coordinates for the second node on the second row, etc.

Example:

```
*INFEM: PLATE: COORD:
 **nodes on plate model, y- and z- coordinates
 1.250000,        0.010000
 1.250000,        0.010000
 1.250000,        0.000000
 1.250000,       -0.010000
```

...

This is completely analogous with the coordinate block for the solid elements.

INFEM: PLATE: TOP:

The '*INFEM: PLATE: TOP:' block defines the topology for the solid model. Thus it specifies the data needed to form the individual elements. The data for the first element is written on the first row, the data for the second element on the second row and so on.

The first entry on a line is a number specifying the element type. At present there is only one plate element type. This element type has the element number 11.

The second entry on a line specifies which set of material data to be used. The materials specification block is given in the next section.

The following entries on a line specifies the two nodes onto which the element is connected.

Example:

```
*INFEM: PLATE: TOP: P1
 **FE elements on plate
   11,    1,    1,    2
   11,    1,    2,    3
   11,    1,    3,    4
```

The above connects nodes, 1–2, in the first element, nodes, 2–3, in the second and nodes, 3–4, in the third element. All these elements use material number 1.

INFEM: PLATE: MAT:

INFEM: PLATE: MAT: ISO:

To specify an isotropic material for a solid model the block header:

```
*INFEM: PLATE: MAT: ISO:
```

is used. Each row specifies one material type, as follows,

Thus a typical steel plate with thickness = $2mm$ is specified by the following example.

Entry number	Entry name
1	Material number
2	E = Young's modulus
3	ν = Poisson's ratio
4	h = Plate thickness
5	ρ = Density
6	η = Loss factor

Table 2.9: Isotropic material data order

Example:

```
*INFEM: PLATE: MAT: ISO:
 1,  2.100e+011,    0.300, 0.002, 7800.000,   0.01
```

where all data is given in SI-units.

INFEM: PLATE: MAT: ORTHO:

An orthotropic material is specified by the block header:

```
*INFEM: PLATE: MAT: ORTHO:
```

The entries on a row are as follows:

where, E_{xx} and E_{yy} are Young's modulus in the elements local x– and y–directions respectively. ν_y is the Poisson's ratio , such that the off diagonal term in the plate material stiffness becomes, $E_{xx}\nu_y$. h is the plate thickness , ρ is the material density and η is the loss factor common for all material strains. These material data are also described in the theory manual, [1].

Alternatively the damping may be specified individually. If so, entry number 8 is the damping, η_{xx}, corresponding to E_{xx}, i.e. $E_{xx} \rightarrow E_{xx}(1 + i\eta_{xx})$. Similarly the damping corresponding to E_{yy} and G_{xy} are given by entries number 9 and 10, whereas entry number 11 corresponds to the damping, $eta_{\nu_{xy}}$, so that $E_{xx}\nu_y \rightarrow E_{xx}\nu_y(1 + i\eta_{\nu_{xy}})$.

Entry number	Entry name
1	Material number
2	E_{xx}
3	E_{yy}
4	G_{xy}
5	ν_y
6	h
7	ρ
8	η

Table 2.10: Orthotropic material data order

Note! Each line must have either 8 or 11 entries.

INFEM: PLATE: RESTR:

Blocking some degrees of freedom in the plate model is made by the:

```
INFEM: PLATE: RESTR:
```

data block. The information is given in the order of:

- Element number of the element to be restrained
- Local node number of the element to be restrained
- Local degrees of freedom numbers of the node to be restrained

The restrained degrees of freedom are given with the numbers:

As for the solid case, any displacement of the node that is unrestrained is replaced with a zero (0).

Example:

1	= restrained displacement in the x–direction
2	= restrained displacement in the y–direction
3	= restrained displacement in the z–direction
4	= restrained rotation about the x–direction

Table 2.11: Restrained displacements for plate

```
*INFEM: PLATE: RESTR:
 10,     2,     1,    2,    3,    0
```

,restrains the local node 2 of the 10:th plate element. The degrees of freedom restrained are, the x– y– and z– displacements of this node, whereas the rotation remains free.

INFEM: PLATE: BEAM:

The data blocks used to specify a beam in a plate submodel are,

```
*INFEM: PLATE: BEAM: MAT:
```

```
*INFEM: PLATE: BEAM: TOP:
```

and as usual the headings are preceded with an * and followed by the tag of the submodel.

The first of the blocks above specifies the properties of a beam with Euler–type bending, longitudinal motion and rotation. The second and third of these blocks specifies the beam's coupling to the solid or plate model respectively.

INFEM: PLATE: BEAM: MAT:

This data input is totally analogous with that defining beam data for a beam connected to a solid. The block header is however replaced with:

```
*INFEM: PLATE: BEAM: MAT:
```

The entries on the following lines are defined in 2.2.1.

INFEM: PLATE: BEAM: TOP:

This places a beam on a node-line of a plate model. The block header is,

```
*INFEM: BEAM: TOP: PLATE: TAG
```

and the entries are given in Table 2.12

Entry number	Entry name
1	Material number
2	Solid element number
3	First local node of solid element
4	Second local node of solid element
5	Non dimensional coordinate between nodes

Table 2.12: Beam–plate topology input data

The beam 'material' number.

The plate element number.

The local node of the plate (1 or 2).

Example:

```
*INFEM: PLATE: BEAM: TOP: P1  3, 5, 2
```

Places a beam with 'material' number 3 on the second node of plate element 5.

INFEM: PLATE: FORCE:

To define a force acting on a plate finite element submodel, the heading,

```
*INFEM: PLATE: FORCE: TAG
```

is used. The order of the data input for each force is given in Table 2.13

The directions of the force is given with the number 1 to 4 according to Table 2.14

Example:

```
*INFEM: PLATE: FORCE: P1
```

Entry number	Entry name
1	element number
2	Node number
3	Direction of force
4	Real part of force
5	Imaginary part of force

Table 2.13: Plate force input data

Number	Displacement
1	x-displacement
2	y-displacement
3	z-displacement
4	rotation (about x-axis)

Table 2.14: Numbers associated with displacements

```
12,   1,  3,  1.22,   0.66

12,   2,  4,  1.0,    0
```

Defines two forces on the plate submodel tagged `P1`. Both forces act on element number 12. The first force acts on the local element node 1 in the z-direction and with the magnitude $f = 1.22 + 0.66$. The second 'force' is a unit moment (torque) about node 2.

INFEM: PLATE: WRITE: WAVEDOMAIN:

Outputs for all wavenumbers and frequencies will give very large output files. Therefore only those frequencies and wavenumbers that are specified will result in written outputs.

Example:

The,

```
*INFEM: PLATE: WRITE: WAVEDOMAIN: P1
```

```
103,  1.1,  0.0
```

```
103,  1.2,  0.0
```

means that the outputs are written either if the frequency is 103 and the wavenumber is 1.1 or 1.2. The last figure, 0.0, is the imaginary part of the wavenumber. The solution is only written if the corresponding frequency-wavenumber pair is given in the `INSYS: WAVEDOMAIN:` block.

INFEM: PLATE: WRITE: TOP:

This block specifies the degrees of freedom for which solutions are written. The order in which the input is written is the same as for specifying forces. Hence, *element, local node, direction.*

Number	Displacement
1	x-displacement
2	y-displacement
3	z-displacement
4	rotation (about x-axis)

Table 2.15: Numbers associated with displacements

Example:

```
*INFEM: PLATE: WRITE: TOP: P2
```

```
14,  1,  4
```

means that the rotation about the x-axis for local node 1 of the 14:th element is written.

INFEM: PLATE: END:

A plate submodel tagged `P1` is ended with the,

```
*INFEM: PLATE: END: P1
```

statement.

2.2.3 INFEM: FLUID:

For a fluid submodel to be made this model has to include the following blocks:

`*INFEM: FLUID: COORD:`

,specifying the coordinates needed for the solid model.

`*INFEM: FLUID: TOP:`

,specifying the topology needed for the solid model and either or both of the material specifying blocks:

`*INFEM: FLUID: MAT:`

which also includes a scaling factor that may be needed for improving the condition number of the system matrix.

`*INFEM: FLUID: RESTR:`

is used to restrain the pressure of some nodes. The mathematical formulation means that unrestrained nodes are seen as connected to rigid walls, whereas restrained nodes are connected to a zero pressure surface.

INFEM: FLUID: COORD:

The `'*INFEM: FLUID: COORD:'` block defines the coordinates for the fluid element model. The y- and z- coordinates for the first node are written on the first row of the block, the coordinates for the second node on the second row, etc.

Example:

```
*INFEM: FLUID: COORD: F1
 **nodes on solid model, y- and z- coordinates
 0.250000,        0.010000
 0.250000,        0.000000
 0.250000,       -0.010000 ...
```

indicates that, for the submodel tagged `F1`, the first node has the y–coordinate = 0.25 and the z–coordinate = 0.01.

INFEM: FLUID: TOP:

The '*INFEM: FLUID: TOP:' block defines the topology for the fluid model. Thus it specifies the data needed to form the individual elements. The data for the first element is written on the first row, the data for the second element on the second and so on.

The first entry on a line is a number specifying the element type. The available element types are the same as for solid elements and they are repeated in Table 2.16.

element type number	element shape	number of nodes	interpolation
1	quadrilateral	4	linear
2	triangle	3	linear
3	quadrilateral	12	cubic
4	triangle	10	cubic
5	quadrilateral	8	quadratic
6	triangle	6	quadratic

Table 2.16: Table of available fluid element types

The second entry on a line is the material number that specifies the set of material data to be used. The materials specification block is given in Section 2.2.3.

The following entries on a line specifies the nodes onto which the element is connected. The node numbers for linear and quadratic elements are always given counter clockwise about the element, see Figures 2.1 and 2.2.

Warning! Only linear triangular elements has been tested by the author. Since the code has been largely copied from the solid element implementation the quadrilateral linear quadratic elements should also be safe to use.

Example:

```
*INFEM: FLUID: TOP: F1
** eletype, material, nodes
5,    1,    8,    7,    6,    5,    1,    2,    3,    4
6,    1,    9,   10,    6,    7,    8,   11,
```

. . .

Note! Each line must have the correct number of entries.

INFEM: FLUID: MAT:

To specify the fluids properties for a fluid fe model, the block header:

```
*INFEM: FLUID: MAT: TAG
```

is used. Each row specifies one material type, as follows,

Entry number	Entry name
1	Material number
2	ρ = Density
3	c_f = Plane velocity
4	η = Loss factor
5	μ = Scaling constant

Table 2.17: Fluid properties data order

,

Thus normal air is specified by the following example.

Example:

```
*INFEM: FLUID: MAT: F2
 1,  1.21,  343,   0.001,  1000
```

As a rule of thumb, the scaling constant for air is 1000. To check if the problem is ill-conditioned the system matrix can be exported e.g. to matlab and where the conditioning number can be estimated. If the fluid is not coupled to a structure the scaling constant will not be needed.

Note! Each line must have 5 entries.

INFEM: FLUID: RESTR:

Blocking some degrees of freedom is made by using the:

```
*INFEM: FLUID: RESTR: TAG
```

data block. The information is given in the order of:

- Element number of the element to be restrained
- Local node number of the element to be restrained

Example:

```
*INFEM: FLUID: RESTR: F2
 7,     6
 7,     5
```

,restrains the local nodes 6 and 5 of the 7:th solid element in the submodel tagged `F2`.

INFEM: FLUID: FORCE:

At the moment force input is not included.

INFEM: FLUID: WRITE:

At the moment either all or no outputs are written.

INFEM: FLUID: END:

A plate submodel tagged `F1` is ended with the,

```
*INFEM: PLATE: END: F1
```

statement.

2.3 INBEM:

2.3.1 INBEM: SOLID:

INBEM: SOLID: MAT:

This block specifies the material data for the solid in the subsystem. The material data is given in the same order as for isotropic materials for finite elements. Thus in the order:

Young's modulus E, Poisson's ratio ν ,density, ρ and the damping, η.

Example:

```
*INBEM: SOLID: MAT: S1
** material number, Youngs mod, Poissons ratio, Density, Loss factor
 7.000e+010,   0.300,2.700e+003,   0.020
```

Note! Since there can be only one material there is no specification of the material number.

INBEM: SOLID: NODES:

This specifies the nodes of the be-submodel. The y- and z-coordinates are written. The node number is never given, but assumed from the position in the list.

Example:

```
*INBEM: SOLID: NODES: S2

0.0,    0.1
0.1,    0.1
0.2,    0.2
 0.3,    0.2
```

indicates that the submodel with tag S2 has the nodes.

Node	y-coord	z-coord
1	0.0	0.1
2	0.1	0.1
3	0.2	0.2
4	0.3	0.2

Table 2.18: Specifying be type

INBEM: SOLID: TOP:

This block specifies the topology of the be-submodel. Thus it specifies how the nodes are connected into elements. The node numbering is clockwise for an internal problem and anticlockwise for an internal problem. For an infinite problem the convention is to expand the domain into infinity and to consider the problem as an internal, thus applying counterclockwise node numbering.

Furthermore there are three different elements to chose from, linear, quadratic and cubic. The element type is decided from the number of nodes that are used for the element.

Example:

```
*INBEM: SOLID: TOP: F2

5, 4, 3

3, 2, 1
```

This describes two elements.

Note! Only quadratic elements are included for the solid boundary elements.

INBEM: SOLID: EDGE:

To suppress reflections at edges at infinite domains (half spaces) the option of edge elements are included. These elements, as most rest of the solid BE model are derived by X. Sheng and described in [2]. An edge element is added to two existing elements at the end of the BE domain. The input specifies the nodes that are to used for the edge element. The first five entries specify the nodes used starting with the node at the edge. The sixth entry specifies in which direction the edge element is to represent. If the sixth entry is `+1` then the edge suppress reflections from waves in the traveling towards the positive y-axis, if it is `-1` it suppress reflections from waves travelling towards the negative y-axis.

Example:

```
*INBEM: SOLID: EDGE: BES1
    7,  6,    5,    4,    3,    1
```

means that the edge element is placed on nodes `7,6,5,4` and `3`, the last `1` signals that the edge is at the right end of the half infinite space.

INBEM: SOLID: BC:

This describes the boundary conditions of the be-submodel for the nodes that are not coupled to another submodel. This data is defined by the equation,

$$cau + cbt = cc \tag{2.1}$$

where u is the displacement in the x, y or z direction and t is the traction vector in the same direction. ca, cb and cc are complex scalars defining the boundary condition. Hence, if a rigid boundary is considered we have,

$$ca = 1 \qquad cb = 0 \qquad cc = 0 \tag{2.2}$$

a boundary moving at a unit displacement is given by,

$$ca = 1 \qquad cb = 0 \qquad cc = 1 \tag{2.3}$$

and so on. The input is given as the node, the direction followed by the real and complex parts of the different coefficients and .

Input	entry number
Node	1
Direction	2
$\Re(ca)$	3
$\Im(ca)$	4
$\Re(cb)$	5
$\Im(cb)$	6
$\Re(cc)$	7
$\Im(cc)$	8
$\Im(cc)$	9

Table 2.19: Specifying boundary conditions

The direction is 1,2 or 3, where 1 denotes the x-direction, 2 denotes the tangential direction and 3 denotes the normal direction. The signs of these directions are given by Figure XX.

Example:

```
*INBEM: SOLID: BC: S2

5, 1,  0,  0,  1,  0,  2,  2

5, 2,  0,  0,  1,  0,  0,  0
```

Means that $ca = 0$, $cb = 1$ and $cc = 2 + 2i$ for node 5 in the x direction, hence a boundary moving with the amplitude $\sqrt{2^2 + 2^2}$ and phase 45 degrees ahead of the reference phase. The tangential direction is however connected to a rigid wall.

Note! It can sometimes be useful to introduce a scaling directly into equation (2.7) by multiplying all entries with some factor.

Note! The surface traction is a distributed force and the value at the node corresponds to the nodal degree of freedom. A force is thus depending on the length of the boundary element. This is different from how forces are implemented in a finite element model.

INBEM: SOLID: WRITE: WAVEDOMAIN:

Since writing all output data from the model would result in very large files, there are options to chose which data should be written.

The current block specifies for which frequencies and wavenumbers more complete data should be written. This is written as the frequency, the real part of the wavenumber and the imaginary part of the wavenumber.

Note!

The frequency and wavenumber must match exactly with those given in the `INSYS: WAVEDOMAIN:` (and complex wavenumbers are not recommended)

Example:

```
*INBEM: SOLID: WRITE: WAVEDOMAIN: S2

200,    1,  0 200,    2,  0
```

means that solution data for the be-subsystem tagged `F2`will be written for the frequency 200 Hz and wavenumbers 1 and 2.

INBEM: SOLID: WRITE: TOP:

This block defines the data on the boundary that should be written. The written information gives the element, the local node and the information that is to be written.

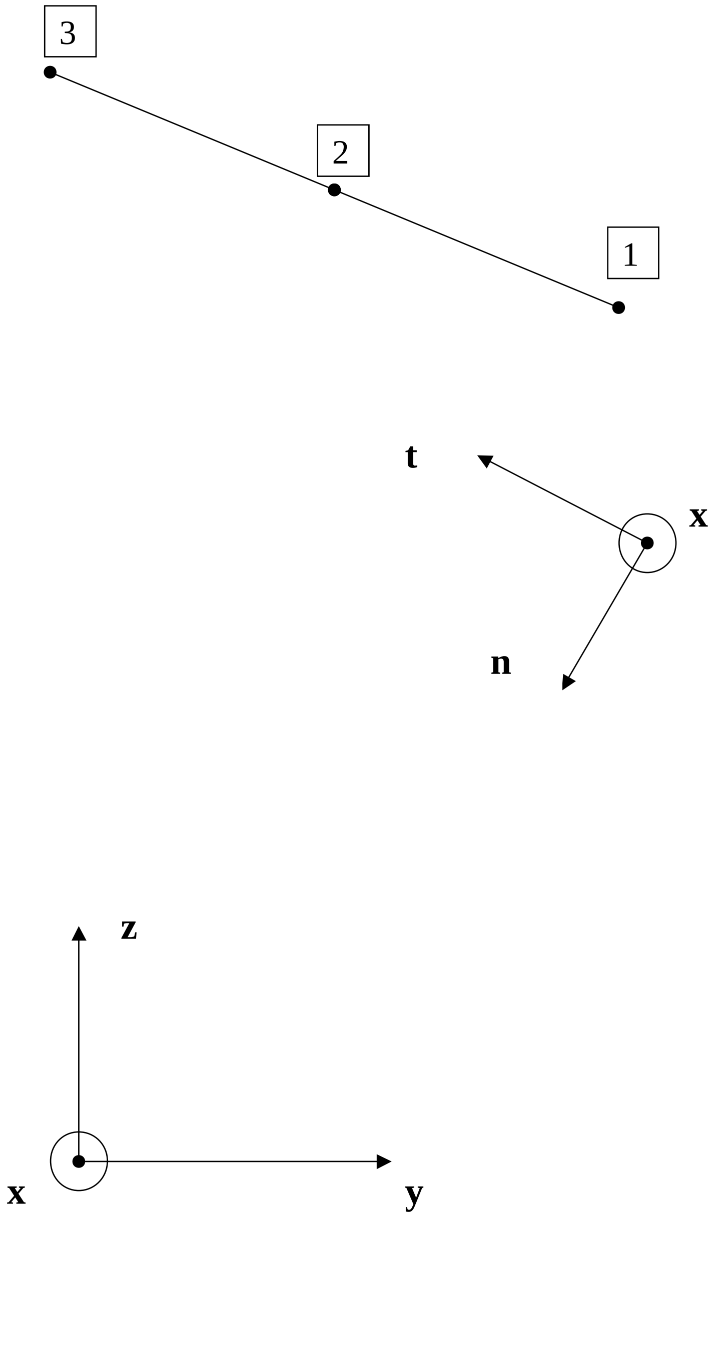

Figure 2.3: Co-ordinates for solid boundary element; The element has local nodes 1,2 and 3. The local co-ordinates are given in the directions x, t and n.

In each row:

- The first entry gives the element number.
- The second entry is the local node number.
- The third entry specifies what information should be written.

The latter information is defined by the numbers 1 to 6 as follows

Output	Number
x-displacement	1
t-displacement	2
n-displacement	3
x-traction	4
t-traction	5
n-traction	6

Table 2.20: Specifying output

Example:

```
*INBEM: SOLID: WRITE: TOP: F2


4,  3, 3
5,  2,  1
```

The first row of data means that the normal velocity of the third node in the fourth element is written. The second row means that the x-traction of the second node in the fifth element is written.

Note! The output dof is defined by a single number according with

x-displacements corresponding to

3n-2 for n=1,2..N

t-displacements corresponding to

3n-1 for n=1,2..N

n-displacements corresponding to

3n for n=1,2..N

where N is the number of nodes in the solid BE model

The dof numbers corresponding to the tractions are given by adding N to the above displacements.

INBEM: SOLID: END:

The data for each submodel must be ended with the

```
INBEM: SOLID: END:
```

statement.

2.3.2 INBEM: FLUID:

INBEM: FLUID: TYPE:

This block specifies if the problem is an interior, exterior or infinite problem. The first number in the row beneath the header signals that an interior problem is considered, the second number signals that an infinite problem is considered. There are three different types of problems, interior, exterior and infinite problems. Here "infinite" means that the boundary is assumed to extend to infinity.

The choice between them is given by specifying two logical inputs, as seen in Table 2.23

Example:

```
*INBEM: FLUID: TYPE: F1
** Internal flag & infinite
```

Type	1:st digit	2:nd digit
Interior	1	0
Exterior	0	0
Infinite	0	1

Table 2.21: Specifying be type

```
 0, 0
```

means, that the fluid-be submodel with tag `F1` defines an external problem.

INBEM: FLUID: INFIELD:

This specifies a pressure field impinging onto the the be-submodel. This field is composed of several plane fields that are given with their amplitude and their angle and possibly with their phase.

The angle is given in degrees and the angle zero corresponds to waves travelling along the positive y-axis (in the crosssectional y-z-plane).

If the phase is omitted a random phase is assumed. This option is useful when diffuse fields are simulated, since the phase then will change for each new wavenumber.

Example:

```
*INBEM: FLUID: INFIELD: F1

3.2,    45,     0
1.1,    90
```

indicates that there are two plane waves impinging on the submodel with tag `F1`. The first wave has an amplitude of 3.2 an angle of 45 degrees and a phase of 0 degrees. The second wave has an amplitude of 1.1 it is propagates in the positive z-axis and the phase is random and changes for each new wavenumber.

INBEM: FLUID: MAT:

This block specifies the material data for the fluid in the subsystem. The material data is given by the density, ρ, the plane wave velocity,c, and the damping, η. An optional fourth input gives the scaling coefficient, μ, that is used to improve numerical conditioning.

Thus, the velocity and pressure is defined by,

$$v = -\mu \nabla \Psi \tag{2.4}$$

and

$$p = i\omega\rho\mu\Psi \tag{2.5}$$

where Ψ is the fluid velocity potential used internally in the software.

Example:

```
*INBEM: FLUID: MAT: F2

1.21,   343,    0.001, 1000
```

would be typical properties for air with a scaling constant of 1000.

Note!

A small damping can be useful for problems involving coincidence, i.e. when $\kappa = k$, that otherwise would give infinite radiation efficiency.

INBEM: FLUID: NODES:

This specifies the nodes of the be-submodel. The y- and z-coordinates are written. The node number is never given, but assumed from the position in the list.

Example:

```
*INBEM: FLUID: NODES: F2


0.0,    0.1
0.1,    0.1
0.2,    0.2
0.3,    0.2
```

indicates that the submodel with tag F2 has the nodes.

Node	y-coord	z-coord
1	0.0	0.1
2	0.1	0.1
3	0.2	0.2
4	0.3	0.2

Table 2.22: Specifying be type

***INBEM: FLUID: TOP:**

This block specifies the topology of the be-submodel. Thus it specifies how the nodes are connected into elements. The node numbering is clockwise for an internal problem and anticlockwise for an internal problem. For an infinite problem the convention is to expand the domain into infinity and to consider the problem as an internal, thus applying counterclockwise node numbering.

Furthermore there are three different elements to chose from, linear, quadratic and cubic. The element type is decided from the number of nodes that are used for the element.

Example:

```
*INBEM: FLUID: TOP: F2

4, 3
3, 2
2, 1
```

This describes three two-noded elements, i.e. elements with linear interpolation.

Note! Currently the quadratic elements are best validated and are therefore recommended.

INBEM: FLUID: BC:

This describes the boundary conditions of the be-submodel for the nodes that are not coupled to another submodel. This data is defined by the equation,

$$cap + cbv_n = cc \tag{2.6}$$

where p is the pressure and v_n is the normal velocity to the boundary at the considered node. ca, cb and cc are complex scalars defining the boundary condition. Hence, if a rigid wall is considered we have,

$$ca = 0 \qquad cb = 1 \qquad cc = 0 \tag{2.7}$$

a boundary moving at a unit velocity is given by,

$$ca = 0 \qquad cb = 1 \qquad cc = 1 \tag{2.8}$$

and so on. The real and complex parts of the different coefficients are given as,

Example:

Input	entry number
Node	1
$\Re(ca)$	2
$\Im(ca)$	3
$\Re(cb)$	4
$\Im(cb)$	5
$\Re(cc)$	6
$\Im(cc)$	7

Table 2.23: Specifying boundary conditions

```
*INBEM: FLUID: BC: F2

5, 0, 0, 1, 0, 2, 2
6, 0, 0, 1, 0, 0, 0
```

Means that $ca = 0$, $cb = 1$ and $cc = 2+2i$ for node 5, hence a boundary moving with the amplitude $\sqrt{2^2+2^2}$ and phase 45 degrees ahead of the reference phase. Node 6 is however connected to a rigid wall.

Note! Although ca, cb and cc will be multiplied with the scaling constant μ in the system matrix assembly, it can sometimes be useful to introduce a scaling directly into equation (2.7) by multiplying all entries with some factor.

INBEM: FLUID: FIELD:

This block sets the field point of the system. That is, the points in the fluid (not on the boundary) for which the solution should be calculated.

Example:

```
*INBEM: FLUID: FIELD: F3

0.12,   -0.24
0.13,   -0.2
```

means that the pressure should be calculated at the two co-ordinates $(y, z) = (0.12, -0.24)$ and $(y, z) = (0.13, -0.2)$, for the submodel tagged `F3`.

INBEM: FLUID: CHIEF:

This specifies the chief points in the system. These are used for external problems and are some points in the corresponding internal domain that are required to be zero. This is made to avoid problems with the frequencies and wavenumbers corresponding to waves that could propagate if the internal domain was filled with fluid. The chief points should thus not be on nodal lines of the cross section of the internal problem, see also [4]. It should however be noted that so far this 'non uniqueness problem' has not caused any problems. Furthermore the problem may be lessened with a small damping in the fluid. The y- and z-co-ordinates are given in the block.

Example:

```
*INBEM: FLUID: CHIEF: F3

0.01,   0.02
0.02,  0.03
```

defines two chief points, one at $(y, z) = (0.01, 0.02)$ and one at $(y, z) = (0.02, 0.03)$.

INBEM: FLUID: WRITE: WAVEDOMAIN:

Since writing all output data from the model would result in very large files, there are options to chose which data should be written. For instance in many problems the most important information is the radiated power, however more detailed information may be wanted for some frequencies to enhance the physical understanding of the problem.

The current block specifies for which frequencies and wavenumbers more complete data should be written. This is written as the frequency, the real part of the wavenumber and the imaginary part of the wavenumber.

Note!

The frequency and wavenumber must match exactly with those given in the `INSYS: WAVEDOMAIN:` (and complex wavenumbers is not recommended)

Example:

```
*INBEM: FLUID: WRITE: WAVEDOMAIN: F2

200,    1,  0
200,    2,  0
```

means that solution data for the be-subsystem tagged `F2`will be written for the frequency 200 Hz and wavenumbers 1 and 2.

INBEM: FLUID: WRITE: TOP:

This block defines the data on the boundary that should be written. The pressure is also written for all field points, otherwise they would not be defined. The written information gives the element, the local node and whether pressure or normal velocity should be written.

In each row:

- The first entry gives the element number.
- The second entry is the local node number.
- The third entry specifies if the pressure or the velocity for this node is to be written. If this entry is `1` the pressure is written. If this entry is `2` the normal velocity is written.

Example:

```
*INBEM: FLUID: WRITE: TOP: F2

4,  3,  2
5,  2,  1
```

The first row of data means that the normal velocity of the third node in the fourth element is written. The second row means that the pressure of the second node in the fifth element is written.

INBEM: FLUID: END:

The data for each submodel must be ended with the

`INBEM: FLUID: END:`

statement.

Example:

```
*INBEM: FLUID: END: F3
```

ends the data input for the be-fluid-submodel `F3`.

2.4 INCOUP:

`*INCOUP:` states that the following block will define data for the coupling between different submodels. There are currently 10 different coupling submodels implemented in WANDS. These are,

- FE-Plate model and FE-Solid model
- FE-Plate model and FE-Fluid model

- FE-Plate model and BE-Fluid model
- FE-Plate model and BE-Solid model
- FE-Solid model and BE-Fluid model
- FE-Solid model and BE-Solid model
- FE-Solid model and FE-Fluid model
- BE-Fluid model and BE-Fluid model
- BE-Solid model and BE-Solid model
- BE-Solid model and BE-Fluid model

Exactly which submodels that are coupled are defined in the `INSYS:` block.

INCOUP: FE-PLATE: BE-FLUID: END:

Is needed to end the input of data for the plate fluid coupling.

2.4.1 INCOUP: FE-PLATE: FE-SOLID:

The data for computing this coupling is included in one block. However data relating to writing the output is given in two seperate blocks. This coupling is defined in the WAFER manual, there are however some small differences. The coupling is therefore described in the following.

The coupling block couples the solid model together with the plate model. The coupling is specified in the data block,

`*INCOUP: FE-PLATE: FE-SOLID:`

The entries on each row of this block are,

where, the local coordinate ξ is defined as for beam elements connected to solid elements. The scaling constant is used to improve the numerical conditioning of the problem,(e.g. if the other matrices in the total system has entries at magnitudes 10^10 the scaling should also be in that order of magnitude)

Entry number	Entry name
1	Plate element number
2	Local node number of plate element (1 or 2)
3	Solid element number
4	First local node of solid element
5	Second node local of solid element
6	Local coordinate,ξ, between solid node numbers
7	(Optional) scaling constant

Table 2.24: Solid–plate coupling data

Example:

```
*INCOUP: FE-PLATE: FE-SOLID: P1S1


  6, 1, 10,  1,  3, -0.5000
```

Couples the 1:st node of plate element 6 to the solid element 10. The coupling is 1/4 from the local node 1 of the solid element on a line to the local node 3 on this element. This data is for the coupling submodel `P1S1` and the scaling constant is by default 1.0.

INCOUP: FE-PLATE: FE-SOLID: WRITE: WAVEDOMAIN:

This specifies the frequencies and wavenumbers for which output data of the model should be written. These frequency, wavenumber pairs must exactly match some of the pairs in `INSYS: WAVEDOMAIN:`. The output is the lagrange multipliers as defined in the theory manual. These can be interpretated as the forces keeping the two systems together.

Example:

```
*INCOUP: FE-PLATE: FE-SOLID: WRITE: WAVEDOMAIN: P1S1


200,    1.1,  0.0
```

writes the lagrange multipliers for the coupling at frequency 200 and wavenumber 1.1. All the Lagrange multipliers for this coupling subsystem are written.

INCOUP: FE-PLATE: FE-SOLID: END:

The coupling subsytem data input is ended with this statement.

Example:

```
*INCOUP: FE-PLATE: FE-SOLID: END: WAVEDOMAIN: P1S1
```

2.4.2 INCOUP: FE-PLATE: FE-FLUID:

This gives the coupling between a plate finite element model and a fluid finite element model. By using the coupling as defined in [5], the coupling is defined by two matrices that are the complex conjugate transposes of each other. When writing the matrix output only one of these matrices are needed. The coupling is made by between the cubic interpolation of the out of plane displacements of the plate and the edge of the fluid finite element. The latter may use linear, quadratic or cubic interpolation.

The input is defined by the statement

INCOUP: FE-PLATE: FE-FLUID: ELEMENTS:

The data needed is the plate element number, the fluid element number, the number of nodes of the fluid to be coupled, the local node numbers in the fluid element, an internal coordinate (optional) and an (also optional) node reversal flag. Except from defining which nodes are needed in the fluid element, this input follows that used for coupling a plate to a fluid boundary element.

Example:

```
*INCOUP: FE-PLATE: FE-SOLID: ELEMENTS: P1F1
```

```
10, 20, 3, 2, 3, 4, 0.1, 0.2, -1
```

This example means that plate element 10 is connected to fluid element 20. There are three nodes in the fluid element that are to be connected, nodes 2, 3 and 4. A coordinate inside the fluid and close to the edge is (0.1,0.2) and the node numbering of the plate element is switched (reversed) compared to the fluid element.

Note! So far only plate to linear triangular fluid elements have been tested.

INCOUP: FE-PLATE: FE-FLUID: END:

This heading signals the end of the plate to fluid FE coupling.

2.4.3 INCOUP: FE-PLATE: BE-SOLID:

This block defines the coupling between a plate element and a solid boundary element. The coupling is defined by the block,

INCOUP: FE-PLATE: BE-SOLID: ELEMENTS:

The data given in this block is the plate element, the boundary element and optionally an internal coordinate in the solid boundary element domain and, also optional, a node reversal flag. The node reversal flag indicates that the local node ordering is reversed, such that node 1 of the plate element is to be connected to local node 3 of the boundary element.

Example:

```
*INCOUP: FE-PLATE: BE-SOLID: ELEMENTS: P1S1


5, 3,  0.33, 0.44, 1
```

Entry number	Entry name
1	Plate element number
2	Solid boundary element number
3	(Optional) y-coordinate of internal node
4	(Optional) z-coordinate of internal node
5	(Optional) node reversal flag

Table 2.25: Plate to solid boundary coupling data

Means that plate element 5 is connected to boundary element 3 in the submodels defined simultaneously with `P1S1`.

INCOUP: FE-PLATE: BE-SOLID: END:

End the input to this coupling.

2.4.4 INCOUP: FE-PLATE: BE-FLUID:

Blocks beginning with this statement defines the coupling between a plate FE model and a fluid BE model.

INCOUP: FE-PLATE: BE-FLUID: ELEMENTS:

The coupling is defined by the elements in the plate and fluid model that are coupled together. Furthermore the coordinate in the interior of the solid domain (or plate domain) must be given. This defines the coupling so that if the elements have the same node numbering, i.e. the 1:st local node in the BE-element corresponds to the 1:st local node in the FE-element (and the highest local node in the BE-model corresponds to the 2:nd node in the FE-model). If this is not the case a ”flag” must be set to indicate that the node numbering is reversed.

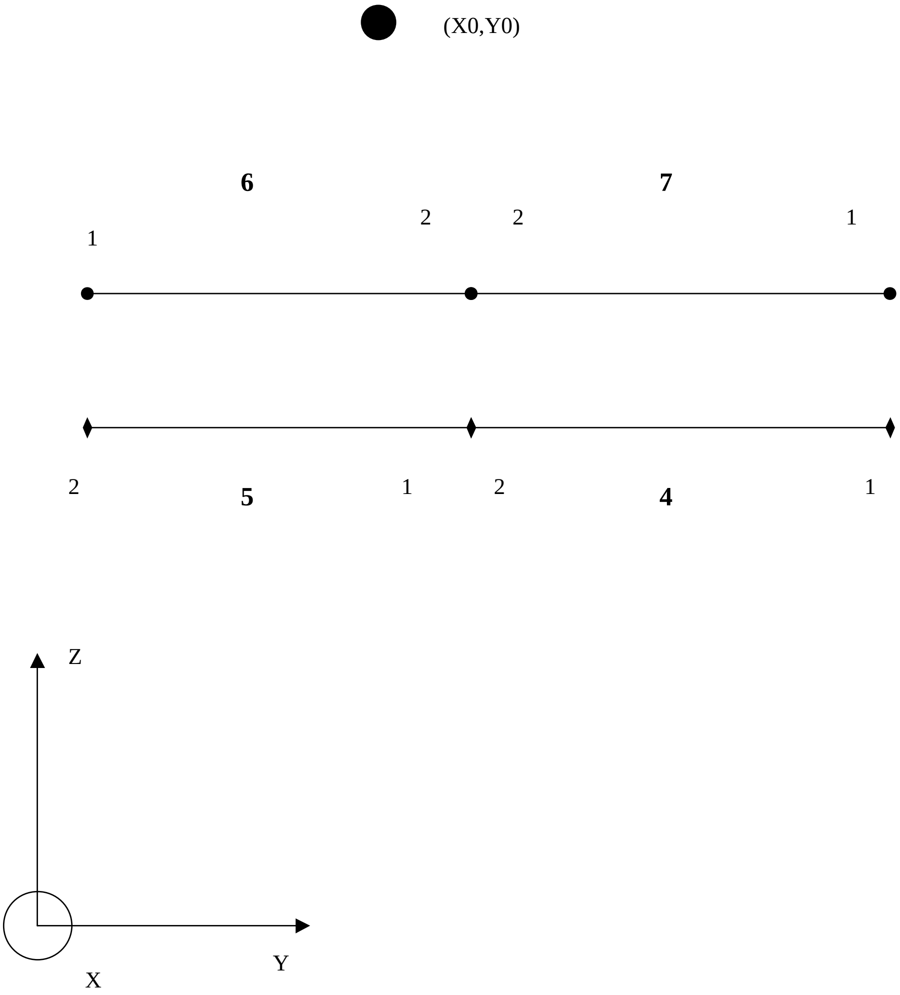

Figure 2.4: Coupling between plate finite elements and fluid boundary elements

Consider figure 2.4. The lines with the dot ends represents a plate model, with the element numbers 6 and 7. The big black dot represents the internal node. The lines with diamond ends represents boundary fluid elements with numbers 4 and 5. The numbers 1 and 2 represents the local node numbers for each element.

To get the right sign of the coupling the sign of the vector product $\mathbf{r_{10}} \times \mathbf{r_{12}}$ is calculated. This sign is however only correct if the first local node of the plate element corresponds to the first local node of the boundary element. If this is not the case it must be indicated with a *node reversal* flag.

The order of the input is given in Table

Input	entry number
plate element	1
boundary element	2
y-coord of internal point	3
z-coord of internal point	4

Table 2.26: Specifying plate fluid coupling

Specifying the internal coordinates is optional and the default coordinate is (0.0,0.0). The node reversal is also optional. If node reversal is required this entry is set to -1.

Example:

With the local nodes according to figure 2.4 and the coordinates $(X_0, Y_0) = (0.78, 1.33)$ the coupling is typically written,

```
*INCOUP: FE-PLATE: BE-FLUID: ELEMENTS: P1F1

7,  4,  0.78,   1.33

6,  5,  0.78,   1.33,   -1
```

Note!

The fluid be-elements can be two, three or four noded.

INCOUP: FE-PLATE: BE-FLUID: END:

The input for the last example is ended with the statement,

```
*INCOUP: FE-PLATE: BE-FLUID: END: P1F1
```

2.4.5 INCOUP: FE-SOLID: FE-FLUID:

The block,

```
*INCOUP: FE-SOLID: FE-FLUID:
```

is used to define coupling between a solid element and a fluid element. The data input is the solid element, the fluid element, number of pairs of nodes to be coupled, the solid nodes, the fluid nodes and optionally also an internal coordinates in the fluid.

Entry number	Entry name
1	Solid element number
2	Fluid element number
3	Number of coupled pairs of nodes (N)
4 to 4+(N-1)	Solid nodes
4+N to 4+(2N-1)	Fluid nodes
4+2N to 4+2N+1	(Optional) Internal coordinates in fluid

Table 2.27: Solid FE–Fluid FE coupling data. N is the number of pairs of coupled nodes

The default internal nodes are (0.0,0.0)

Example

```
*INCOUP: FE-SOLID: FE-FLUID: S1F1

10, 22, 3, 2, 3, 4,  6, 7, 8, 0.25, 0.35
```

Means that the 10:th solid element is coupled to the 22:nd fluid element through the nodes 2,3 and 4 in the solid element and 6, 7 and 8 in the fluid element. A suitable internal coordinate in the fluid is (0.25, 0.35)

INCOUP: FE-SOLID: FE-FLUID: END:

This is used to end the input to the FE-solid to FE-fluid coupling block.

2.4.6 INCOUP: FE-SOLID: BE-SOLID:

This block is used to couple solid finite elements with solid boundary elements. The data block is specified by giving the element number for the finite element, the element number for the boundary element, the solid element node numbers and optionally an internal coordinate. The internal coordinate is within the boundary element domain. By default the internal coordinate is (0.0,0.0). If a different internal coordinate is to be used the line has to end with `-1`.

Entry number	Entry name
1	Finite element number
2	Boundary element number
3 to 5	Local nodes in the finite element
6	(Optional) y-coordinate of internal node
7	(Optional) z-coordinate of internal node
8	-1 (only needed if internal coordinate is given)

Table 2.28: Solid FE–Solid BE coupling data.

Note! There are only quadratic boundary elements included. The only coupling to currently allowed is with quadratic finite elements. However, the input is data is prepared for using other options.

INCOUP: FE-SOLID: BE-SOLID: END:

Signals the end of the FE-solid to BE-solid coupling

2.4.7 INCOUP: FE-SOLID: BE-FLUID:

INCOUP: FE-SOLID: BE-FLUID: ELEMENTS:

The coupling between a solid finite element and a fluid boundary element is made similarly to that between plate element and fluid elements. The difference is that the local node numbers of the solid must also be given and Node reversal is still included as an option.

Example:

Consider the Figure 2.5. Here an eight noded solid element is coupled to a three noded fluid boundary element. The solid element has element number 2 and the fluid element has element number 9.

If the coupling submodel is tagged `S1F2` the coupling between these two elements is written as,

```
*INCOUP: FE-SOLID: BE-FLUID: ELEMENTS: S1F2

2,  9,  1,  8,  7,  2.3,    4.6
```

If node reversal was to be used a `, -1` would be added to the line above.

INCOUP: FE-SOLID: BE-FLUID: END:

The input for the last example is ended with the statement,

```
*INCOUP: FE-SOLID: BE-FLUID: END: S1F2
```

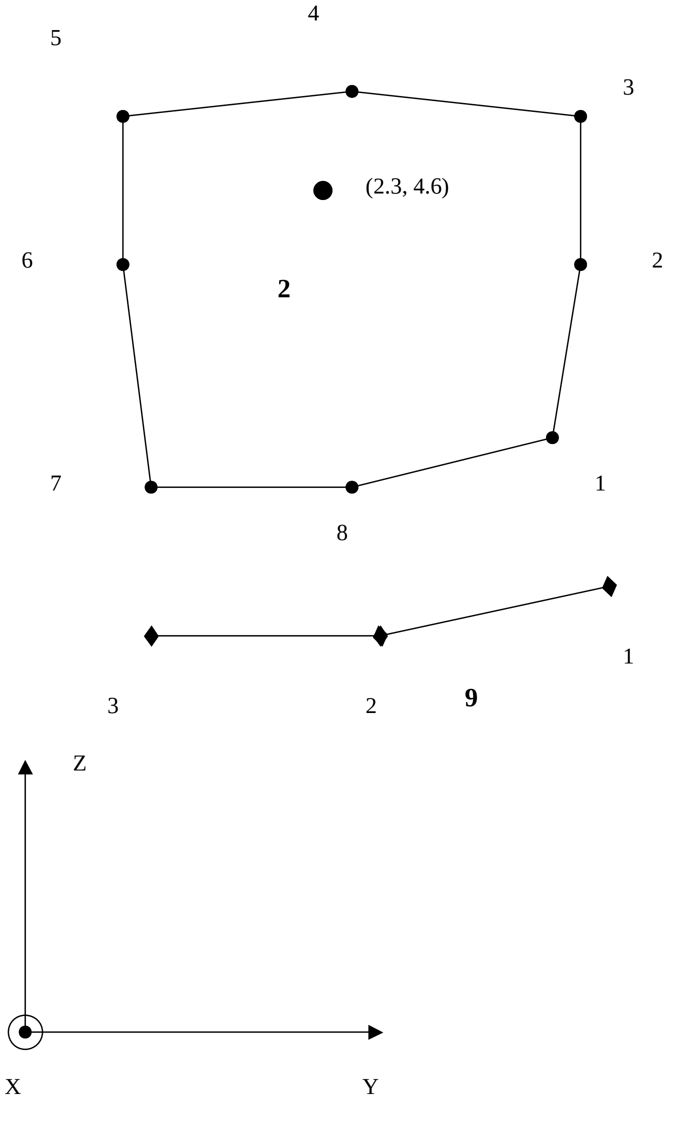

Figure 2.5: Coupling between a solid finite element and fluid boundary element

2.4.8 INCOUP: BE-FLUID: BE-FLUID:

The coupling between two fluids can also be useful sometimes, for instance when considering underwater sound propagation, where different layers of water have different properties.

INCOUP: BE-FLUID: BE-FLUID: MAT:

The coupling is usually made so that the pressures and normal velocities at the boundary are the same in both models. However other boundary conditions may sometimes be useful. In such cases impedance matrices at the boundary can be defined by the user. The impedance matrices are defined by,

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} p_a \\ v_a \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} p_b \\ v_b \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{2.9}$$

where p_a and v_a is the pressure and normal velocity at the boundary respectively for the first BE-submodel. Similarly p_b and v_b is the pressure and normal velocity at the boundary respectively for the second BE-submodel. The matrices are also assumed to be symmetric, so that $a_{12} = a_{21}$ and $b_{12} = b21$.

There are two predefined boundary conditions in WANDS. The first states that $p_a = p_b$ and $v_a = -v_b$, which is the normal way to couple the systems. This is the default boundary condition set and it is numbered `0`. The second predefined boundary condition states that $p_a = p_b$ and $v_a = v_b$, which sometimes can be useful if the normal velocity is defined in the same directions for both submodels. This set of boundary conditions is numbered `-1`.

When writing this data, the line number after the heading gives the impedance number. Each line of data is either 8 numbers long, if all entries in the matrices are real, or 16 numbers long, if the matrices are complex. The order of input of matrix entries are as follows. If there are only real valued entries, the order of the entries are:

$a_{11}, a_{12}, a_{22}, b_{11}, b_{12}, b_{22}, c_1, c_2$

If some numbers are complex the order of the entries are:

$\Re(a_{11}), \Im(a_{11}), \Re(a_{12}), \Im(a_{12}), \Re(a_{22}), \Im(a_{22}), \Re(b_{11}), \Im(b_{11}), \Re(b_{12}), \Im(b_{12}), \Re(b_{22}), \Im(b_{22}), \Re(c_1), \Im(c_1), \Re(c_2), \Im(c_2)$

(thus 16 numbers must be written on one line)

Example:

```
*INCOUP: BE-FLUID: BE-FLUID: MAT: F1F2


10.0, 0.0, 100.0, -10.0, 0.0, 100.0, 0.0, 0.0
```

is a different way to express the default boundary condition set.

INCOUP: BE-FLUID: BE-FLUID: TOP:

The topology of this coupling requires that the be-elements of the respective model have the same number of nodes. The topology is given for each node. The entries are,

Entry number	Entry name
1	1:st BE submodel element number
2	Local node number of 1:st BE-submodel element
3	1:st BE submodel element number
4	Local node number of 1:st BE-submodel element
5	BC model number

Table 2.29: Solid–plate coupling data

If the 5:th entry is missing it is by default set to `0`, (see the last section)

Example:

```
*INCOUP: BE-FLUID: BE-FLUID: TOP: F1F2


5,1,8,2,1
```

```
5,2,8,1
```

The first row means that the 1:st local node of the 5:th element of the first submodel is connected to the 2:nd local node of the 8:th element in the second submodel. Furthermore the boundary condition used is the first one specified by the,

```
*INCOUP: BE-FLUID: BE-FLUID: MAT: F1F2
```

statement. The second row means that the 2:nd node of the 5:th element is coupled to the 1:st node of the 8:th element by the default boundary condition.

INCOUP: BE-FLUID: BE-FLUID: END:

This block ends the coupling-subsystem between the fluid BE-models.

2.4.9 INCOUP: BE-SOLID: BE-SOLID:

This couples two solid boundary elements. This is implemented without the option of specifying an impedance at the boundary, that can be used for coupling fluid boundary element domains. The coupling is made node by node, rather that element by element.

The coupling is specified by the block,

INCOUP: BE-SOLID: BE-SOLID: TOP:

The data in this block is given in the order of the node in the first submodel, the direction in this node which is to be coupled, the node in the second submodel, the direction which is to be coupled in this node. An optional flag can be given as a fifth input argument. If this flag is set to `-1`, it means that the coupling is such that the direction in the first submodel is negative to the direction in the second submodel.

Example:

Consider Figure 2.6

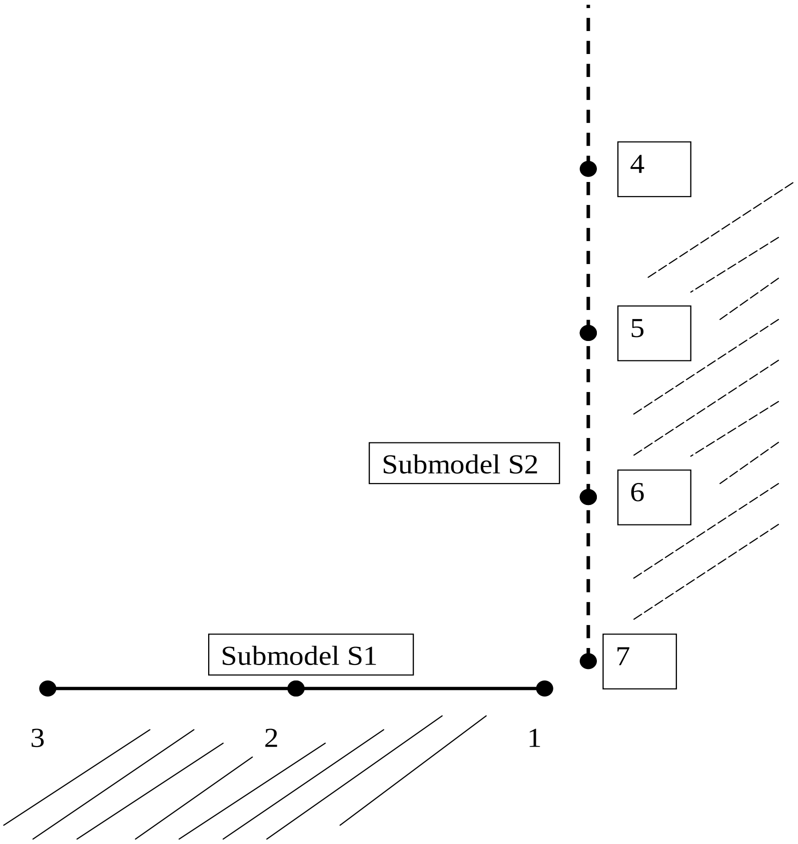

Figure 2.6: Coupling between two solid BE-domains

Assume that the node numbering in the topology inputs to submodel S1 and submodel S2 follows that indicated in the figure, i.e. the internal domains are those indicated by the shaded areas. Also consider the coordinates in Figure 2.3.

If we want to couple Node 1 of submodel S1 with Node 7 of submodel S2, we first note that the x-direction is always out of the plane, thus,

$$u_1^{S1} = u_7^{S2} \tag{2.10}$$

The tangential displacement of Node 1 of submodel S1 is to the left. This corresponds to a normal displacement into the solid domain S2. Thus,

$$v_1^{S1} = -w_7^{S2} \tag{2.11}$$

Finally the normal displacement in S1 is pointing downwards as is the tangential displacement of model S2. Thus,

$$w_1^{S1} = v_7^{S2} \tag{2.12}$$

If the coupling model is denoted `S1S2`, the for these three relations becomes,

```
*INCOUP: BE-SOLID: BE-SOLID: TOP: S1S2

1, 1, 7, 1
1, 2, 7, 3, -1
1, 3, 7, 2
```

Note! This coupling has not been fully validated.

2.4.10 INCOUP: BE-SOLID: BE-FLUID:

This coupling creates the possibility of coupling a solid domain to a fluid domain. The coupling is defined through an impedance matrix like that used for coupling two fluid domains, i.e. through

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} t_3^s \\ v^s \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} p^f \\ v^f \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \tag{2.13}$$

where t_3^a is the traction vector working in compression on the solid boundary, v^s is the corresponding velocity into the solid domain, p^f is the pressure at the fluid boundary and v^f is the velocity into the fluid domain.

For most cases it is sufficient to use,

$$t_3^s = p^f \tag{2.14}$$

and

$$v^s = -v^f \tag{2.15}$$

There is also one option built in where

$$t_3^s = p^f \tag{2.16}$$

and

$$v^s = v^f \tag{2.17}$$

These two options are predefined in WAMDS as impedance 'materials' 0 and -1 respectively. If a different impedance material is requested it can be defined through the

INCOUP: BE-SOLID: BE-FLUID: MAT:

statement. The input is totally analogous with those for the impedance between two fluid boundary elements, given by the `INCOUP: BE-FLUID: BE-FLUID: MAT:` statement.

INCOUP: BE-SOLID: BE-FLUID: TOP:

This specifies which nodes in the solid submodel are coupled to which nodes in the fluid submodel. The input data is

Entry number	Entry name
1	Node in solid BE submodel
2	Direction to be coupled in solid BE submodel
3	Node in fluid BE submodel
4	Impedance material to be used (Optional)

Table 2.30: Input data for coupling between solid and fluid boundary element submodels

The impedance material used is `0` by default, i.e. a hard wall coupling. The displacement direction in the solid BE submodel is usually `3`, but there is a possibility to use the other directions as well, e.g. when the two boundaries are positioned at an angle to each other.

Note! The solid to fluid boundary element coupling only gives a condition for the out of plane displacements and forces of the solid boundary elements. To create a fully determined system more constraints has to be defined, e.g. by adding boundary conditions.

Chapter 3

Output files

3.1 Types of output files

There are several different types of output files written by WANDS. The output files are grouped as follows,

- Topology file.
- Matrix files.
- Solution files.

The topology file gives information about the node numbering and degrees of freedom in the finite element subsystems. Output for boundary element subsystems have not yet been implemented. These outputs are however less important since they are easily found from the input file.

The matrix files gives the matrices for the different submodels. When considering a ”double loop” solution, the boundary element matrices are only written for the first frequency and wavenumber, since this information otherwise would give very many files.

There are two solution files written for the ”double loop” solution. The first is the solutions specified at different frequencies, wavenumbers and degrees of freedom in the model. The second is the radiated power and radiation efficiency for the different boundary element submodels.

3.2 Topology file

The topology file is written in order see how global and local degrees of freedom are related. This is especially important for solid finite elements, where the global node numbering is difficult to find directly from the input file. On the other hand, the fluid boundary topology is directly given from the input. Hence, fluid boundary element topology is not given in the topology file.

The topology file is therefore called `FE_topology.out`.

The output data in the topology output file is, just like the input file, ordered in a number of blocks. Each block begins with the line:

```
*OUTFEM:
```

possibly followed by an information line beginning with, `**`, e.g.

```
*OUTFEM: SOLID: NODECOORD:
**nodenumber,y-coordinate, z-coordinate
```

is starting a block that gives data for the nodal coordinates of the solid sub–model. The actual data follows below these two first lines. The line starting with `**` is only there for information and is usually not referred to in the following of this chapter.

In the following of this Chapter, the different blocks are listed in the order they appear. Note however that blocks that does not contain any data are not written. Also at the end of the Chapter there is a Figure summarizing the different blocks.

3.2.1 OUTFEM: SOLID:

If there is a solid sub–model present, the data for this model is written first in the `'FE_topology.out'` file.

SOLID: SIZE:

This, first block, has one line of data only. This data is,

Entry number	Entry name
1	Number of degrees of freedom in the model
2	Number of elements
3	Number of nodes
4	Number of nodes on the z-axis
5	Number of restrained nodes

Note! The symmetry option is not yet included in the model, thus the number of nodes on the z-axis is never counted on the program and thus (in WAFER 1.0) always zero.

Example:

```
*OUTFEM: SOLID: SIZE: S2
**number of: dofs, elements, nodes, nodes on z-axis, restrained nodes
159,   10,   53,    0,    0
```

Tells that this solid model has 159 degrees of freedom, 10 solid elements, 53, nodes and no restrained nodes in the `S2` tagged submodel.

SOLID: NODECOORD:

Each data line in this block corresponds to one node in the model. The following data is written,

Example:

Entry number	Entry name
1	Node number
2	y–coordinate of the node
3	z–coordinate of the node

```
*OUTFEM: SOLID: NODECOORD: S1
** node number, y-coordinate, z-coordinate
    1,  0.300, -0.010
    2,  0.300,  0.000
    3,  0.300,  0.010
    4,  0.275,  0.010
    5,  0.250,  0.010
    6,  0.250,  0.000
    7,  0.250, -0.010
    8,  0.275, -0.010
    9,  0.350, -0.010
   10,  0.350,  0.000
   11,  0.350,  0.010
   12,  0.325,  0.010
   13,  0.325, -0.010
```

is 13 node coordinates of the submodel tagged `S1`

Note! The node numbering in the solid model output is generally not the same as the node numbering given in the input. This is because the program renumbers the nodes to minimize the bandwidth of the output matrices.

SOLID: ELETOP:

This data block gives the node numbers used to form each element. The first entry of a line of data gives the element number. The following entries gives the nodes. The node numbers are given in the order used to build the element, as discussed in Section 2.2.1.

Example:

```
*OUTFEM: SOLID: ELETOP: S1
**elements and their node numbers
   1,   1,   2,   3,   4,   5,   6,   7,   8
   2,   9,  10,  11,  12,   3,   2,   1,  13
```

describe two eight–noded elements , '1' and '2'. The first uses nodenumbers '1' to '8' and the second uses node numbers '9, 10, 11, 12, 3, 2, 1' and '13'. Combining this information with that for the node coordinates given in the example in Section 3.2.1, the mesh for the two elements is given in Figure 3.1.

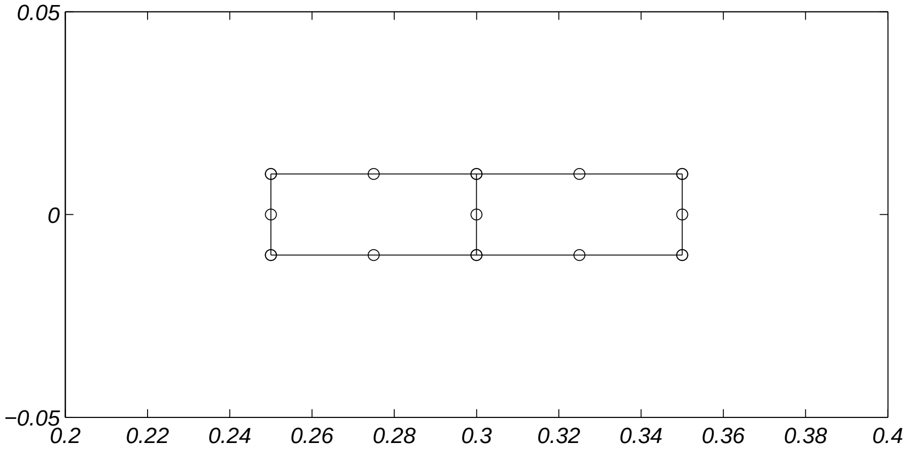

Figure 3.1: Solid two element mesh, 'o' denotes nodes

Note! The fact that there are 8 nodes in each element, tells us that the elements are of type 5, as seen in Figure 2.2.

SOLID: NODEDOF:

This data block relates the nodes with the degrees of freedom in the solid element sub–model. For each line the data written is given in Table 3.1

Example:

```
 *OUTFEM: SOLID: NODEDOF: S3
```

Entry number	Entry name
1	Node number
2	degree of freedom for the x–displacement of the node
3	degree of freedom for the y–displacement of the node
3	degree of freedom for the z–displacement of the node

Table 3.1: Output of solid model degrees of freedom

```
**nodes and their dof numbers
   1,    1,    2,    3
   2,    4,    5,    6
   3,    7,    8,    9
   4,   10,   11,   12
```

,gives the degrees of freedom associated with the first four nodes of the model. The degree of freedom, `'12'`, corresponds to the z–displacement of node 4. This also corresponds to the 12:th entry of the vector, $\tilde{\mathbf{W}}$, in equation 3.2(given that this equation represents the solid model only).

Note! If a degree of freedom is zero, (`'0'`), it is restrained.

3.2.2 OUTFEM: PLATE:

If there is a plate sub–model present, the data for this model is written second in the `'FE_topology.out'` file. The block data basically follow those for a solid sub–model.

PLATE: SIZE:

This, first block, has one line of data only. This data is,

Note! The symmetry option is not yet included in the model, thus the number of nodes on the z-axis is never counted in the program and thus (in version 1.0 of

Entry number	Entry name
1	Number of degrees of freedom in the model
2	Number of elements
3	Number of nodes
4	Number of nodes on the z-axis
5	Number of restrained nodes

WAFER) always zero.

Example:

```
*OUTFEM: PLATE: SIZE: P1
**number of: dofs, elements, nodes, nodes on z-axis, restrained nodes
 42,   10,   12,    0,    2
```

Tells that this plate submodel (P1) has 42 degrees of freedom, 10 solid elements, 12, nodes of which 2 are restrained.

PLATE: NODECOORD:

Each data line in this block corresponds to one node in the model. The following data is written,

Entry number	Entry name
1	Node number
2	y–coordinate of the node
3	z–coordinate of the node

Example:

```
*OUTFEM: PLATE: NODECOORD: P2
** node number, y-coordinate, z-coordinate
   1,  0.000,  0.000
```

```
    2,  0.050,  0.100
    3,  0.100,  0.100
    4,  0.150,  0.000
```

...

PLATE: ELETOP:

This data block gives the node numbers used to form each element. The first entry of a line of data is the element number. The following two are the local nodes 1 and 2 of the element.

Example:

```
*OUTFEM: PLATE: ELETOP: P1
**elements and their node numbers
  1,   1,   2
  2,   2,   3
  3,   3,   4
```

describes three plate elements, ’1’, ’2’ and ’3’.

The first element uses node numbers ’1’ and ’2’. The second element uses node numbers ’2’ and ’3’. The third element uses node numbers ’3’ and ’4’. By using the node coordinate data from the example in Section 3.2.2, the three element mesh in Figure 3.2

PLATE: NODEDOF:

This data block relates the nodes with the degrees of freedom in the plate element sub-model. For each line the data written is,

Example:

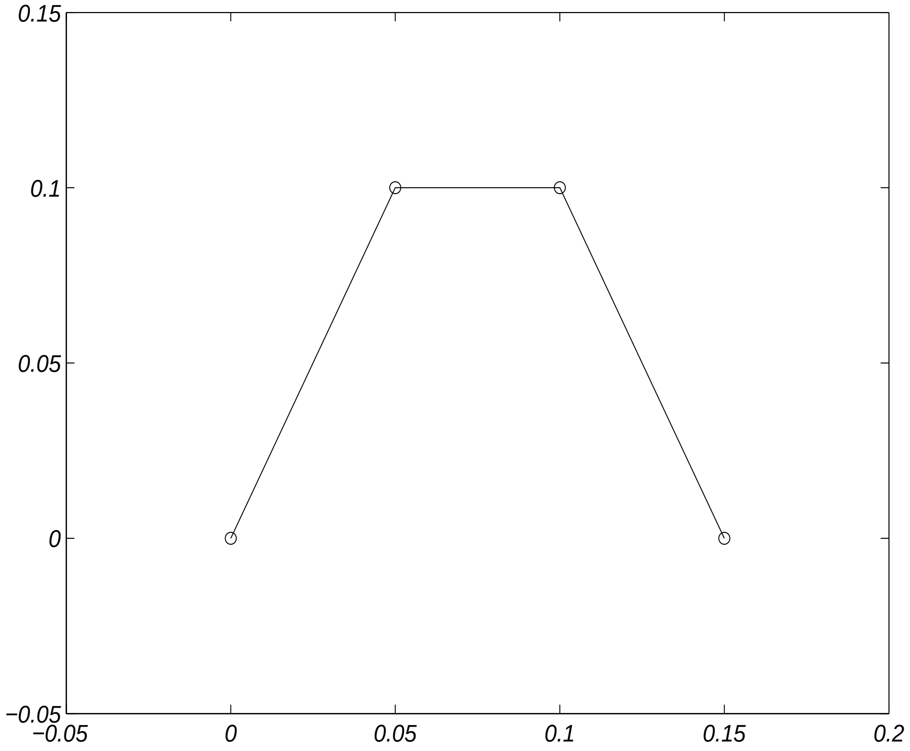

Figure 3.2: Plate three element mesh, 'o' denotes nodes

Entry number	Entry name
1	Node number
2	degree of freedom for the x–displacement of the node
3	degree of freedom for the y–displacement of the node
4	degree of freedom for the z–displacement of the node
5	degree of freedom for the rotation about the x–axis of the node

```
*OUTFEM: PLATE: NODEDOF:
**nodes and their dof numbers
   1,    0 ,    0,    0,    1
   2,    2 ,    3,    4,    5
```

,gives the degrees of freedom associated with the first two nodes of the model described in Appendix A.

Note! If a degree of freedom is zero, (`0`), it is restrained.

3.2.3 OUTFEM: BEAM:

The data for the beam models are written in blocks starting with,

`*OUTFEM: BEAM:`

Data for the beam sub-model connected to the solid sub-model starts with,

`*OUTFEM: BEAM: SOLID: TAG`

and data for the beam sub-model connected to the plate sub-model starts with,

`*OUTFEM: BEAM: PLATE: TAG`

BEAM: PLATE: NODEDOF:

Consider a beam connected to a plate element node. This data block gives the node number and the corresponding degrees of freedom in the plate element sub-model. Thus,

Example:

```
*OUTFEM: BEAM: PLATE: NODEDOFS: P1
**Nodenumber, dofs (4)
2,   5,   6,   7,   8
```

,tells us that the first beam element (= first line of data in the block) is connected to plate node number `2` and that the degrees of freedom associated with this node is, `5,  6, 7, 8`, corresponding to the x–, y– , z– displacements and the rotation about the x–axis.

Note! Except from the node number this information is redundant, since it also can be found in the `'*OUTFEM: PLATE: NODEDOF:'` block.

3.2.4 BEAM: PLATE: NODECOORD:

This data block gives the node coordinates of the plate–nodes onto which beam elements are coupled, e.g.

```
*OUTFEM: BEAM: PLATE: NODECOORD: P2
**Nodenumber, y-coord, z-coord
 2,  0.050,  0.000
```

Tells us that node '2' has the coordinates $y = 0.050$ and $z = 0.000$.

Note! This data is redundant, since it can be found in the '*OUTFEM: PLATE: NODECOORD:' data block.

BEAM: SOLID: NODEDOF:

Consider a beam connected to a solid. This data block gives the node numbers and the corresponding degrees of freedom in the solid element sub-model.

The output data on each row is as follows,

Entry number	Entry name
1	First node number of coupling
2	Second node number of coupling
3	Degree of freedom for the x–displacement of the first node
4	Degree of freedom for the y–displacement of the first node
5	Degree of freedom for the z–displacement of the first node
6	Degree of freedom for the x–displacement of the second node
7	Degree of freedom for the y–displacement of the second node
8	Degree of freedom for the z–displacement of the second node

Example:

```
*OUTFEM: BEAM: SOLID: NODEDOFS: S2
**Nodenumbers (2) and dofs (6)
 24,   26,   70,   71,   72,   76,   77,   78
```

The example indicates a node coupled to a point somewhere on a line between nodes 24 and 26 of the solid model. The degrees of freedom for these nodes are 70, 71, 72 and 76, 77, 78 respectively.

Note! The node numbers given are those of the solid, the beam does not have a node number of it's own.

BEAM: SOLID: NODECOORD:

This data block gives the node coordinate of the beam. The data on each line is,

Entry number	Entry name
1	First node number of coupling
2	Second node number of coupling
3	y–coordinate of beam
4	z–coordinate of beam

Example:

```
*OUTFEM: BEAM: SOLID: NODECOORD:
**Nodenumbers (2) + y- and z-coord for coupling
  24,   26, 0.500, 0.000
```

gives the information that the first beam (= first line of data) is connected to nodes 24 and 26 of the solid model. The global coordinate of the beam is $y = 0.500$ and $z = 0.000$.

3.2.5 OUTFEM: COUPLING:

These blocks gives information about the coupling between the solid and the plate sub–models. As for the solid, plate and beam sub–models the data is only written if a coupling is actually defined.

COUPLING: SIZE:

This block gives the size of the coupling.

Example:

Entry number	Entry name
1	Number of rows in the coupling matrix
2	Number of columns in the coupling matrix
3	Number of coupling points

```
*OUTFEM: COUPLING: SIZE: P1S1
**rows, columns, couplings
    8,  201,    2
```

There are 4 equations for each coupling point, (x–, y–, z–displacements and rotation about the x–axis). Hence, 2 couplings in the example implies 8 coupling equations and thus, 8 rows in the coupling matrix.

COUPLING: PLATE: NODEDOF:

This blocks gives the node numbers and degrees of freedom for the coupling to the plate sub–model. For each line of data in this block,

Entry number	Entry name
1	Node number in the plate–submodel
2	x-displacement degree of freedom of the node
3	y-displacement degree of freedom of the node
4	z-displacement degree of freedom of the node
5	rotation degree of freedom of the node

Example:

```
*OUTFEM: COUPLING: PLATE: NODEDOF: P1S1
**Nodenumber, dofs (4)
    6,   18,   19,   20,   21
    7,   22,   23,   24,   25
```

The above example gives the information that, node `6` of the plate sub–model is involved in the first coupling (= first data line) between the solid and plate sub–models. The degrees of freedom numbers are `18, 19, 20, 21` corresponding to the x–, y–, z– displacements and rotation about the x–coordinate, respectively. Similarly node `7` is involved in the second coupling.

COUPLING: PLATE: NODECOORD:

This block gives the coordinates of the plate nodes involved in the coupling. Thus for each line of data we have,

Entry number	Entry name
1	Node number in the plate sub–model
2	x–coordinate of the node
3	y–coordinate of the node

Note! This data block is basically redundant since the same information is given in the `OUTFEM: PLATE: NODECOORD:` block.

COUPLING: SOLID: NODEDOF:

This block gives data for the nodes and degrees of freedom involved in the coupling to the solid sub–model. Each line of data specifies one coupling.

Entry number	Entry name
1	First node number in the solid sub–model coupling
2	Second node number in the solid sub–model coupling
3	x–displacement degree of freedom for the first node
4	y–displacement degree of freedom for the first node
5	z–displacement degree of freedom for the first node
6	x–displacement degree of freedom for the second node
7	y–displacement degree of freedom for the second node
8	z–displacement degree of freedom for the second node

Example:

```
*OUTFEM: COUPLING: SOLID: NODEDOF:
**Nodenumbers (2) and dofs (6)
    5,    7,   13,   14,   15,   19,   20,   21
```

COUPLING: SOLID: NODECOORD:

Each line of data in his block gives information about the coordinates for the coupling on the node. The first data line corresponds to the first coupling etc. The data is given in the following order,

Entry number	Entry name
1	First node number in the solid sub–model coupling
2	Second node number in the solid sub–model coupling
3	x–coordinate for the coupling point
4	y–coordinate for the coupling point

This is exactly the same order as the `OUTFEM: BEAM: SOLID: NODECOORD:` block.

Note! Together with the data in the `'OUTFEM: COUPLING: SOLID: NODECOORDS:'` block, this block gives a mean to check the difference between the plate and solid coupling coordinates. In theory there should be no difference at all, but for practicality coupling between different coordinates is allowed in WAFER.

3.2.6 OUTFEM: END:

The line,

```
*OUTFEM: END: TAG
```

signals the end of the output for the `TAG` submodel.

3.2.7 Summary output data blocks

Figure 3.3 summarize all the `OUTFEM:` data blocks in this chapter.

3.3 Matrix files

Files containing the matrix entries are written for each subsystem for which this is specified in the `INSYS:` block. Each matrix is named according to which type of matrix it is followed by the subsystem tag and the extension `.out`.

Example:

```
m_ssS1.out
```

is the mass matrix (`m`) for a solid (the `ss`) for a subsystem tagged `S1`.

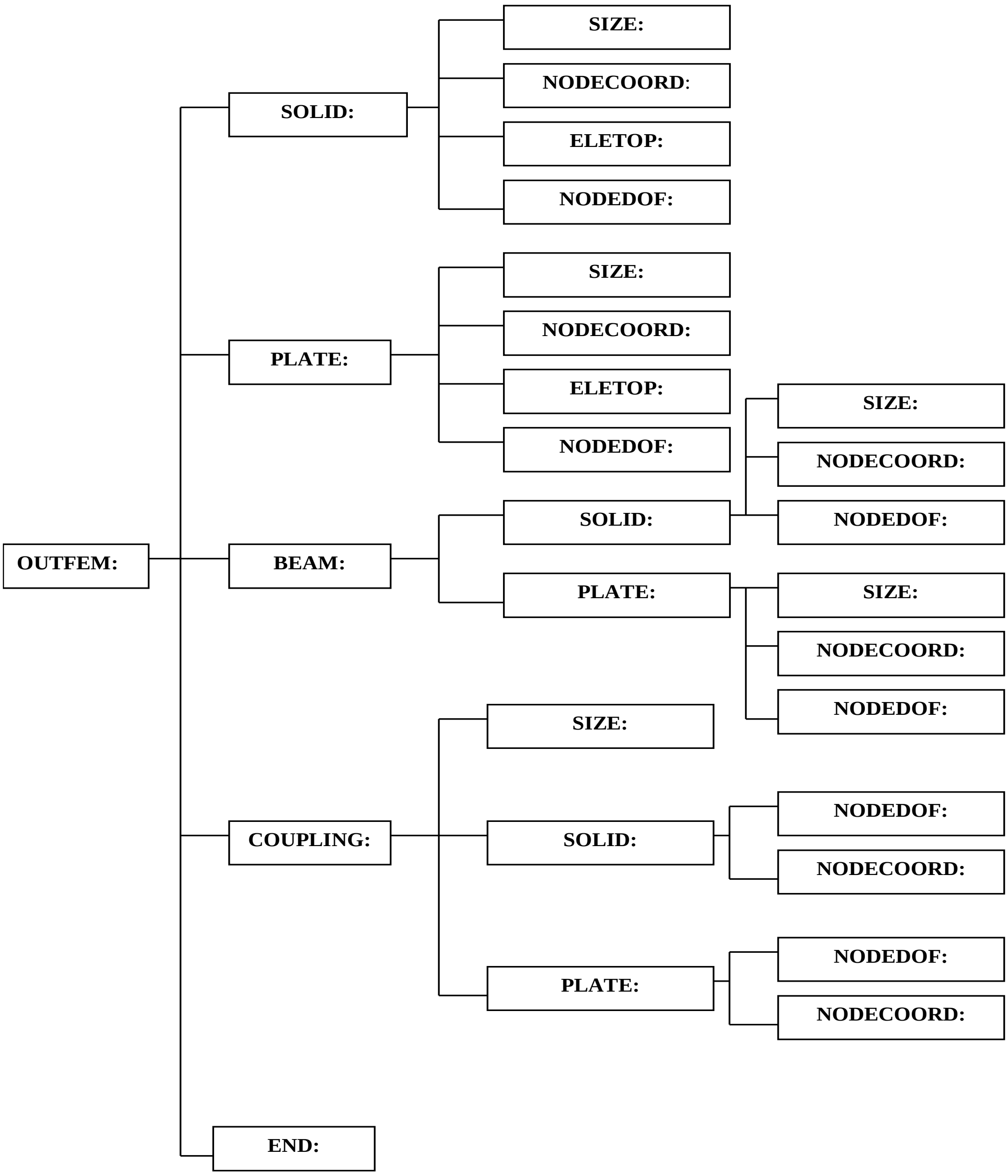

Figure 3.3: Summary of OUTFEM: data blocks

The matrix files are written with in a sparse matrix format,

3.3.1 Matrix files for plate subsystems

The waveguide-FE formulation for a plate subsystem takes either the weak form:

$$\delta L_\omega = \int_x \sum_{i=0}^{2} \sum_{j=0}^{2} \frac{\partial^i \delta \tilde{\mathbf{U}}}{\partial x^i} \mathbf{A}_{ij} \frac{\partial^j \tilde{\mathbf{U}}}{\partial x^j} - \omega^2 \delta \tilde{\mathbf{U}}^{\mathrm{H}} \left[\mathbf{M}\right] \tilde{\mathbf{u}} - \delta \tilde{\mathbf{U}}^{\mathrm{H}} \tilde{\mathbf{F}} \mathrm{d}x = \mathbf{0} \tag{3.1}$$

or the strong form,

$$\left[\mathbf{K_4} \frac{\partial^4}{\partial x^4} + \mathbf{K_2} \frac{\partial^2}{\partial x^2} + \mathbf{K_1} \frac{\partial^2}{\partial x^2} + \mathbf{K_0} - \omega^2 \mathbf{M}\right] \tilde{\mathbf{U}} - \tilde{\mathbf{F}} = \mathbf{0} \tag{3.2}$$

How these equations are derived is discussed in the theory manual. What is important for the discussion here is that there are some relations between the different matrices. This means that not all matrices has to be written to output files. The relations are,

$$\mathbf{A}_{20} = \mathbf{A}_{02}^T \tag{3.3}$$

$$\mathbf{A}_{10} = \mathbf{A}_{01}^T \tag{3.4}$$

$$\mathbf{K}_4 = \mathbf{A}_{22} \tag{3.5}$$

$$\mathbf{K}_2 = \mathbf{A}_{02} - \mathbf{A}_{11} + \mathbf{A}_{20} \tag{3.6}$$

$$\mathbf{K}_1 = \mathbf{A}_{01} - \mathbf{A}_{10} \tag{3.7}$$

Furthermore

$$\mathbf{K}_3 = \mathbf{A}_{12} = \mathbf{A}_{21} = \mathbf{0} \tag{3.8}$$

The matrices written as outputs are are,

Matrix	Real part	Imaginary part
$\mathbf{M}$	`m_psTAG.out`	-
$\mathbf{K}_4$	`k4_psTAG.out`	`c4_psTAG.out`
$\mathbf{K}_0$	`k0_psTAG.out`	`c0_psTAG.out`
$\mathbf{A}_{01}$	`a01_psTAG.out`	`b01_psTAG.out`
$\mathbf{A}_{11}$	`a11_psTAG.out`	`b11_psTAG.out`
$\mathbf{A}_{02}$	`a02_psTAG.out`	`b02_psTAG.out`

where "`TAG`" is the tag of the submodel, e.g. "`P1`".

Beams connected to this system are given their own matrices that may be added to equation (3.1) or (3.2). For a strict beam model the only non zero matrices will be the $\mathbf{M}$ the $\mathbf{A}_{22}$ and $\mathbf{A}_{11}$. The output matrices are

Matrix	Real part	Imaginary part
$\mathbf{M}$	`m_bpTAG.out`	-
$\mathbf{A}_{11} = \mathbf{K}_2$	`k2_bpTAG.out`	`c2_bpTAG.out`
$\mathbf{A}_{22} = \mathbf{K}_4$	`k4_bpTAG.out`	`c4_bpTAG.out`

These matrices have the same size as the plate matrices in the same submodel. Hence the combined plate–beam system is obtained by adding the respective matrices together.

3.3.2 Matrix files for solid subsystems

These matrices are very much like those for a plate subsystem. The mathematical differences are that the $\mathbf{K}_4$ ($= \mathbf{A}_{22}$) and $\mathbf{A}_{02}$ matrices does not exist. Also, the "`_ps`" part of the matrix files is substituted with "`_ss`". Hence,

Matrix	Real part	Imaginary part
$\mathbf{M}$	`m_psTAG.out`	-
$\mathbf{K}_0$	`k0_ssTAG.out`	`c0_ssTAG.out`
$\mathbf{A}_{01}$	`a01_ssTAG.out`	`b01_ssTAG.out`
$\mathbf{A}_{11}$	`a11_ssTAG.out`	`b11_ssTAG.out`

Beams connected to this system are given their own matrices that may be added to equation (3.1) or (3.2). For a strict beam model the only non zero matrices will be the $\mathbf{M}$ the $\mathbf{A}_{22}$ and $\mathbf{A}_{11}$. The output matrices are

Matrix	Real part	Imaginary part
$\mathbf{M}$	`m_bsTAG.out`	-
$\mathbf{A}_{11} = \mathbf{K}_2$	`k2_bsTAG.out`	`c2_bsTAG.out`
$\mathbf{A}_{22} = \mathbf{K}_4$	`k4_bsTAG.out`	`c4_bsTAG.out`

Note that the solid model does not have a $\mathbf{K}_4$ matrix, but the (euler) beam has.

3.3.3 Matrix files for plate-solid coupling

The plate to solid coupling is described by the $\mathbf{C}_{sp1}$ and $\mathbf{C}_{sp2}$ matrices. Different ways to implement this coupling is described in the theory manual, [1]. The matrix file name is written

Matrix	Matrix
$\mathbf{C}_{sp1}$	`CP_TAG.out`
$\mathbf{C}_{sp2}$	`CS_TAG.out`

The names are chosen to reflect the descriptions of the plate ,`P`, and solid, `C`, couplings respectively.

3.3.4 Other matrix files

Currently other subsystem matrix files are not explicitly written from the WANDS program when the double loop solution is employed. However, the total system matrix is written for the first frequency-wavenumber pair. From this large matrix all BE-matrices and coupling matrices can be extracted. The file is named `sysmat11_TAG.out`, the vector on the right hand side is found in `sysvec11_TAG.out`. Here `TAG` refers to the tag of the system and `11` refers to the first frequency and the first wavenumber.

3.4 Solution output

There are two solution output files in WANDS. The first is called `solutionTAG.out` and the second is called `powerTAG.out`. `TAG` is here the systems tag as given in the `* INSYS: TITLE:` block.

The "solutionTAG.out" file

This file starts with the header,

```
* OUTSYS: SOLUTION: DLOOP:
```

```
** outputs for double loop solution **
```

Then for each frequency-wavenumber pair, the header,

```
 **************************
```

```
* OUTSYS: WAVEDOMAIN:
```

is written followed by the frequency and wavenumber.

The solutions for the respective submodel are then written.

The block,

```
* OUTFEM: PLATE: SOLUTION: TAG
```

writes the solution for the different plate submodels. The solution is written as a number of rows with the degree of freedom in the subsystem followed by the real and imaginary parts of the solutions for this degree of freedom.

Likewise the block,

```
* OUTFEM: SOLID: SOLUTION: TAG
```

writes the solution for a solid submodel in the same way.

The block with the header

```
* OUTCOUP: FE-PLATE: FE-SOLID: SOLUTION: TAG
```

writes the coupling degrees of freedom for a coupling between a plate and a solid submodel, i.e. the forces needed to keep the two models together.

The block with the header

```
* OUTBEM: FLUID: SOLUTION: BOUNDARY: TAG
```

writes the solution on the boundary, this is either the pressure or the normal velocity, depending on the degree of freedom in the system.

Note! This is not the actual solution from the system matrix, (which is the fluid velocity potential and its normal derivative). This must be noted if comparisons are made with solutions from the system matrix `sysmat.out`.

The block with the header,

```
*OUTBEM: FLUID: SOLUTION: FIELD: TAG
```

details the pressure at the different field points. The field point is written first, followed by the real and imaginary parts of this pressure.

The 'powerTAG.out' file

This file writes the power radiated or possibly absorbed at a fluid boundary element subsystem. If the power data is written for the `F1` and `F2` tagged fluid BE-submodels, the first line in the output file will read,

```
* OUTSYS: POWER: DLOOP: F1    F2
```

The data for each line is then given by Table 3.2

Entry	Description
`1`	Frequency
`2`	Wavenumber
`3n`	Power
`3n+1`	Radiation efficiency
`3n+2`	Length of BE-domain

Table 3.2: Data order in the "powerTAG.out" file

where `n=1,2,3...` is the number of the subsystem that is written, e.g. `n=1` for `F1` and `n=2` for `F2` etc.

Thus, the power,the radiation efficiency and the length for the first fluid BE model for which the power is requested is found in entries 3, 4 and 5.

Chapter 4

Example of multidomain coupling

4.1 Topology

Here a multidomain model is considered to exemplify the capacity of WANDS. To make the model easy to follow it is kept very simple and it bares little resemblance to any physical model. The model elements and topology is shown in Figure 4.1.

The solid lines corresponds to FE models. Top left is a plate FE model, P1, with two plate elements. Top right is a solid FE model, S1, made of two triangular elements. The two FE models are coupled at the third node of the plate. There are four fluid BE models in this system, numbered, F1 to F4, from the top down. The F1 BE model is coupled both the plate and the solid. The F2 and F3 models are coupled together. A wave is impinging on the F2 model at an angle of 45 degrees. Finally, the F4 model has a rigid boundary. The system matrix assembled from this model will look like this,

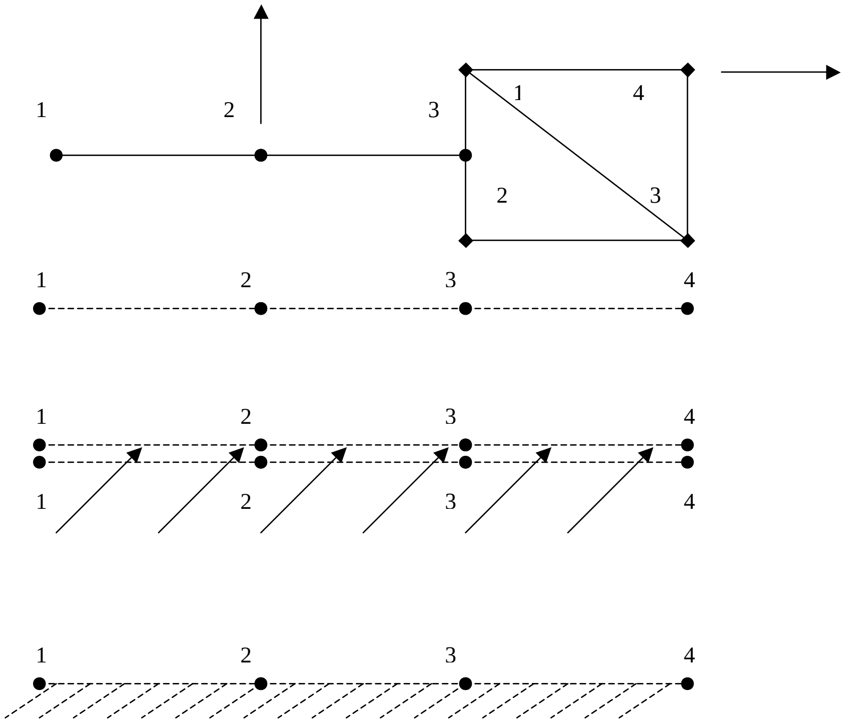

Figure 4.1: Simple example of multidomain coupling.

4.2 Input file

An input file for this system is given in the following,

```
*INSYS: TITLE:

Multidomain test 1 MULT1

*INSYS: WAVEDOMAIN:

 100.000,   0.500,   0.000
 100.000,   1.500,   0.000
 125.000,   0.500,   0.000
 125.000,   1.500,   0.000

*INSYS: MODELS: FE-PLATE:
** First row = name
** Second row = tag

STEEL PLATE P1
 1, 0

*INSYS: MODELS: FE-SOLID:
** First row = name
** Second row = tag

STEEL SOLID S1
 1, 0

*INSYS: MODELS: COUP: FE-PLATE: FE-SOLID:
** First row = name
** Second row = coupled systems (plate system #)&(fluid sysmtem #)
```

```
COUPLING PLATE SOLID P1S1 P1 S1
 1, 0

*INSYS: MODELS: BE-FLUID:
** First row = name
** Second row = tag

FLUID 1 F1
 1, 1, 1,
FLUID 2 F2
 1, 0, 1,
FLUID 3 F3
 1, 0, 1,
FLUID 4 F4
 1, 0, 1,
*INSYS: MODELS: COUP: FE-PLATE: BE-FLUID:
** First row = name
** Second row = coupled systems (plate system #)&(fluid sysmtem #)

COUPLING TO FLUID P1F1 P1 , F1
 1

*INSYS: MODELS: COUP: FE-SOLID: BE-FLUID:
** First row = name
** Second row = coupled systems (plate system #)&(fluid sysmtem #)

COUPLING TO FLUID S1F1, S1 , F1
 1

*INSYS: MODELS: COUP: BE-FLUID: BE-FLUID:
** First row = name
** Second row = coupled systems (plate system #)&(fluid sysmtem #)
```

```
COUPLING BETWEEN FLUIDS F2F3, F2 , F3
 1

*INSYS: END:




*INFEM: PLATE: MAT: ISO: P1
** material number, Youngs mod, Poissons ratio, Thickness, Density, Loss factor
 1,7.000e+010,   0.300,2.530e-003,2700.000,   0.000

*INFEM: PLATE: COORD: P1
**nodes on plate
      -0.020000,        0.020000
      -0.010000,        0.020000
       0.000000,        0.020000

*INFEM: PLATE: TOP: P1
**Finite elements of plate
   11,    1,    1,    2
   11,    1,    2,    3

*INFEM: PLATE: FORCE: P1
**element, local node, direction, real amplitude, complex amplitude
    2,    1,    3, 1.00, 1.00

*INFEM: PLATE: WRITE: WAVEDOMAIN: P1
**frequency and wavenumber where output should be written
 100.000,   1.500,   0.000
 125.000,   0.500,   0.000

*INFEM: PLATE: WRITE: TOP: P1
**element, local node, local dof for written solution
```

```
    1,    2,    4
    2,    1,    1

*INFEM: PLATE: END: P1



*INFEM: SOLID: MAT: ISO: S1
** material number, Youngs mod, Poissons ratio, Density, Loss factor
 1,7.000e+010,   0.300,2.700e+003,   0.020

*INFEM: SOLID: COORD: S1
**nodes of solid
       0.000000,        0.030000
       0.020000,        0.030000
       0.000000,        0.010000
       0.020000,        0.010000

*INFEM: SOLID: TOP: S1
**Finite elements of plate
    2,    1,    1,    3,    4
    2,    1,    1,    4,    2

*INFEM: SOLID: FORCE: S1
**element, local node, direction, real amplitude, complex amplitude
    2,    3,    2, 2.00, 1.00
*INFEM: SOLID: WRITE: WAVEDOMAIN: S1
**frequency and wavenumber where output should be written
 100.000,   0.500,   0.000
 125.000,   1.500,   0.000

*INFEM: SOLID: WRITE: TOP: S1
**element, local node, local dof for written solution
```

```
    2,    3,    2
    2,    2,    3

*INFEM: SOLID: END: S1



*INCOUP: FE-PLATE: FE-SOLID: P1S1
    2,    2,    1,    1,    2,  0.500000,10000000.000000

*INCOUP: FE-PLATE: FE-SOLID: WRITE: WAVEDOMAIN: P1S1
**frequency and wavenumber where output should be written
 100.000,   0.500,   0.000

*INCOUP: FE-PLATE: FE-SOLID: END: P1S1



*INBEM: FLUID: TYPE: F1
** Internal flag & unbounded
 0, 1

*INBEM: FLUID: MAT: F1
   1.21, 343.00,   0.00,100000.00

*INBEM: FLUID: NODES: F1
**nodes on boundary
      -0.020000,        0.010000
      -0.010000,        0.010000
       0.000000,        0.010000
       0.020000,        0.010000

*INBEM: FLUID: TOP: F1
```

```
** Topology 2 noded elements
    1,    2
    2,    3
    3,    4

*INBEM: FLUID: FIELD: F1
** y and z coordinates
      -0.010000,       0.050000
       0.000000,       0.050000

*INBEM: FLUID: END: F1




*INBEM: FLUID: TYPE: F2
** Internal flag & unbounded
 0, 1

*INBEM: FLUID: INFIELD: F2
** amplitude, angle and phase of incoming wave
 1.00,45.00, 0.00
*INBEM: FLUID: MAT: F2
   1.21, 343.00,   0.00,100000.00

*INBEM: FLUID: NODES: F2
**nodes on boundary
      -0.020000,        0.000000
      -0.010000,        0.000000
       0.000000,        0.000000
       0.020000,        0.000000

*INBEM: FLUID: TOP: F2
** Topology 2 noded elements
```

```
    1,    2
    2,    3
    3,    4

*INBEM: FLUID: WRITE: WAVEDOMAIN: F2
**frequency and wavenumber for which output should be written
 100.000,   0.500,   0.000

*INBEM: FLUID: WRITE: TOP: F2
**element, local node, local dof for written solution
    2,    1,    1
    2,    1,    2

*INBEM: FLUID: END: F2




*INBEM: FLUID: TYPE: F3
** Internal flag & unbounded
 0, 1

*INBEM: FLUID: MAT: F3
   1.21, 343.00,   0.00,100000.00

*INBEM: FLUID: NODES: F3
**nodes on boundary
      -0.020000,        0.000000
      -0.010000,        0.000000
       0.000000,        0.000000
       0.020000,        0.000000

*INBEM: FLUID: TOP: F3
** Topology 2 noded elements
```

```
    4,    3
    3,    2
    2,    1

*INBEM: FLUID: END: F3



*INBEM: FLUID: TYPE: F4
** Internal flag & unbounded
 0, 1

*INBEM: FLUID: MAT: F4
   1.21, 343.00,   0.00,100000.00

*INBEM: FLUID: NODES: F4
**nodes on boundary
      -0.020000,       -0.100000
      -0.010000,       -0.100000
       0.000000,       -0.100000
       0.020000,       -0.100000

*INBEM: FLUID: TOP: F4
** Topology 2 noded elements
    1,    2
    2,    3
    3,    4

*INBEM: FLUID: BC: F4
 1,    0.0000,    0.0000,   -1.0000,    0.0000,    0.2000,    0.0000
 2,    0.0000,    0.0000,   -1.0000,    0.0000,    0.2000,    0.0000
 3,    0.0000,    0.0000,   -1.0000,    0.0000,    0.2000,    0.0000
 4,    0.0000,    0.0000,   -1.0000,    0.0000,    0.2000,    0.0000
```

```
*INBEM: FLUID: END: F4



*INCOUP: FE-PLATE: BE-FLUID: ELEMENTS: P1F1
 1, 1,       0.000000,       1.000000, 0

*INCOUP: FE-PLATE: BE-FLUID: END: P1F1

*INCOUP: FE-SOLID: BE-FLUID: ELEMENTS: S1F1
 3, 1, 2, 3,       0.030000,       0.030000,-1

*INCOUP: FE-SOLID: BE-FLUID: END: S1F1



*INCOUP: BE-FLUID: BE-FLUID: TOP: F2F3
 1, 1, 3, 2,-1
 1, 2, 3, 1,-1
 2, 2, 2, 1,-1
 3, 2, 1, 1,-1

*INCOUP: BE-FLUID: BE-FLUID: END: F2F3
```

4.3 System matrix

$$\begin{bmatrix}
\mathbf{H_{F1}} & -\mathbf{G_{F1}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{H_{F2}} & -\mathbf{G_{F2}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H_{F3}} & -\mathbf{G_{F3}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H_{F4}} & -\mathbf{G_{F4}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
i\omega\rho_1\mu_1\mathbf{C_{1P1}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D_{P1}} & \mathbf{0} & \mathbf{C_{sp1}^{T}} \\
i\omega\rho_1\mu_1\mathbf{C_{1S1}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D_{S1}} & \mathbf{C_{sp2}^{T}} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C_{bF4}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{A_1} & \mathbf{A_2} & \mathbf{B_1} & \mathbf{B_2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{I_{2P1}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C_{2P1}} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{I_{2S1}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C_{2S1}} & \mathbf{0}
\end{bmatrix} \tag{4.1}$$

Note! It is useful to compare this ‘equation’ with Figure 4.2, that shows where the actual non-zero entries are.

4.4 Output files

The output files from this model follows.

4.4.1 Matrix files

For the plate model the following matrix output files are written,

```
a01_psP1.out
```

```
a02_psP1.out
```

```
a21_psP1.out
```

```
k0_psP1.out
```

`k4_psP1.out`

`m_psP1.out`

Example:

As an example the `m_psP1.out` file is:

```
   3   3    385.5780268325915693949923E-03
   3   4    543.7638734913396477976821E-06
   3   7    133.4693129109982556723679E-03
   3   8   -321.3150122701889568194167E-06
   4   3    543.7638734913396477976821E-06
   4   4    988.6615807850734673889589E-09
   4   7    321.3150122701889026093081E-06
   4   8   -741.4961806389386861511868E-09
   7   3    133.4693129109982556723679E-03
   7   4    321.3150122701889026093081E-06
   7   7    771.1560536651831387899847E-03
   7  11    133.4693129109982556723679E-03
   7  12   -321.3150122701889568194167E-06
   8   3   -321.3150122701889568194167E-06
   8   4   -741.4961806389386861511868E-09
   8   7   -108.4202172485504434007453E-21
   8   8      1.9773231615701473582944E-06
   8  11    321.3150122701889026093081E-06
   8  12   -741.4961806389386861511868E-09
  11   7    133.4693129109982556723679E-03
  11   8    321.3150122701889026093081E-06
  11  11    385.5780268325915693949923E-03
  11  12   -543.7638734913396477976821E-06
  12   7   -321.3150122701889568194167E-06
  12   8   -741.4961806389386861511868E-09
  12  11   -543.7638734913397562178994E-06
  12  12    988.6615807850736791471957E-09
```

Hence, the entry at row 3 and column 4 is, `543.7638734913396477976821E-06` and the entries listed are `0`.

Note! Since there is no damping in the plate there are no damping matrices written.

For the solid model the following matrix files are written,

```
a01_ssS1.out

b01_ssS1.out

k0_ssS1.out

c0_ssS1.out

k2_ssS1.out

c2_ssS1.out

m_ssS1.out
```

The coupling matrices between the plate and solid FE models are,

```
CP_P1S1.out

CS_P1S1.out
```

The system matrix file is called,

sysmat11_MULT1.out

This file has complex entries and the first entries in the system matrix file is written,

```
   1   1     50.0000000000000000000000E+03      0.0000000000000000000000E+00
   1   5   -450.3728892635749616601970E+00    124.9983805119668289762558E+00
   1   6   -679.2391631527348181407433E+00    249.9773576999272393095453E+00
   1   7   -797.8678353296842260533595E+00    374.8301864715142528439173E+00
   1   8   -470.7643817742237501988711E+00    249.7800711162615812099830E+00
   2   2     50.0000000000000000000000E+03      0.0000000000000000000000E+00
   .
   .
   .
```

A "spy-plot" of this matrix reveals where the non-zero entries are. This can be seen in Figure 4.2

Note! It is useful to compare Figure 4.2 with "equation" (4.1).

The system vector is written in,

sysvec11_MULT1.out

which has the same format as the matrix file, i.e. the column is written although its always 1.

4.4.2 Topology file

The topology for the model is written in the file,

FE_topologyMULT1.out

The content of this file is:

```
*OUTFEM: PLATE: SIZE: P1
**number of: dofs, elements, nodes, nodes on z-axis, restrained nodes
   12,    2,    3,    0,    0
```

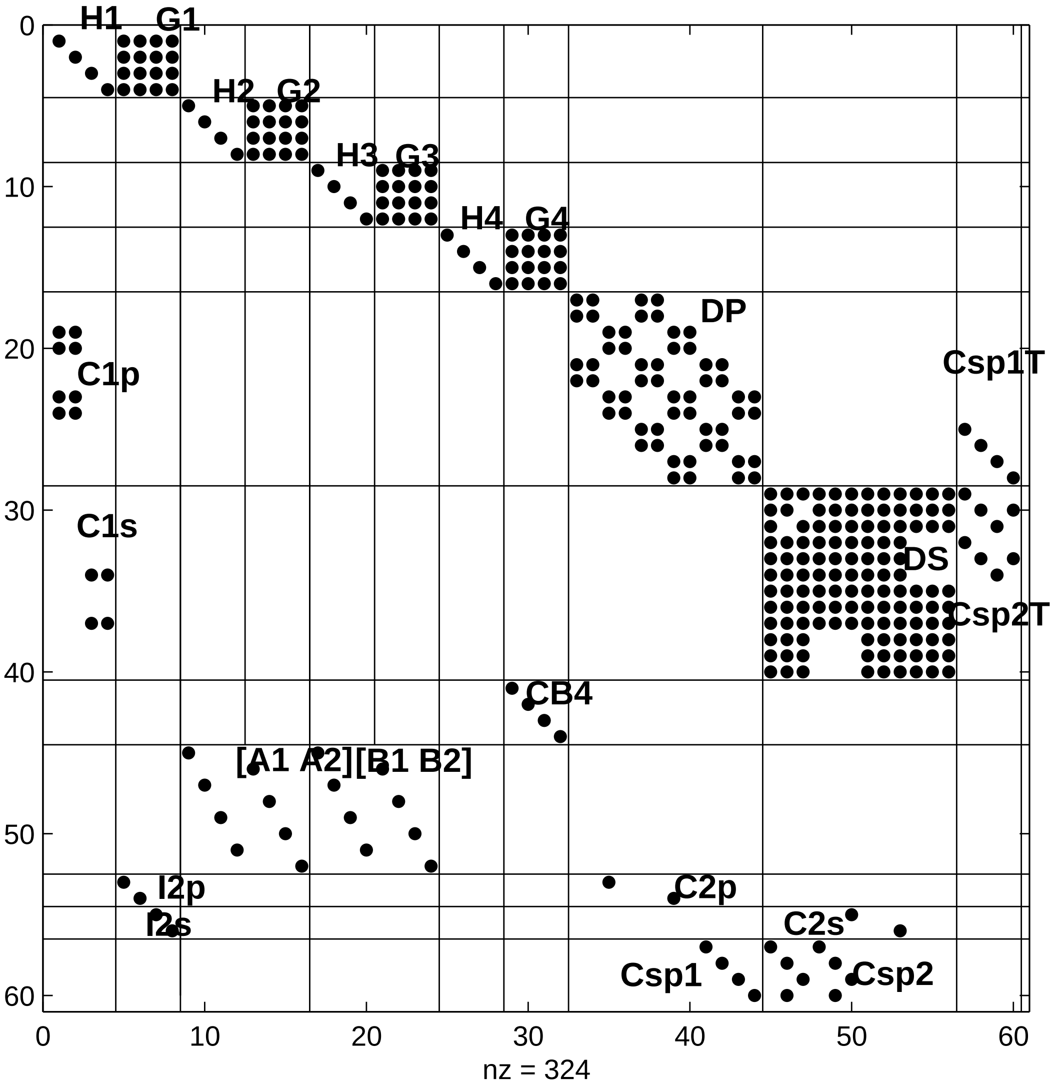

Figure 4.2: Simple example of multidomain coupling.

```
*OUTFEM: PLATE: ELETOP: P1
**elements and their node numbers
   1,   1,   2
   2,   2,   3

*OUTFEM: PLATE: NODEDOF: P1
**nodes and their dof numbers
    1,    1,    2,    3,    4
    2,    5,    6,    7,    8
    3,    9,   10,   11,   12


*OUTFEM: PLATE: NODECOORD: P1
** node number, y-coordinate, z-coordinate
    1, -0.020,  0.020
    2, -0.010,  0.020
    3,  0.000,  0.020

*OUTFEM: PLATE: ELEDATA: P1
**elements, element type, material number,
   1,  11,   1
   2,  11,   1

*OUTFEM: END: P1
*OUTFEM: SOLID: SIZE: S1
**number of: dofs, elements, nodes, nodes on z-axis, restrained nodes
   12,    2,    4,    0,    0

*OUTFEM: SOLID: NODECOORD: S1
** node number, y-coordinate, z-coordinate
    1,  0.000,            0.030
    2,  0.000,            0.010
    3,  0.020,            0.010
    4,  0.020,            0.030

*OUTFEM: SOLID: ELETOP: S1
**elements and their node numbers
   1,   1,   2,   3
   2,   1,   3,   4

*OUTFEM: SOLID: NODEDOF: S1
**nodes and their dof numbers
    1,    1,    2,    3
    2,    4,    5,    6
    3,    7,    8,    9
    4,   10,   11,   12

*OUTFEM: SOLID: ELEDATA: S1
**elements, element type, material number,
   1,   2,   1
   2,   2,   1

*OUTFEM: END: S1
*OUTFEM: COUPLING: SIZE:
**rows, columns, couplings
    4,   24,    1

*OUTFEM: COUPLING: PLATE: NODEDOFS:
**Nodenumber, dofs (4)
    3,    9,   10,   11,   12

*OUTFEM: COUPLING: PLATE: NODECOORDS:
**Nodenumber, y-coord, z-coord
    3, 0.000, 0.020

*OUTFEM: COUPLING: SOLID: NODEDOFS:
```

```
**Nodenumbers (2) and dofs (6)
    1,    2,    1,    2,    3,    4,    5,    6

*OUTFEM: COUPLING: SOLID: NODECOORDS:
**Nodenumbers (2) + y- and z-coord for coupling
    1,    2, 0.000, 0.010
```

4.4.3 Output solution files

There are two solution files written.

solution.out file

The first is the solution of different degrees of freedom in the model. This file is called,

`solutionMULTI1.out`

and is written as:

```
* OUTSYS: SOLUTION: DLOOP:
** outputs for double loop solution **

 *************************
* OUTSYS: WAVEDOMAIN:
  100.0000    0.5000

* OUTFEM: SOLID: SOLUTION: S1
   11     -4.166492578662040755833 6E-06     -2.0835066784371709822316E-06
    9    -10.3268652433837791813735E-06     -6.3081890450272507227876E-06

* OUTCOUP: FE-PLATE: FE-SOLID: SOLUTION: P1S1
    1     -2.7433813919419473984834E-09      5.0260650594723535973029E-09
    2     22.4371736726148144767695E-09     11.2172150047098120576689E-09
    3   -176.2843519389624602385487E-09   -133.6274841693544090705925E-09
    4      1.1629638711614068238003E-09    860.9694520751741685019031E-12

*OUTBEM: FLUID: SOLUTION: FIELD: F1
 ** Field point number, pressure
   1   0.11760E+04   0.85657E+03
   2   0.13726E+04   0.10490E+04

* OUTBEM: FLUID: SOLUTION: BOUNDARY: F1
    1    623.9044709684915233083302E+00    420.4933170111973481652967E+00
    2   -887.8783316899985038617160E+00   -555.8014257257027566083707E+00
    3     -3.8108479522688717224810E+03     -2.4924352272922105839825E+03
    4     10.7200362225402495823801E+03      9.6671792092180348845432E+03

* OUTBEM: FLUID: SOLUTION: BOUNDARY: F2
    2     -7.3106234943610230522120E-06     11.6785310208800396518424E-06
    6    -39.6355207226260990371013E-09     64.8856079664521850411990E-09



 **************************
* OUTSYS: WAVEDOMAIN:
  100.0000    1.5000

* OUTFEM: PLATE: SOLUTION: P1
    8   -170.8622417818891113824002E-09   -113.9917279696348071086175E-09
    5   -111.4805029272724156203840E-09    202.2199120147631675068151E-09

*OUTBEM: FLUID: SOLUTION: FIELD: F1
 ** Field point number, pressure
   1   0.12979E+04   0.91719E+03
   2   0.15166E+04   0.11124E+04
```

```
* OUTBEM: FLUID: SOLUTION: BOUNDARY: F1
    1    673.4151778368072882585693E+00    450.6933838806346557248617E+00
    2     -1.0228362780147924695484E+03   -637.7001736196050387661671E+00
    3     -4.2997304998204499497660E+03     -2.7816326771186913902056E+03
    4     11.9834380327341714291833E+03      9.9819826753244306019042E+03

* OUTBEM: FLUID: SOLUTION: BOUNDARY: F2
    2     -8.3878623836462694139801E-06     13.4536735222950494452344E-06
    6    -50.4945100832724170152955E-09     82.7899494803008104808759E-09



 **************************
* OUTSYS: WAVEDOMAIN:
  125.0000    0.5000

* OUTFEM: PLATE: SOLUTION: P1
    8   -128.9982519099285172021471E-09    -88.6466084543794495675199E-09
    5    -20.9626787953281573783617E-09     38.3633946017774730287088E-09

*OUTBEM: FLUID: SOLUTION: FIELD: F1
 ** Field point number, pressure
   1   0.11736E+04   0.86687E+03
   2   0.13697E+04   0.10682E+04

* OUTBEM: FLUID: SOLUTION: BOUNDARY: F1
    1    622.1102683896001508401241E+00    419.4127498852166695542110E+00
    2   -883.0806697231184898555512E+00   -552.9156980142005295419949E+00
    3     -3.7946360478077690459031E+03     -2.4894538229296204008278E+03
    4     10.6442091218566783936694E+03     10.0182719596782844746485E+03

* OUTBEM: FLUID: SOLUTION: BOUNDARY: F2
```

```
    2     -5.8181333191448287529076E-06      9.2923407428321011763787E-06
    6    -31.4211973286830690292269E-09     51.4312645186329173227937E-09



 **************************
* OUTSYS: WAVEDOMAIN:
  125.0000    1.5000

* OUTFEM: SOLID: SOLUTION: S1
   11     -2.7242167504724339905395E-06     -1.3642502711016589164104E-06
    9     -7.6122339506299225802654E-06     -4.6476289464383750747820E-06

*OUTBEM: FLUID: SOLUTION: FIELD: F1
 ** Field point number, pressure
   1   0.12459E+04   0.90357E+03
   2   0.14555E+04   0.11071E+04

* OUTBEM: FLUID: SOLUTION: BOUNDARY: F1
    1    651.5801722562193845078582E+00    437.5375939119343229322112E+00
    2   -962.4199062633100538732833E+00   -601.3021650901413295287057E+00
    3     -4.0828107540545220217609E+03     -2.6611209744122538722877E+03
    4     11.4264141117857252538670E+03     10.2314498624709685827838E+03

* OUTBEM: FLUID: SOLUTION: BOUNDARY: F2
    2     -6.3272867349391578574392E-06     10.1271990354938271880264E-06
    6    -36.5023923866967164943671E-09     59.7863456414619528608894E-09
```

power.out file

The second file is the power output file. This file is here called,
powerMULT1.out
and it is written,

```
* OUTSYS: POWER: DLOOP: F1    F2    F3    F4
** frequency, wavenumber, power for tags, rad eff for tags,  length for tags
** zero if not calculated

  100.0000    0.5000      2.86819E+003    4.76840E+027    4.76840E+027 ...
  100.0000    1.5000      2.60902E+003    1.91601E+027    1.91601E+027 ...
  125.0000    0.5000      2.65290E+003    9.24796E+027    9.24796E+027 ...
  125.0000    1.5000      2.54966E+003    6.14412E+029    6.14412E+029 ...
```

where the dots indicate that all the data can not be written on the width of the paper in this manual.

Chapter 5

Some tutorial examples

5.1 Introduction

This chapter describes four short example programs in WANDS. These examples where constructed for a mini-workshop on WANDS held in March 2007. The examples uses MATLAB for pre- and post-processing. The first example is described in the previous chapter. The following examples consider properties of a pipe.

The examples have been chosen to get the user started with WANDS. This means that the example codes could be used as a starting point for other applications. The examples should also give some hints about what to look for when debugging or validating a code.

5.2 Prerequisites

The WANDS-CD contains a number of files related to WANDS. Besides this *.pdf document the following files and folders are needed.

The `WANDS2_1.exe` file

The `WANDSoutput` folder

The `WANDSday` folder

The `Utilities` folder

First, copy the `WANDSday` folder to the `c:\` drive on your computer. Second, open MATLAB and make sure that copies of the folders 'Utilities' and 'WANDSoutput' are in your MATLAB path.

5.3 Multitest example

The first example is described at the end of Chapter 3 of the WANDS theory manual [1]. This example does not have much of a physical meaning, instead it is constructed to show an example of how the system matrix for each wavenumber and frequency pair is made of matrices from different subsystems.

5.3.1 Have a quick look at the configuration

First, have a look at this system in the last chapter and also have a look inside the input file `multi.dat`. As can be seen, there are many lines to be written even for this relatively simple problem. To avoid mistakes, the input files for WANDS have usually been automatically written by MATLAB programs. The `multi.dat` file was created by running the `multi_test.m` routine.

5.3.2 Run WANDS

Second, run the `WANDS_2_1.exe` program (double click the icon). When the program asks for an input file write the path for the file `multi.dat`, e.g.

```
c:\WANDSday\multi.dat
```

and give a suitable output directory when the program asks for that, e.g.

```
c:\WANDSoutput
```

The program should run just a few seconds and the calculation should end with a 'beep' sound. A system matrix has been formed and solved for a few wavenumbers and frequencies, which can be seen in the command line window.

5.3.3 Examine the system matrix

When checking for errors, either when developing or using WANDS, the system matrix is the first thing to be investigated. Therefore, the system matrix for the first wavenumber-frequency pair is always written in the output directory. This file is called `SYSMAT_001.out` and, like other matrix outputs, each row corresponds to one non-zero entry in the matrix. The first two integers in a row in the file are the entry's row and column in the matrix. The third and fourth numbers in a row are the real and imaginary part of the entry.

Import the system matrix to MATLAB.

One way to import the matrix into MATLAB is to use the '`read_mat.m`' MATLAB routine found in the `Utilities` folder, e.g.

```
>>SMAT=read_mat('c:\WANDSoutput\SYSMAT_001.out',0,0);
```

`SMAT` now contains the system matrix in the ordinary sparse MATLAB format. To view the spy plot of this matrix, simply write,

```
>>spy(SMAT)
```

The spy plot should be almost identical to that found in the previous chapter. One difference found when reviewing this example was that the $\mathbf{H_{F1}}$ block matrix at the upper left corner was full rather than diagonal. The diagonality of this matrix depends on the boundary surface to being straight line. Hence, this diagonality is be very sensitive to a small numerical errors. By viewing this upper left corner, it was found that the off-diagonal entries were about 16 orders of magnitude less than the diagonal entries and thus totally insignificant. This can also be verified by the reader.

5.4 Pipe example without fluid

In the remainder of these notes a pipe is considered. To begin with neither external nor internal fluid is considered. However, when internal fluid is considered the interior of the pipe must be meshed for the fluid finite elements.

5.4.1 The pipe mesh

A simple way to make meshes in MATLAB is to use the MATLAB pde-toolbox. If you have this toolbox installed you can view the mesh used here by writing the MATLAB commands,

```
>>load circ5_medium

>>figure

>>pdeplot(p,e,t)

>>axis equal
```

The mesh information is contained in the `p`, `e` and `t` variables. The MATLAB routine `pipe_A.m` uses this information to create a WANDS input file `pipe_A.dat`. Run `pipe_A.m` to make sure that the original version of the `*.dat` file is in the `c:\WANDSday` directory.

```
>>pipe_A
```

Of course the geometry of the perimeter of a pipe does not need a mesh generator at all, but the case in the next section where a fluid is introduced is greatly simplified by automatic meshing.

5.4.2 Run WANDS

Now, run WANDS again, this time the input file is `pipe_A.dat`, i.e.

```
c:\WANDSday\pipe_A.dat
```

for the input file and

```
c:\WANDSday
```

for the output folder. Again the calculations should just take a few seconds.

5.4.3 Import and save FE-matrices

Now, import the waveguide-FE matrices into MATLAB. Of course the `read_mat` routine may be used again. However an option is to use the routine `mtrx_conv_wands2`, which imports all matrices with one (rather long) command line in MATLAB, e.g.

```
>>[K4,K2,K1,K0,M0]=mtrx_conv_wands2('out_dir','c:\WANDSoutput','type'...

,'plate','tag','FEP1','systag','P_A');
```

These matrices will be accessed by other routines that import them from a file. Hence, save the matrices by, e.g.

```
>>save c:\WANDSday\pipe_A K4 K2 K1 K0 M0
```

5.4.4 Calculate dispersion relations

Now, inspect the dispersion relations, which is a good first investigation of any physical characteristics of a waveguide. As reported in [5], this is done by solving the eigenproblem,

$$\left[\mathbf{K_4}(-i\kappa)^4 + \mathbf{K_2}(-i\kappa)^2 + \mathbf{K_1}(-i\kappa) + \mathbf{K_0} - \omega^2\mathbf{M_2}\right]\tilde{\mathbf{W}} = \mathbf{0} \tag{5.1}$$

repeatedly for different real values of κ. The dispersion relations are shown by plotting these values of κ against the resulting eigenfrequencies.

Instead of writing a new routine for this task the routine, `disprel2` can be used. Run this routine with the command,

```
[freq_r,kappa_r]=disprel2('file','c:\WANDSday\pipe_A.mat','kappa',[0:0.1:10]);
```

Only frequencies up to about $2\,kHz$ are of interest here. Use the MATLAB command `axis` to zoom in to these,

```
>>axis([0 2000 0 10])
```

The result should look something like Figure 5.1,

As can be seen, there are three lines at low frequencies. Two of these lines are straight. The lower straight line corresponds to a longitudinal wave, whereas the straight line above corresponds to a torsional wave. The curved wave starting from zero frequency is the (Euler type) flexural wave. At higher frequencies, higher order waves cut-on.

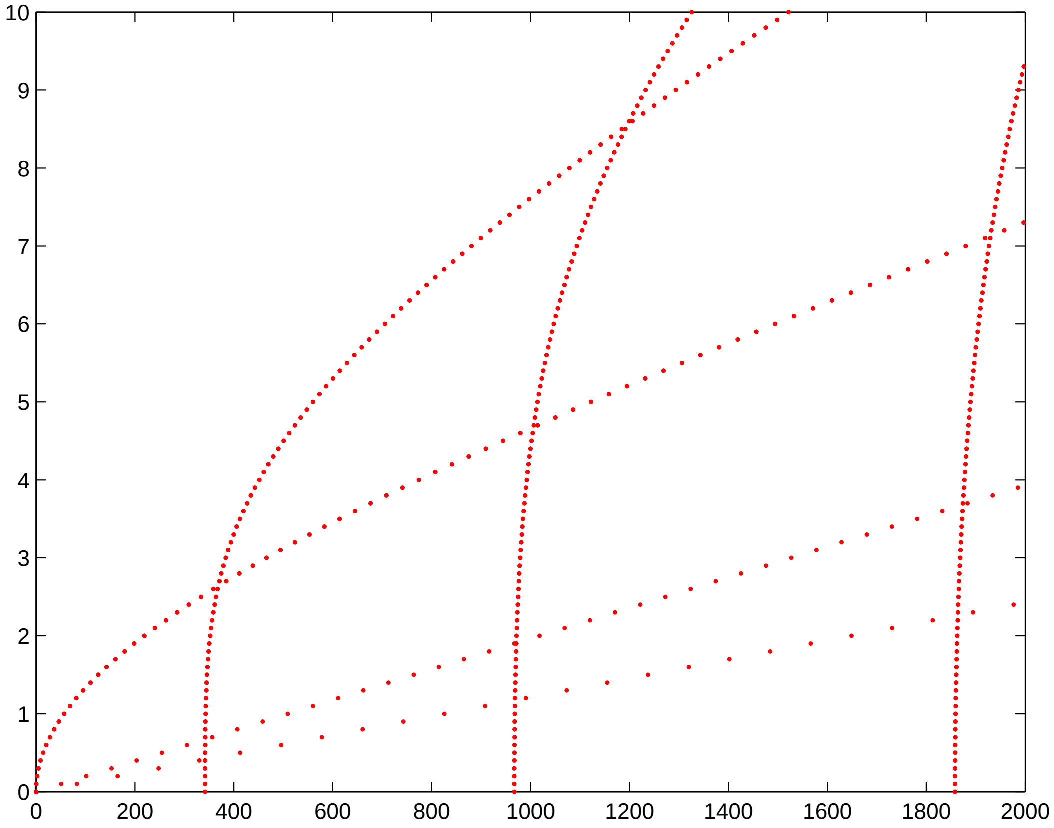

Figure 5.1: Dispersion relations for pipe

5.4.5 Show modeshape

For a pipe these higher order waves may be expressed by trigonometric functions. However, if more geometrically complicated structures are considered, the waveshapes can be found from the eigenvector, $\tilde{\mathbf{W}}$ in equation (5.1), where each entry corresponds to a nodal displacement along the x-axis. These displacements must of course be related to the geometry of the waveguide. For a plate-strip structure, like that analysed here, the node numbering is essentially the same as in the input file. For a solid or fluid mesh, however, WANDS changes the node numbering to make the matrix bandwidth smaller. Therefore, the correct relationships between degrees of freedom and coordinate displacements should be read from the output file `FE_topologyTAG.out`, where `TAG` is the `TAG` of the system. For the current pipe example `TAG = P_A`, thus the topology file is called `FE_topologyP_A.out`.

A mode plotting routine that reads data from this file and plots a mode shape is `mode_plot2`. To use this routine the frequency and wavenumber at, or very close to, a branch in the dispersion plot has to be specified. Get these by using the MATLAB `ginput`, command. (First make sure that the curtrent figure is the dispersion plot)

```
>>[freq0,kappa0]=ginput(1)
```

Now click on the dot at $\kappa = 5$ and $frequency \approx 560\,Hz$ in the dispersion plot window. Then use this data in the `mode_plot2` routine.

```
>>mode_plot2(freq0,kappa0,'out_dir','c:\WANDSday\','mtrx_file',...

'c:\WANDSday\pipe_A.mat','window',9,'type','plate','systag','P_A',...

'tag','FEP1','Amp',1)
```

(write it on one line). Zoom in MATLAB Figure 9. The result should look like Figure 5.2

Repeat for some other points of the dispersion relations, but notice that the amplitude, i.e. the value after `'Amp'`, might have to be changed to give a good plot.

5.4.6 Point mobility in an infinite pipe

First run

The solution for infinite waveguides subjected to point forces may be calculated with residue calculus, this procedure is detailed in [5] (Chapter 7) and in [6]. The major computational effort with this method is in the solving of equation (5.1) for a given frequency, thus finding complex wavenumbers and eigenvectors. This is a polynomial eigenproblem that is easiest to solve by using the `polyeig` MATLAB routine. Since this eigenproblem is computationally demanding as well as essential for many waveguide FE applications, other solution methods might be favoured, see [7] and [5] (p. 31-33). Another difference compared to the problem for dispersion relations is that the system here has damping.

The mobility solution for a radial point force on the `pipe_A` model is calculated by the `pipe_A_infsol.m`. Hence,

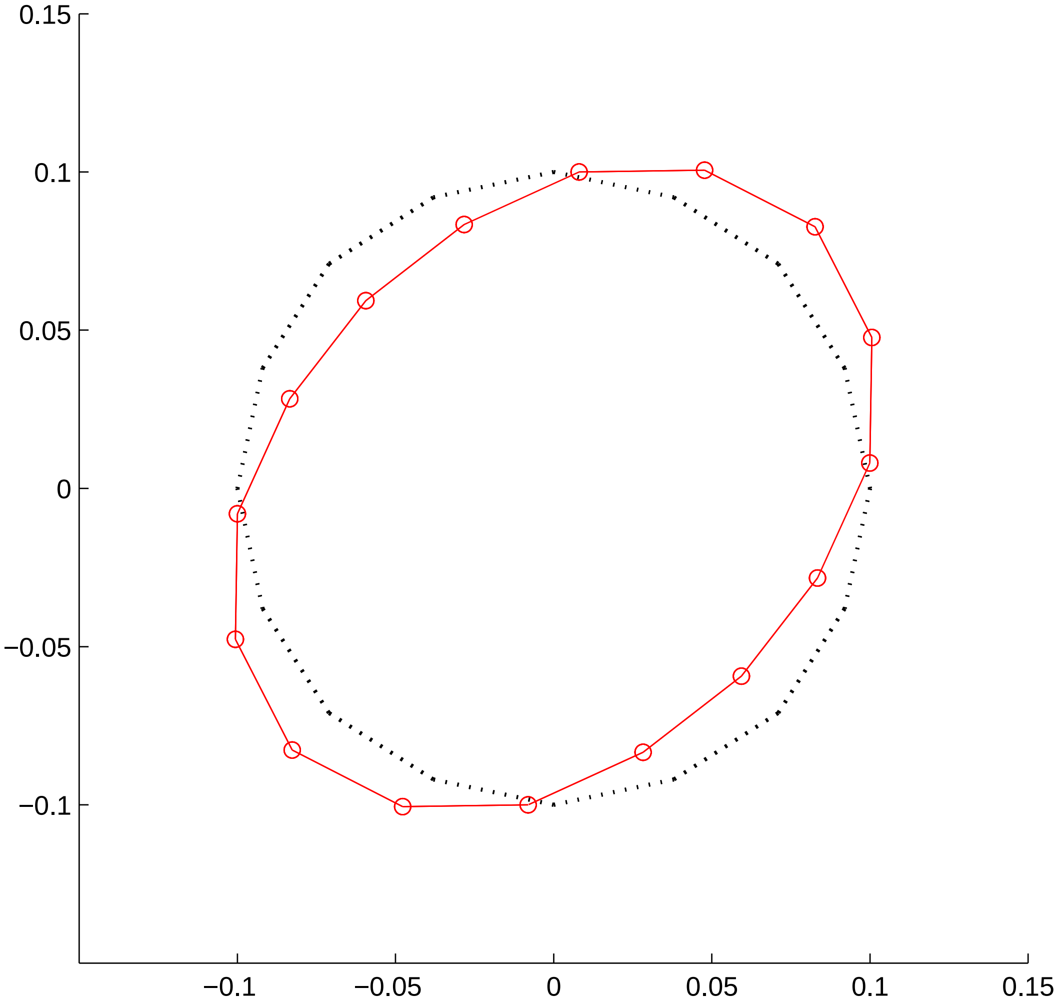

Figure 5.2: Wave-shape for pipe

```
>>pipe_A_infsol
```

The solution will take a few minutes to compute and the result will look something like 5.3.

The dashed line is the result for the corresponding Euler beam model. As seen, the numerical solution does not agree with the analytical solution.

Second run

Before explaining this error we change the input file slightly.

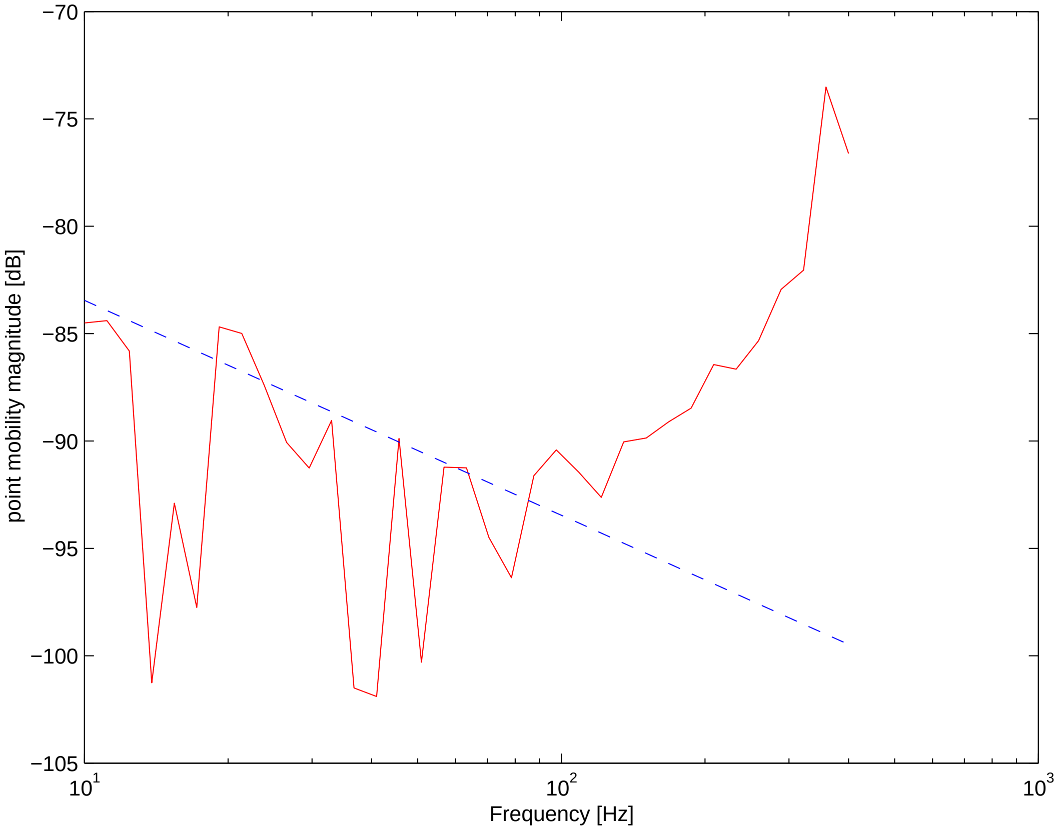

Figure 5.3: Point mobility for pipe

Open the `pipe_A.dat` file.

At the block heading `*INFEM: PLATE: MAT: ISO: FEP1` add a second plate 'material' with the same properties as the existing plate 'material' except for a slightly increased thickness of $6\,mm$ instead of $5\,mm$. The new material block will then be,

```
*INFEM: PLATE: MAT: ISO: FEP1|\\
** material number, Youngs mod, Poissons ratio, Thickness...
 1,2.100e+011,   0.300,5.000e-003,7800.000,   0.000
 2,2.100e+011,   0.300,6.000e-003,7800.000,   0.000
```

Next, change the first plate element of the pipe to have the 6 mm thickness. This is done by replacing the material specification from `1` to `2`. The new topology block will then begin with,

```
*INFEM: PLATE: TOP: FEP1
**FE elements on plate
   11,    2,    1,    9
   11,    1,    9,    5
   11,    1,    5,   10
   ...
   ...
```

Save and run WANDS for the new `pipe_A.dat` file. Then import the FE matrices into MATLAB and save them,

```
>>[K4,K2,K1,K0,M0]=mtrx_conv_wands2('out_dir','c:\WANDSoutput','type'...
,'plate','tag','FEP1','systag','P_A');
```

and

```
>>save c:\WANDSoutput\pipe_A K4 K2 K1 K0 M0
```

Then run the `pipe_A_infsol` program again

```
>>pipe_A_infsol
```

The mobility result this time looks much better (Figure 5.5).

Discussion

At low frequencies the numerical and analytical solutions agree well. At about 60 Hz the numerical solution flattens, which is what is expected when the flexural wave

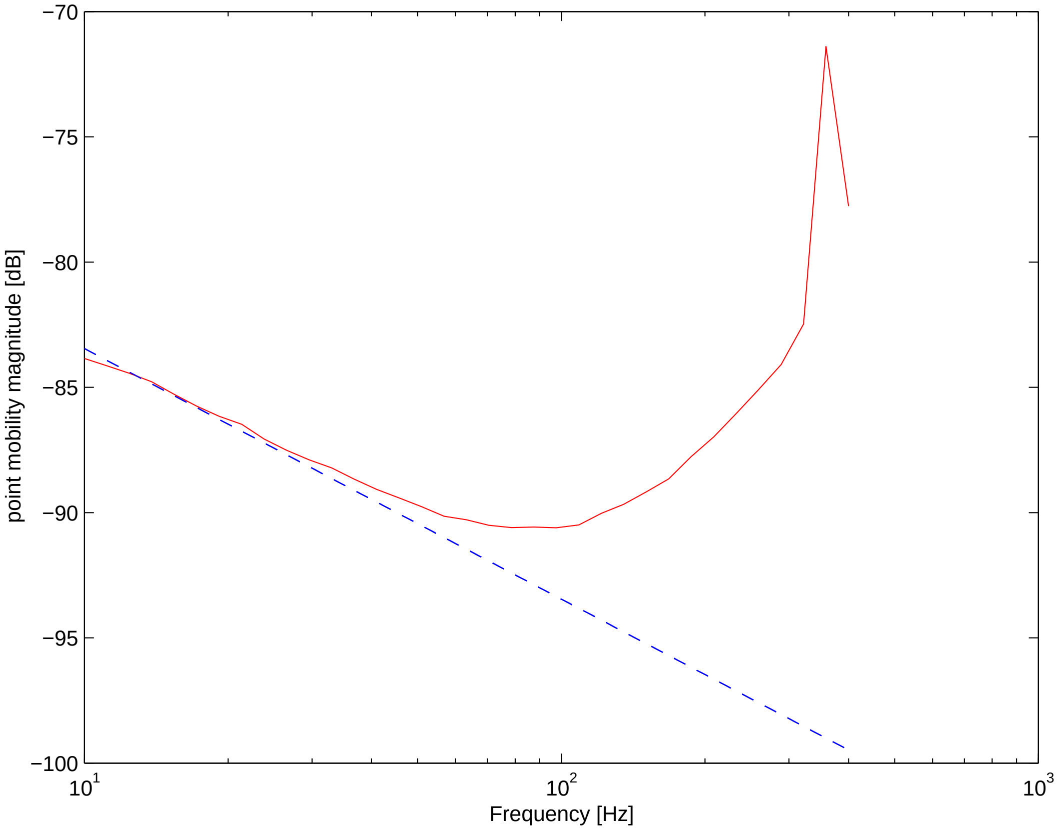

Figure 5.4: Point mobility for pipe

also includes shearing over the cross-section, e.g. for a Timoshenko type beam. The mobility peak at about $400\,Hz$ correspond to the first cut-on frequency. This behaviour is further explained in [6].

The reason for the numerical problems encountered in the first run, is due to the double symmetry of the pipe cross-section. For any (non-restrained) beam there are two flexural waves at low frequencies. However, for the double symmetry of the pipe cross-section these two waves have the same wavenumbers. The problem is that the eigensolver, can not distinguish between these two waves. Hence the eigenvectors (waveshapes) will not be correct and, as a consequence, the calculated mobility is also in error. When a small change in the material thickness is introduced, the ambiguity in the eigenvectors disappear and the mobility calculations become stable.

The reader may verify this by running the `disprel2` routine again and compare the resulting dispersion curves with those previously obtained. When zooming in on the flexural wave, it can be seen that there indeed are two ‘Euler-curves’ for the modified pipe model.

5.5 Fluid filled pipe

Now, an internal fluid is introduced in the pipe model. This means that there are three different sub-models, the pipe walls, the fluid and the coupling between the two. The WANDS input file is called `pipe_B.dat` and it is created by running the matlab routine `pipe_B.m`. The fluid is approximated with linear triangular elements and the mesh has already been discussed. Run `pipe_B.m`.

```
>>pipe_B.m
```

Then run WANDS with `pipe_B.dat` as the input file and `WANDSoutput` as the output directory.

Import the files into MATLAB with

```
>>[K4,K2,K1,K0,M0]=mtrx_conv_wands2('out_dir','c:\WANDSoutput'...
,'type','plate','tag','FEP1','systag','P_B');
```

to get the plate FE model matrices and

```
>> [K2f,K0f,M0f]=mtrx_conv_wands2('out_dir','c:\WANDSoutput',...
'type','fluid','tag','FEF1','systag','P_B');
```

to get the fluid FE matrices.

The coupling matrix may have rows and/or columns that are filled with zeros. Because of this its size may not be determined only by the indices of the entries. However, the size is fully determined by the sizes of the coupled FE-models. The plate model has 64 d.o.f. and the fluid model has 41 d.o.f. (this is easily verified by using the MATLAB `size` command). Only the upper (right) half of the antisymmetric coupling matrix is written from WANDS, to import this matrix into MATLAB write,

```
>>M1ur=read_mat_new('c:\WANDSoutput\M1_urPF1.out',[1 64],[1 41]);
```

5.5.1 Dispersion plot

The homogeneous system of equations for a combined plate-fluid FE system can be written as, see [5] or [8],

$$\left[\mathbf{K_4}(-i\kappa)^4 + \mathbf{K_2}(-i\kappa)^2 + \mathbf{K_1}(-i\kappa) + \mathbf{K_0} + i\omega\mathbf{M_1} - \omega^2\mathbf{M_2}\right]\tilde{\mathbf{W}} = \mathbf{0} \tag{5.2}$$

where

$$\mathbf{K_2} = \begin{bmatrix} \mathbf{K_{2f}} & \mathbf{0} \\ \mathbf{0} & \mathbf{K_{2p}} \end{bmatrix} \tag{5.3}$$

Subscripts '**p**' and '**f**' denotes plate and fluid system respectively. The other $\mathbf{K_j}$ and the $\mathbf{M_2}$ matrix have similar blocks, although the fluid blocks may be zero. The $\mathbf{M}_1$ matrix, is anti-symmetric and may be written as,

$$\mathbf{M_1} = \begin{bmatrix} \mathbf{0} & \mathbf{M_{1ur}} \\ -\mathbf{M_{1ur}}^T & \mathbf{0} \end{bmatrix} \tag{5.4}$$

Equation (5.2) signifies that, even for the dispersion relation where known real valued wavenumbers (i.e. propagating waves) are used, a quadratic eigenproblem has to be solved.

`pipe_B_disp` is a MATLAB routine that calculates these dispersion curves for the present example. Hence run,

```
>>pipe_B_disp
```

The resulting plot is shown below

The red (big) dots correspond to the coupled system. The black (small) dots correspond to the uncoupled plate system and the blue rings to the uncoupled fluid. In reference [9] an accurate model that utilizes the axisymmetry of fluid filled pipes is presented. Dispersion curve results from this method are found in the MATLAB

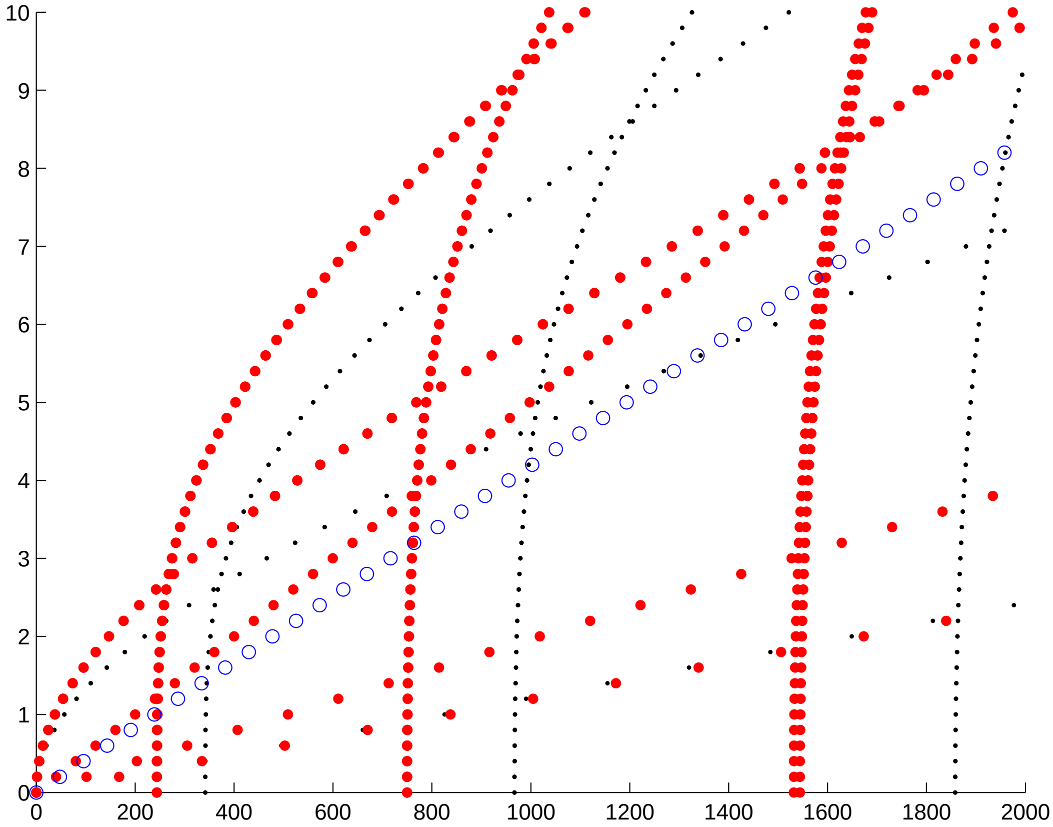

Figure 5.5: Dispersion relation for fluid filled pipe

figure file `disprel_pipe_val.fig`. Open this figure and plot the current coupled dispersion curves on this plot.

```
>>plot(omega/2/pi,kappa,'r.')
```

As can be seen, the differences for lower order waves, i.e. curves starting from zero frequency, are very small, whereas waves cutting on at higher frequencies show larger differences. This is expected since the fairly crude mesh cannot resolve the mode shapes for these waves. Also the overestimation of the cut on frequencies are expected since the stiffness matrices tends to have larger errors than the mass matrices. This is due to the derivatives involved when forming the former matrices.

5.6 External fluid

5.6.1 Improving conditioning

This example models the radiation of a pipe, subjected to a point load. The unbounded external fluid is modelled with a BE-model. The internal fluid is no longer needed and not included in the model. The modelled external fluid has properties typical of air. Since the pipe thickness is $5\,mm$ and has properties typical of steel, this causes a problem. If the ordinary velocity potential, ψ, defined such that the pressure, p is given by,

$$p = i\omega\psi \tag{5.5}$$

, the system matrix might become ill-conditioned. This is because the block-matrices for the plate strips and the fluid boundary elements will have very different orders of magnitude.

To illustrate this, run the MATLAB routine `pipe_C.m` with input argument =1.

```
>>Yfield=pipe_C(1);
```

This will create an input file `pipe_C.dat` that uses the velocity potential according to equation (5.6). (`Yfield` is just the coordinates for the field pressure).

Run this file in WANDS and import the first system matrix,

```
>>SMAT=read_mat('c:\WANDSday\SYSMAT_001.out',0,0);
```

Now consider the condition number of this matrix,

```
>> cond(full(SMAT))

ans =

  1.2878e+011
```

Hence, the condition number is in the order of $1 \cdot 10^{11}$, which is still satisfactory, but not really good. To improve the condition number, the velocity potential may be defined with an arbitrary scalar constant μ, see [8]. Then,

$$p = i\omega\mu\psi \tag{5.6}$$

Set $\mu = 1 \cdot 10^5$, by

```
>>Yfield=pipe_C(1e5);
```

and run WANDS again. Import the system matrix again and check the condition number. This will now have decreased to about $2 \cdot 10^7$. This seems to be the maximum reduction obtainable by setting the scaling factor μ.

Similar problems may also occur when thin plate strip elements are used. For these problems a possible cure is to use a length measure close to the thickness of the plate strips, i.e. using a mm length scale rather than one in metres. (But be careful to recalculate all parameter values when doing this).

5.6.2 Results

To view the present results, open the `solutionP_C.out` file. This file contains information about the results for different wavenumber-frequency pairs. First results for the frequency 100 Hz and wavenumber $0m^{-1}$ are tabulated, scroll down to

```
*OUTBEM: FLUID: SOLUTION: FIELD: BEF1|
 ** Field point number, pressure
```

Then copy and paste the 100 rows below into MATLAB. When you paste into MATLAB, first write,

```
>>field0=[
```

Then paste the copied section, e.g,

```
  1     -3.2561925953759210741745E-03    107.41705017683799078861 28E-06
```

Finally write,

```
];
```

to complete the input.

Repeat this copy and paste for the field for the last calculated frequency and wavenumber ($200\,Hz$ and $2.0\,m^{-1}$). Call this `field2`.

Now plot the results of `field0` in MATLAB figure 20.

```
>> figure(20)
>> plot(Yfield,field0(:,2),'k')
>> hold on
>> plot(Yfield,field0(:,3),'k--')
```

and the results of `field2` in MATLAB figure 22.

```
>> figure(22)
>> plot(Yfield,field2(:,2),'k')
>> hold on
>> plot(Yfield,field2(:,3),'k--')
```

The solid lines in these figures are the real values of the pressure and the dashed are the imaginary parts. These MATLAB figures are reproduced in Figures 5.6 and 5.7.

An analytical solution for the problem is calculated and plotted with the MATLAB routine, `ana_pipe_rad`,

```
>>ana_pipe_rad
```

MATLAB figure 20 (Figure 5.6 in this document) shows the two-dimensional case, since the wavenumber used there is zero and the analytical solution is a type of Hankel function.

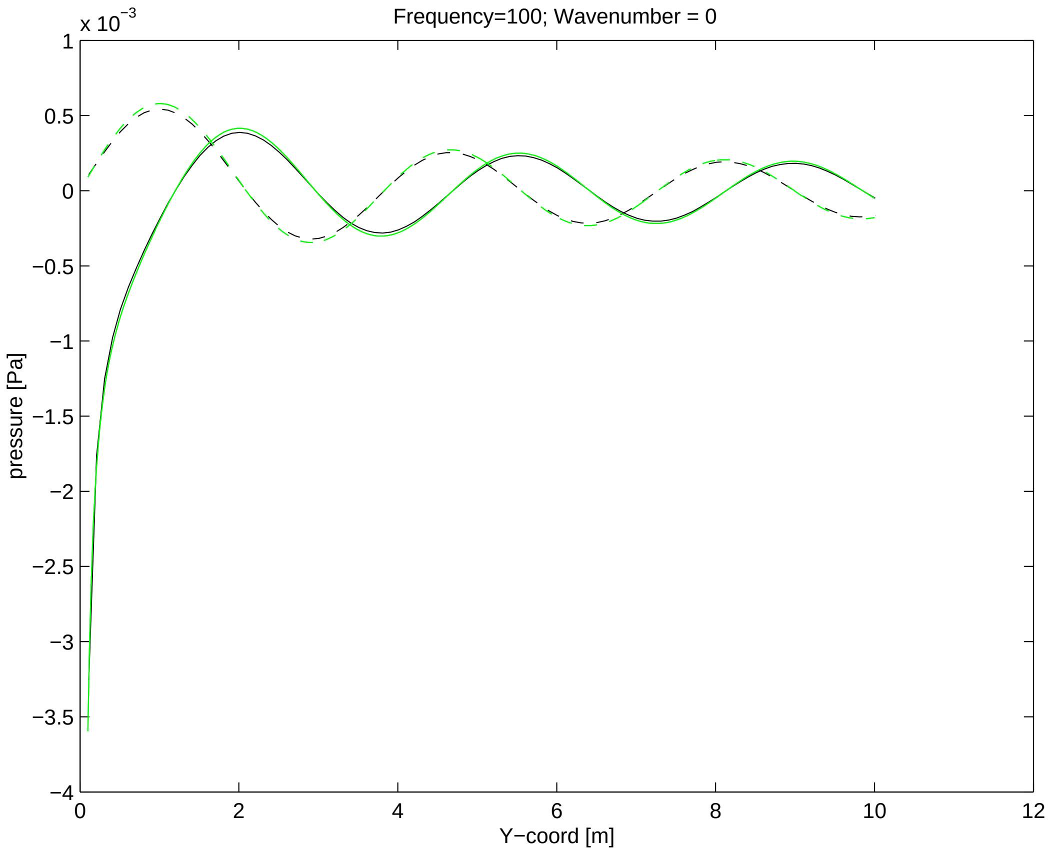

Figure 5.6: Pressure radiation from pipe

MATLAB figure 22 (Figure 5.7 in this document) on the other hand, show a nearfield solution. This is because the imposed wavenumber there is 2.00 which is greater than the wavenumber for freely propagating waves in air at this frequency, $(\omega/c_f = (2\pi \cdot 100/343 \approx 1.83))$.

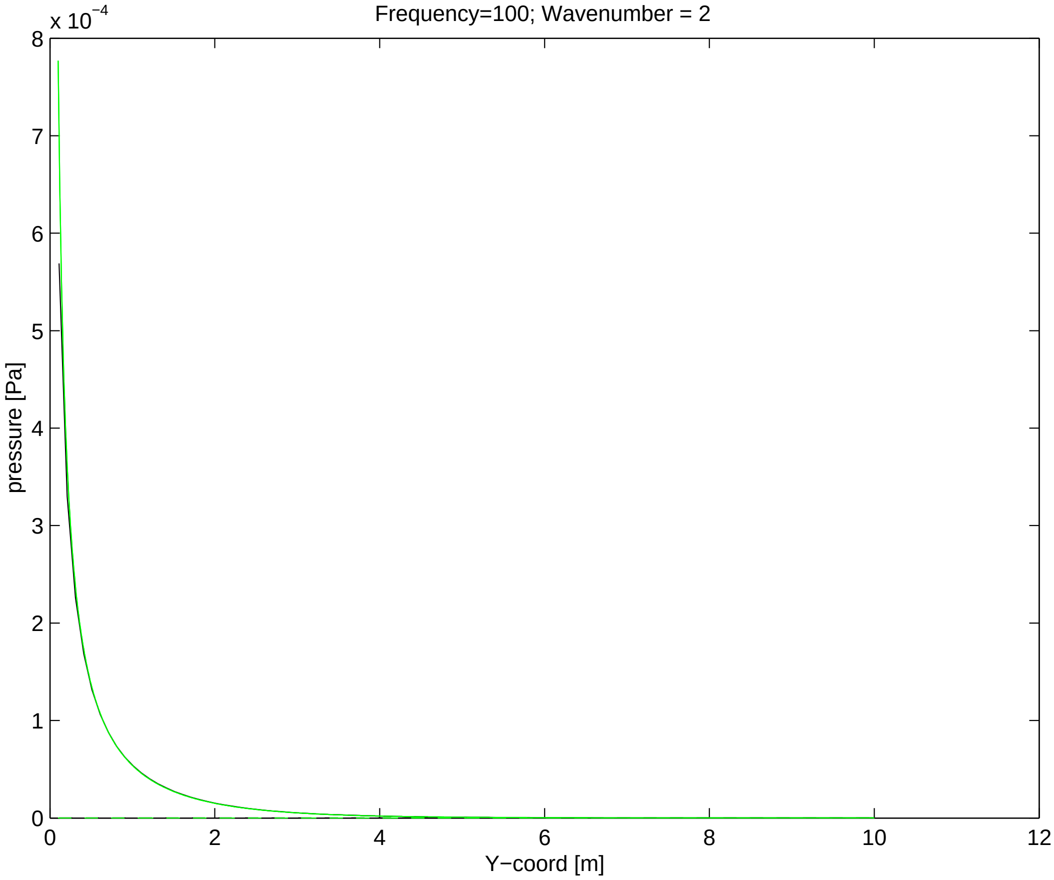

Figure 5.7: Pressure radiation from pipe

For practical applications nearfield solutions are often not needed since they do not radiate any acoustic energy. Hence, the wavenumber domain is often limited to $k < \omega/c_f$.

References

[1] Nilsson C-M. Jones C.J.C. Theory manual for wands 2.1 wave-number-domain fe -be software for structures and fluids. ISVR Technical Memorandum No: 975, 2007.

[2] Sheng X. Jones C.J.C. Thompson D.J. Computer programs for the discrete wavenumber finite/boundary element model for ground vibration and noise from railway tunnels. ISVR Technical Memorandum No:915, 2003.

[3] Cook R.D. Malkus D.S. Plesha M.E. *Concepts and applications of finite element analysis.* Wiley, cop, third edition, 1988.

[4] Wu T.W. *Boundary element acoustics: fundamentals and computer codes.* 2000.

[5] Nilsson C-M. *Waveguide Finite Elements Applied on a Car Tyre.* PhD thesis, Royal Institute of Technology, Aeronautical and Vehicle Engineering /MWL, 2004. TRITA-AVE 2004:21, ISBN 91-7283-798-5, http://media.lib.kth.se/dissengrefhit.asp?dissnr=3812.

[6] Nilsson C.M. and Finnveden. S. Input power to waveguides calculated by a finite element method. *Journal of Sound and Vibration*, 305, 2007.

[7] Tisseur F. and Meerbergen K. The quadratic eigenvalue problem. *SIAM Review*, 43(2):235–286, 2001.

[8] Nilsson C.M. and Finnveden. S. Waves in thin-walled fluid-filled ducts with arbitrary cross sections.

[9] Finnveden S. Spectral finite element analysis of the vibration of straight fluid-filled pipes with flanges. *Journal of Sound and Vibration*, 199(1):125–154, 1996.

Index